

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

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

СИСТЕМА УПРАВЛІННЯ МЕХАНІЗМОМ ПІДЙОМУ ШАХТНОЇ ПІ-
ДІЙМАЛЬНОЇ УСТАНОВКИ З НЕЙРОРЕГУЛЯТОРОМ NN
PREDICTIVE CONTROLLER
(тема кваліфікаційної роботи)

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

_____/ Тетяна БОГДАНОВА
(підпис) (ім'я та прізвище)

Керівник _____/ Олена БЛИЗНИЧЕНКО
(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 магістерського рівня вищої освіти

студенту (ці) Тетяна БОГДАНОВА
(ім'я, прізвище)

1. Тема роботи Система управління механізмом підйому шахтної підіймальної установки з нейрорегулятором NN Predictive Controller
затверджена наказом по університету № 4801-5/3704 від “21” листопада 2024 р.
2. Термін здачі закінченої роботи “5” грудня 2024 р.
3. Вихідні дані до роботи Вантажопідйомність судин $Q_{\text{п}}=15,2 \cdot 10^3$ кг; продуктивність $A=600$ т/год; висота підйому $H=580$ м; маса порожнього скипа $m_{\text{ск}}=9,8 \cdot 10^3$ кг; діаметр барабана $D_{\text{б}}=5$ м; шлях дотягування $S_{\text{дот}}=2$ м; швидкість дотягування $V_{\text{дот}}=0,5$ м/с; маса погонного метра каната $m_{1\text{к}}=11,6$ кг; тривалість паузи між циклами $t_{\text{п}}=11$ с. Система управління – астатична. Перерегулювання не більше 10%, час регулювання до 5 с
4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму підйому шахтної підіймальної установки. Розрахувати двоконтурну систему підлеглого регулювання з внутрішнім контуром струму і зовнішнім контуром швидкості. Оптимізувати контур струму за модульним критерієм, а контур швидкості – за симетричним критерієм. Розробити математичну модель трьохмасової системи. Виконати моделювання на ЕОМ та провести аналіз показників якості перехідних процесів змінних стану трьохмасової системи. Розробити структурну схему нейромережової системи управління. Виконати синтез нейрорегулятора на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Провести аналіз результатів моделювання системи з нейрорегулятором.
5. Перелік графічного матеріалу (презентаційний матеріал) 1. Магістерська кваліфікаційна робота. Формальні ознаки. 2. Механізм підйому. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Управління з прогнозом. Математична модель. 6. Нейромережові системи управління. Структурні схеми. 7. Система управління. Перехідні процеси. 8. Трьохмасова система. Математична модель. 9. Трьохмасова 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

Олена БЛИЗНИЧЕНКО

(ім'я, прізвище)

Завдання прийняв до виконання

Тетяна БОГДАНОВА

(ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфік
аційної 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 трьохмасової системи та моделювання трьохмасової системи на ЕОМ.	16.10.2023-05.11.2023	
4	Синтез нейромережової системи управління на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2024-25.11.2024	
5	Оформлення роботи	26.11.2024-05.12.2024	

Студент (ка)

Тетяна БОГДАНОВА

(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 219 сторінок, 63 рисунок, 3 таблиці, 70 використаних джерела, 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 трьохмасовою 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 трьохмасової системи. Розглянута структура нейромережевої системи управління і виконаний синтез нейрорегулятора NN Predictive Controller в системі MATLAB. Проведено моделювання нейромережевої системи управління з нейрорегулятором на ЕОМ. Встановлено, що синтезована нейромережева система має високі показники якості функціонування.

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

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

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

ABSTRACT

An explanatory message consists of: 219 pages, 63 figures, 3 tables, 70 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 threemass 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 threemass system are calculated. The structure of the network control system is considered and the synthesis of neuroregulator NN Predictive Controller is executed in the system of MATLAB. The simulation of neural network control system with neuroregulation is conducted on a computer. It is set that synthesized the neural network system has high indexes of quality of functioning.

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

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

Keywords: ARTIFICIAL NERO, NEURAL NETWORKS, MINE LIFTING SETTING, ELECTRIC DRIVE OF THE LIFTING MECHANISM, SYSTEM OF SLAVE CONTROL, ONEMASS AND THREEMASS SYSTEM, NEURAL NETWORK SYSTEM, NEUROREGULATOR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

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

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ РОБОТИ	
І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	14
1.1 Основні етапи розвитку теорії штучних нейронних мереж	14
1.2 Аналіз структур систем нейрорегулювання. Вибір структури системи	18
1.3 Вимоги до систем автоматизації і електроприводів шахтних підйомних машин.....	29
1.4 Постановка задачі синтезу нейромережевої системи управління	31
ВИСНОВКИ ДО РОЗДІЛУ 1	36
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПІДЙОМУ ШАХТНОЇ ПІДЙІМАЛЬНОЇ УСТАНОВКИ ...	38
2.1 Технічні характеристики шахтної підйимальної установки. Вимоги до системи управління.....	38
2.2 Вибір системи електроприводу шахтної підйимальної установки.....	39
2.3 Розрахунок швидкісної і навантажувальної діаграм електроприводу	42
2.4 Вибір електродвигуна по потужності	49
2.5 Вибір основного електроустаткування силового кола електроприводу	51
2.6 Розрахунок параметрів силового кола електроприводу	55
2.7 Функціональна схема системи автоматичного управління	59
2.8 Рівняння динаміки і передавальні функції об'єкту управління.....	62
2.9 Синтез послідовного коректуючого пристрою контура струму	66
2.10 Синтез послідовного коректуючого пристрою контура швидкості	71
2.11 Дослідження динамічних характеристик системи	79
ВИСНОВКИ ДО РОЗДІЛУ 2	83

РОЗДІЛ 3 ДОСЛІДЖЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ВЛАСТИВОСТЕЙ ПІДЙОМНОГО КАНАТА	85
3.1 Математическая модель трюхмасової системи	85
3.2 Розрахунок перехідних процесів трюхмасової системи	95
ВИСНОВКИ ДО РОЗДІЛУ 3	105
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	106
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	106
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	108
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	111
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	154
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	156
ВИСНОВКИ ДО РОЗДІЛУ 4	164
ВИСНОВКИ	166
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	168
ДОДАТОК ДОДАТОК А. МОДЕЛЬ НЕЙРОНА	174
А.1 Структурна схема нейрона	174
А.2 Функції активації нейронів	175
А.3 Нейрон з векторним входом.....	178
ДОДАТОК Б. АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ.....	180
Б.1 Одношарові мережі.....	180
Б.2 Багатошарові мережі	182
Б.3 Мережі з прямою передачею сигналу	185
ДОДАТОК В. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ	187
В.1 Завдання навчання.....	187
В.2 Процедури адаптації і навчання	188

В.3 Явище перенавчання	190
В.4 Властивість узагальнення	193
В.5 Методи навчання нейронних мереж.....	195
ДОДАТОК Г. УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ	
З ПРОГНОЗОМ	204
Г.1 Алгоритм нейроуправління з прогнозомі	204
Г.2 Структура нейронної мережі для моделювання об'єкту.....	211
Г.3 Прогноз з використанням нейронної мережі.....	215
Г.4 Похідні рівнянь нейронної мережі.....	216
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	218

ВСТУП

Актуальність теми. Вдосконалення систем автоматизованого управління промисловими установками засноване на новітніх наукових та технічних досягненнях і є критично важливим для підвищення продуктивності виробництва, зростання ефективності праці та покращення якості виготовляної продукції. Еволюція автоматизованих систем управління включає в себе розширення практичного застосування адаптивного управління, вдосконалення векторного управління електроприводами із змінним струмом, широке використання управляючих обчислювальних систем на різних рівнях складності, розвиток прямого цифрового управління та інших технологічних напрямів. Одним із перспективних шляхів є використання систем управління на основі нейромереж.

Системи управління з застосуванням нейромереж належать до нелінійних динамічних систем. У цих системах штучна нейронна мережа може виконувати різноманітні функції. Це, по-перше, функція адаптивного регулятора нелінійного багатозв'язаного об'єкту. Тут можливі два варіанти функціонування нейронних мереж. По-перше – нейронна мережа навчається і одночасно формує дію, що управляє, на вході виконавчого пристрою системи управління. Мета навчання мережі і мета управління об'єктом співпадають, що відбивається в заданні єдиній цільовій функції системи. Мережа навчається в реальному часі, в темпі протікання процесів в системі (режим on-line). У другому варіанті робота мережі складається з двох етапів: 1) попереднього етапу навчання мереж заданої оптимальної функції управління і 2) етапу відтворення апроксимації цієї функції в режимі управління об'єктом за тих же умов або близьких до них. Цільові функціонали навчання мережі і управління об'єктом можуть відрізнятися один від одного. Такий варіант застосування нейронної мережі для управління – так зване супервізорне управління – знайшов переважне розповсюдження до теперішнього часу, хоча процес синтезу нейромережевого контролера і настройка його параметрів в цьому випадку протікає не в реальному часі (режим off-line).

Вибір підходу до навчання мережі (on-line або off-line) залежить від особливостей завдання та визначає тип алгоритму навчання (наприклад, безпошукові чи пошукові схеми, глобальна чи локальна оптимізація тощо). Наприклад, у галузях промисловості, де зібрані значні обсяги даних про поведінку технічного об'єкта, більш ефективним є використання off-line навчання з застосуванням генетичних алгоритмів, методів випадкового пошуку чи «статистичного» навчання. З іншого боку, для об'єктів, технічні характеристики яких змінюються під час експлуатації, найзручніше використовувати on-line алгоритми для налаштування мережі.

Нейронні мережі знаходять застосування як ідентифікатори для оцінювання вектора стану нелінійних систем і як розширені фільтри Калмана.

Застосування нейронної мережі як оптимізатора для настройки параметрів регуляторів з типовими законами регулювання і для настройки параметрів алгоритмів адаптації, що реалізуються на основі відомих методів теорії адаптивних систем. Для позначення класу нелінійних систем з такого роду застосуванням штучних нейронних мереж використовується термін нейромережеві системи управління. Найбільше розповсюдження для цілей управління знайшли багатопарові нелінійні нейронні мережі [1-12].

Завдання щодо подальшого підвищення ефективності роботи підприємств гірничодобувної промисловості не можуть бути реалізовані без впровадження автоматизації виробничих процесів. Це, у свою чергу, призводить до зростання вимог до систем управління та електроприводів підйомних установок, які є ключовими елементами технологічного комплексу для видобутку 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.

Тому тема роботи, присвячена розробці сучасних систем управління механізмом підйому шахтної підйимальної установки, є актуальною.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління електроприводом механізму підйому шахтної підйимальної 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 трьохмасовою електромеханічною системою механізму підйому шахтної підйимальної установки.

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

5. Одержали подальший розвиток методи синтезу нейромережових структур з прогнозом для управління складними динамічними об'єктами з математичними моделями у вигляді трьохмасових електромеханічних систем.

Практичне значення одержаних результатів полягає в розробці нейромережової системи управління трьохмасовою електромеханічною системою механізму підйому шахтної підйимальної установки, що забезпечує високоякісне регулювання.

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

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

Серед них: синтезована система підлеглого регулювання координат електроприводу; розроблена модель системи з урахуванням пружних властивостей

підйомного 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 ННІ «УПА» ХНУ ім. В.Н. Каразіна (Харків, 2024 р.).

Структура і об'єм роботи Робота складається з вступу, 4 розділів основної частини, висновків, додатків і списку використаних літературних джерел. Повний об'єм роботи складає 219 сторінок, серед них: 39 рисунків за текстом, 24 рисунків на 24 окремих сторінках, 3 таблиці, 5 додатків на 46 сторінках, список літературних джерел з 70 найменування на 6 сторінках.

РОЗДІЛ 1

ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ РОБОТИ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Основні етапи розвитку теорії штучних нейронних мереж

Штучні нейронні мережі створюються за принципами організації та роботи їх біологічних аналогів. Вони мають здатність вирішувати різноманітні завдання, такі як розпізнавання образів, ідентифікація, прогнозування, оптимізація та управління складними об'єктами. Зростання продуктивності комп'ютерних систем значною мірою асоціюється з використанням ШНМ, особливо нейрокомп'ютерів (НК), основою яких є штучні нейронні мережі.

Термін «нейронні мережі» з'явився ще до середини 50-х років XX століття. Основні досягнення в цій галузі пов'язані з працями таких вчених, як У. Маккалох, Д. Хебб, Ф. Розенблатт, М. Мінський та Дж. Хопфілд. У 1943 році У. Маккалох (W. McCulloch) та У. Питтс (W. Pitts) [13] запропонували модель нейрона і виклали основи теорії функціонування мозку. У 1949 році Д. Хебб (D. Hebb) [14] описав природу з'єднань нейронів у мозку та їхню взаємодію (клітинні ансамблі, синаптична пластичність), а також вперше запропонував правила навчання для нейронних мереж. У 1957 році Ф. Розенблатт (F. Rosenblatt) [15,16] розробив принципи організації та функціонування персептронів і представив технічну реалізацію першого в світі нейрокомп'ютера Mark. В 1959 р. Д. Хьюбел (D. Hubel) і Т. Визель (T. Wiesel) [17] показали розподілений і паралельний характер зберігання і обробки інформації в біологічних нейронних мережах. В 1960-1968 рр. проводились активні дослідження в області штучних нейронних мереж, наприклад, адаліна і мадалина В. Уидроу (W. Widrow) (1960-1962 р.р.) [18,19], асоціативні матриці К. Штайнбуха (K. Steinbuch) (1961 р.) [20]. В 1969 р. було опубліковано книгу М. Мінського (M. Minsky) і С. Пейперта (S. Papert) «Персептрони» [21], в якій доводиться принципова обмеженість можливостей персептронів, внаслідок чого сталося згасання інтересу до штучних нейронних мереж.

В кінці 1970-х рр. відбулося відновлення інтересу до штучних нейронних мереж як наслідок накопичення нових знань про діяльність мозку, а також значного прогресу в області мікроелектроніки і комп'ютерної техніки. В 1982-1985 рр. Дж. Хопфілд (J. Hopfield) [22,23] запропонував сімейство оптимізуючих нейронних мереж, що моделюють асоціативну пам'ять. В 1985 р. з'явилися перші комерційні нейрокомп'ютери, наприклад, Mark III фірми TRW (США). В 1987 р. розпочалося широкомасштабне фінансування розробок в області ШНМ і НК в США, Японії і Західній Європі (японська програма «Hitap Frontiers і європейська програма «Basic Research in Adaptive Intelligence and Neurocomputing»). В 1989 р. розробки і дослідження в області ШНМ і НК ведуться практично всіма крупними електротехнічними фірмами. Нейрокомп'ютери стають одним з найдинамічніших секторів ринку. Агентством DARPA (Defence Advanced Research Projects Agency) міністерства оборони США почато фінансування програми по створенню надшвидкодійних зразків НК для різноманітних застосувань.

В 1992 р. роботи в області ШНМ знаходяться стадії інтенсивного розвитку. Щорічно проводиться десятки міжнародних конференцій і форумів по нейронних мережах, число спеціалізованих періодичних наукових видань з вказаної тематики досягло двох десятків найменувань. В 1993 р. число міжнародних конференцій з ШНМ і НК досягло ста. В 1994 р. річний об'єм продажів на ринку ШНМ і НК перевищив 2 млрд. доларів, а щорічний приріст склав 50%.

В середині дев'яностих років минулого сторіччя відбулася активізація дослідницьких організацій в області ШНМ і НК в Україні (Інститут кібернетики ім. Глушкова в Києві, Харківський політехнічний інститут, Харківський інститут радіоелектроніки і ін.). В Національному технічному університеті «ХПІ» проводяться роботи по застосуванню методів нейронних мереж в рішенні задач управління електромеханічними системами [24,25]. Вчені продемонстрований виключно високий потенціал поєднання двох обчислювальних технологій — штучних нейронних мереж і генетичних алгоритмів для вирішення завдань синтезу інтелектуальних систем управління. Поєднання цих технологій дозволило синтезувати нейромережеву модель системи управління, яка не тільки уміє бу-

дувати внутрішні уявлення про зовнішній світ, прогнозувати поведінку об'єкту управління, але так само володіє здатністю генерувати оптимальне управління відповідно до вимог еталонної моделі. У численних публікаціях показана ефективність використання нейромережевих структур для регулювання координат одномасових і двомасових електромеханічних систем з негативним в'язким тертям.

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

В Харківському національному університеті радіоелектроніки проводяться значні дослідження, спрямовані на розробку методів управління нелінійними динамічними системами з використанням штучних нейронних мереж [5]. Особлива увага приділяється вдосконаленню технік навчання нейронних мереж, ідентифікації нелінійних нестационарних об'єктів за допомогою нейронних мереж, а також розробці нейромережевих систем управління для таких об'єктів.

Нове тисячоліття відзначилось переходом на субмікронні і нанотехнології, а також успіхи молекулярної і біомолекулярної технології приводять до принципово нових архітектурних і технологічних рішень по створенню нейрокомп'ютерів.

Глибоке дослідження штучних нейронних мереж вимагає знань у таких галузях, як нейрофізіологія, наука про пізнання, психологія, фізика (зокрема статистична механіка), теорія управління, теорія обчислень, штучний інтелект, статистика та математика, розпізнавання образів, комп'ютерне бачення, паралельні обчислення та апаратне забезпечення (цифрове і аналогове). Водночас

штучні нейронні мережі сприяють розвитку цих дисциплін, надаючи нові інструменти та концепції. Такий симбіоз є необхідним для ефективного дослідження 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 - архітектури штучних нейронних 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.

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, в додатку Б – структуру багат шарових нейронних мереж.

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

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

1.2 Аналіз структур систем нейрорегулювання. Вибір структури системи

На сьогоднішній день в спеціальній літературі по нейрорегулюванню часто згадуються терміни „пряме” і „неаряме” регулювання в різних інтерпретаціях. Тому відразу слід конкретизувати обидва терміни і одноманітно застосовувати їх далі в тексті. Загальноприйнято розрізняти методи регулювання прямою і непрямою адаптацією залежно від того, яким чином ведеться підстроювання параметрів регулятора. При прямій адаптації зміна параметрів регулятора проводиться безпосередньо, щоб забезпечити виконання певного критерію якості. При непрямій адаптації спочатку здійснюється ідентифікація параметрів моделі, які потім використовуються для синтезу регулятора. Цей же критерій покладений в основу представленого нижче огляду методів нейрорегулювання.

Ще одна комбінація, що часто вживається в літературі, – „online”, „offline” тренування. Online тренування тісно пов'язане з поняттям „регулювання в реальному часі” (*real-time control*). У системах реального часу введення, обробка і виведення інформації відбувається в межах одного періоду дискретності, величина якого визначається в першу чергу характеристиками об'єкту регулювання, а не наявною обчислювальною потужністю. Аналогічно online тренування відповідає випадку, коли тренування мережі (як мінімум одна епоха) і регулювання, що виконується цією ж мережею, проходять одночасно і в межах одного періоду дискретності.

Проведення online тренування в системах регулювання реального часу уточнюється діаграмою на рис. 1.1 [26]. На рисунку прийняті наступні позначення: y – регульована координата; u – дія на об'єкт; T – період дискретності; t_1 – попередня обробка даних датчиків; t_2 – тренування мережі; t_3 – обчислення і видача дії на об'єкт; m, n – число попередніх значень змінних.

Кількість структур систем нейрорегулювання, запропонованих і випробуваних на сьогоднішній день, дуже велика. Одні системи нейрорегулювання використовують аналітичний закон регулювання і нейромоделі. Нейронна мережа

відновлює при цьому елементи матриць моделі об'єкту. Це так звані структуровані методи регулювання. Проте повністю використовувати весь потенціал апроксимації нейронних мереж дозволяють неструктуровані методи регулювання. У наступних нижче підрозділах будуть стисло розглянуті структури систем нейрорегулювання, актуальні на сьогоднішній день.

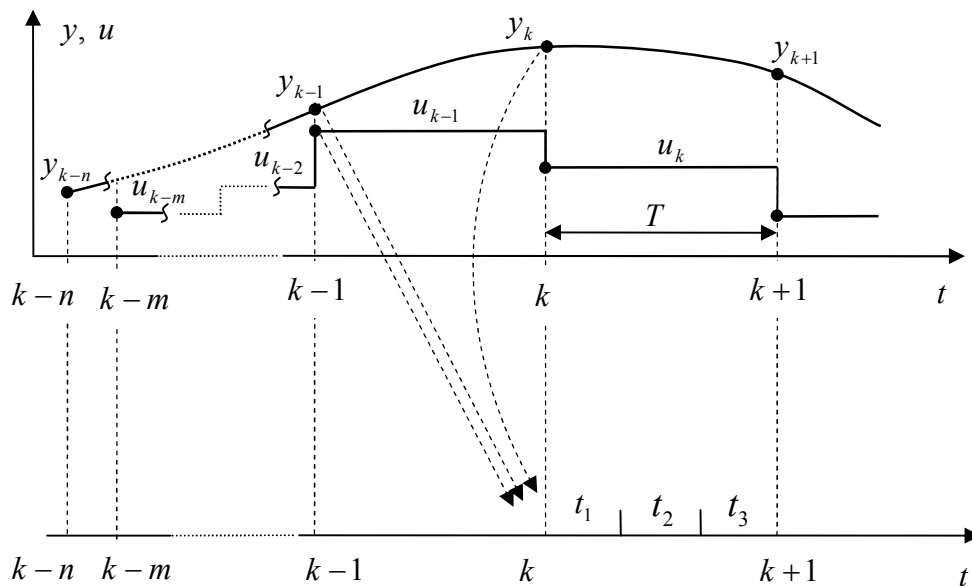


Рисунок. 1.1 – Регулювання і тренування мережі в реальному часі

1.2.1 Пряме інверсне регулювання.

Термін «*Direct inverse control*» був введений Вербосом [27]. Пряме інверсне нейрорегулювання – це метод, найчастіше вживаний на сьогоднішній день (рис. 1.2). Тренування мережі при регулюванні цим методом може проводитися як offline, так і online. Згідно класифікації Псалтиса і ін. [28], ці методи названі «*General learning*» і «*specialized learning*». Відмінності структур прямого інверсного регулювання з offline і online тренуванням розглядаються нижче.

Offline тренування. Пряме інверсне регулювання з offline тренуванням засноване на пропозиції, що інверсне відображення простору виходів в простір входів існує. При цьому спочатку в режимі offline формують модель інверсної динаміки об'єкту регулювання **NC** (перемикачі на рис. 1.2 в нижньому положенні). Потім отримана інверсна модель включається послідовно з об'єктом

(перемикачі на рис. 1.2 у верхньому положенні) і використовується як регулятор. Назва методу "Пряме інверсне регулювання" згідно введеного визначення тут повністю підтверджується.

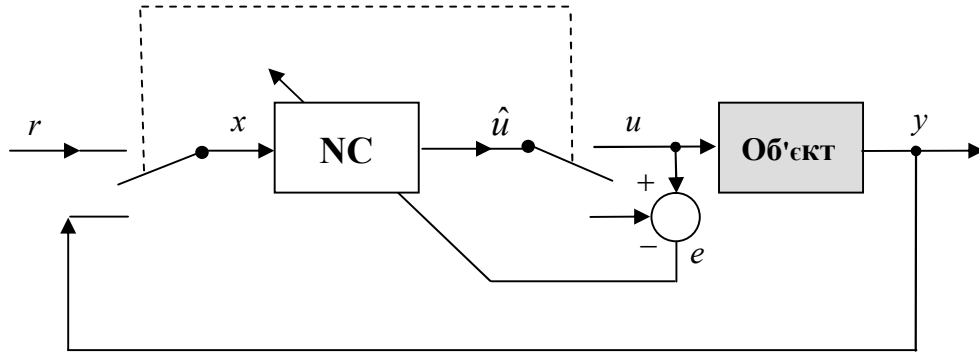


Рисунок 1.2 – Пряме інверсне регулювання з offline тренуванням

Очевидний недолік даної структури регулювання – недостатня робастність через відсутність явного зворотного зв'язку. Тому умова реалізації цієї структури регулювання – стійкість об'єкту регулювання. На практиці ця умова не завжди виконується.

Online тренування. Структура системи прямого інверсного регулювання з тренуванням мережі в режимі online представлена на Рис. 1.3.

Кінцева мета тренування нейрорегулятора **NC** вважається досягнутою, якщо він генерує такий сигнал дії на об'єкт, який приводить до відстежування регульованою координатою y сигналу завдання r . Регулятор навчається безперервно і одночасно з процесом регулювання, тому цей метод застосовний для змінних в часі об'єктів регулювання.

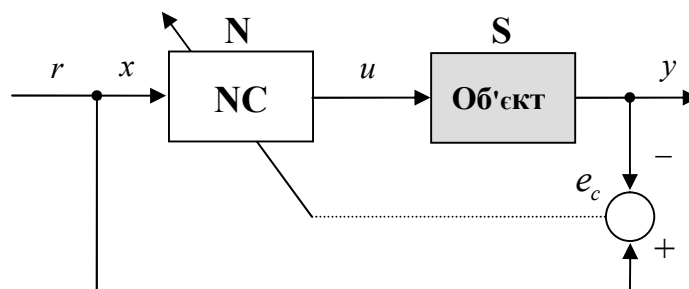


Рисунок 1.3 – Пряме інверсне регулювання з online тренуванням

Істотний недолік тренування мережі в режимі online – необхідність коректування сигналу помилки регулювання e_c оскільки e_c формується на виході об'єкту регулювання, а не на виході регулятора. Пунктирна лінія на рис. 1.2 відображає виникаючий пропуск. Формули обчислення градієнта приведені в таблиці 1.1.

Таблиця 1.1 – Обчислення градієнта при регулюванні з online тренуванням

Вихідний сигнал мережі	$u = \mathbf{N}(x, \mathbf{w})$
Регульована координата	$y = \mathbf{S}(u)$
Помилка	$e_c = r - y$
Градiєнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e_c} \frac{\partial e_c}{\partial \mathbf{w}} = -e_c \frac{\partial y}{\partial \mathbf{w}} = -e_c \frac{\partial y}{\partial u} \frac{\partial u}{\partial \mathbf{w}}$

Похідна $\frac{\partial u}{\partial \mathbf{w}}$ може бути завжди обчислена. Складову $\frac{\partial y}{\partial u}$ називають матрицею Якобі. Для відомих об'єктів регулювання матриця Якобі може бути отримана аналітично, а її елементи – розраховані. На жаль, об'єкт регулювання найчастіше точно невідомий. Приблизно похідна може бути обчислена за допомогою різницевого рівняння [28]:

$$\frac{\partial y}{\partial u} \approx \frac{\Delta y}{\Delta u} = \frac{y(k) - y(k-1)}{u(k) - u(k-1)}.$$

По аналогії з лінійними методами адаптивного регулювання запропоновано також використовувати замість величини похідної її знак [29], якщо про об'єкт відома якісна інформація. Проте при зашумлених даних обидва методи не здатні ефективно вирішити вказану проблему.

Слід зазначити, що регулювання в перший момент часу нетренованим регулятором в розімкненій системі може привести до неконтрольованої поведінки системи. Щоб подолати цей недолік, слід проводити грубе попереднє тренування нейрорегулятора.

1.2.2 Непряме інверсне нейрорегулювання.

У літературі зустрічаються різні назви цього методу регулювання, наприклад, *feedforward inverse control* у Наренда [30] або *specialized inverse learning* у Ханта [31]. Структурна схема системи регулювання представлена на рис. 1.4.

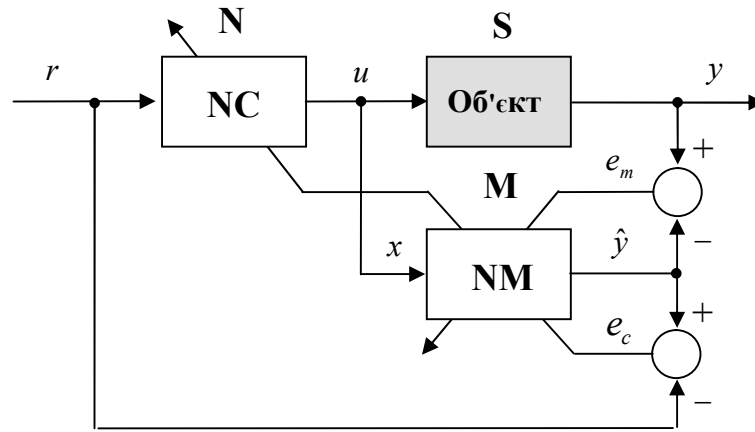


Рисунок 1.4– Непряме інверсне нейрорегулювання

Непряме нейрорегулювання – це двохетапний процес. На першому етапі відбувається тренування нейромоделі об'єкту **NM** у режимі *offline*.. Після успішного тренування на основі нейромоделі у момент часу τ правилом підстановки можна обчислити матрицю Якобі:

$$\left. \frac{\partial y}{\partial u} \right|_{t=\tau} \approx \left. \frac{\partial \hat{y}}{\partial u} \right|_{x(k)} = \frac{\partial y^{(n)}}{\partial n^{(n)}} \frac{\partial n^{(n)}}{\partial a^{(n-1)}} \frac{\partial a^{(n-1)}}{\partial n^{(n-1)}} \dots \frac{\partial y^{(1)}}{\partial n^{(1)}} \frac{\partial n^{(1)}}{\partial x_0},$$

де $n^{(i)}, a^{(i)}$ – зважена сума і вихід i -го шаруючи мережі $1 \leq i \leq L$;

$a^{(L)} = \hat{y}$ – вихід нейромоделі;

$x_0 = u$ – один з входів нейромоделі.

Завдяки цьому помилка e_c може бути використана для підстроювання вагових коефіцієнтів нейрорегулятора **NC** у другій фазі – фазі регулювання.

Проблематичною тут виявляється ситуація, коли нейрорегулятор генерує дію на об'єкт поза діапазоном, в якому тренувалася нейромодель об'єкту. У роботах Наренди і ін. [31] досліджувався варіант непрямого нейрорегулювання,

в якому нейромодель **NM** підстроювалася в режим online. Ця концепція системи регулювання має проблеми забезпечення стійкості в перші моменти часу тренування нейрорегулятора. Явно виражений зворотний зв'язок в системі також відсутній. Ішигуро [32] запропонували ефективнішу структуру „*neural compensator*”. У ній нейронна мережа формує сигнал компенсації паралельно класичному регулятору, а сигнал помилки для тренування мережі формується побічно як різниця між сигналом дії на об'єкт і вихідним сигналом аналітичної моделі об'єкту.

1.2.3 Регулювання з внутрішньою моделлю.

У системі регулювання з внутрішньою моделлю (*Internal model Control*, ІМС) використовується як пряма, так і інверсна модель об'єкту. Схема вперше була запропонована Гарсія і Морарі [33] з аналітичними нелінійними моделями. Застосування нейромоделей може служити яскравим прикладом того, як нейротехнології можуть бути інтегровані в класичні адаптивні системи регулювання. Структурна схема системи представлена на рис. 1.5.

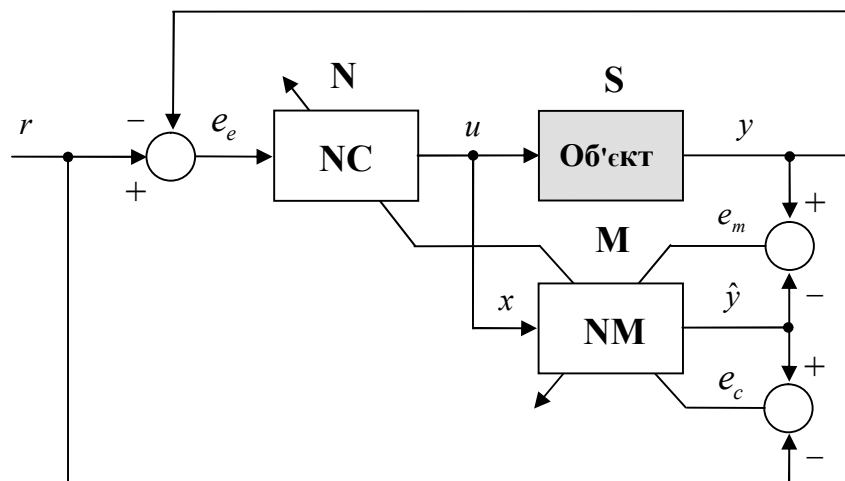


Рисунок 1.5 – Система регулювання з внутрішньою моделлю

Аналогічно непрямому нейрорегулюванню, спочатку здійснюється тренування нейромоделі **NM** у режимі offline. З допомогою **NM** формується матриця Якобі для корекції сигналу помилки e_n щоб забезпечити тренування інве-

рсного нейрорегулятора **НС** у режимі online. На відміну від непрямого регулювання обидві нейронні моделі тут є складовими частинами контура регулювання, оскільки сигнал помилки між моделлю об'єкту і самим об'єктом e_m замикається на вхід нейрорегулятора.

Дана структура має одну цікаву особливість. Якщо модель об'єкту ідеальна ($\mathbf{M} = \mathbf{S}$), то ІМС стає ідентичною непрямому нейрорегулюванню з розімкненим контуром ($e_m = 0$), завдяки чому стає можливим доказ стійкості. Примітно також, що при асимптотично стійких збуреннях і сигналі завдання відсутня статична помилка регулювання ($y = r$) за умови, що нейрорегулятор є інверсною ланкою моделі об'єкту ($\mathbf{N} = \mathbf{M}^{-1}$). Якість моделі об'єкту при цьому – другорядна. Це витікає з наступних виразів:

$$\hat{y} = r_e \cdot \mathbf{N} \cdot \mathbf{M}; \quad r_e = r - y + \hat{y}.$$

Метод ІМС застосовний, проте, тільки до стійких об'єктів регулювання. Щоб підтримувати стійкість контура регулювання зазвичай потрібний фільтр низьких частот на вході нейрорегулятора, оскільки в реальних умовах ідеальні моделі практично не реалізуються. В принципі, дана концепція регулювання має ті ж слабкі місця, що і непряме інверсне нейрорегулювання. В порівнянні з класичними ІМС нейронні мережі нічого не додають окрім спрощення потрібній ідентифікації нелінійних динамічних систем.

1.2.4 Feedforward-Feedback-Error регулювання.

Дана концепція регулювання, в принципі, теж відображає структуру прямого інверсного регулювання. Як видно з рис. 1.6, в систему додатково інтегровані ще один зворотний зв'язок і третя нейронна мережа – *Feedback-Error-регулятор*.

Обидва нейрорегулятора $\mathbf{NC}_{FF}$ і $\mathbf{NC}_{FB}$ тренуються спочатку в режимі offline роздільно і з різними функціоналами якості. Модель об'єкту $\mathbf{NM}$ повинна бути також натренована в режимі offline. Після того, як модель об'єкту $\mathbf{NM}$

отримана, процес регулювання може бути початий, а помилка e_c скоректована за допомогою прямої нейромоделі для отримання сигналу підстроювання нейрорегуляторів. NC_{FF} виконує роль нейронного компенсатора в прямому каналі, як і при непрямому інверсному регулюванні. За рахунок Feedback-Error-регулятора NC_{FB} утворюється замкнутий контур, що позитивно впливає на якість регулювання. Даний метод регулювання застосовано, наприклад в [34,35].

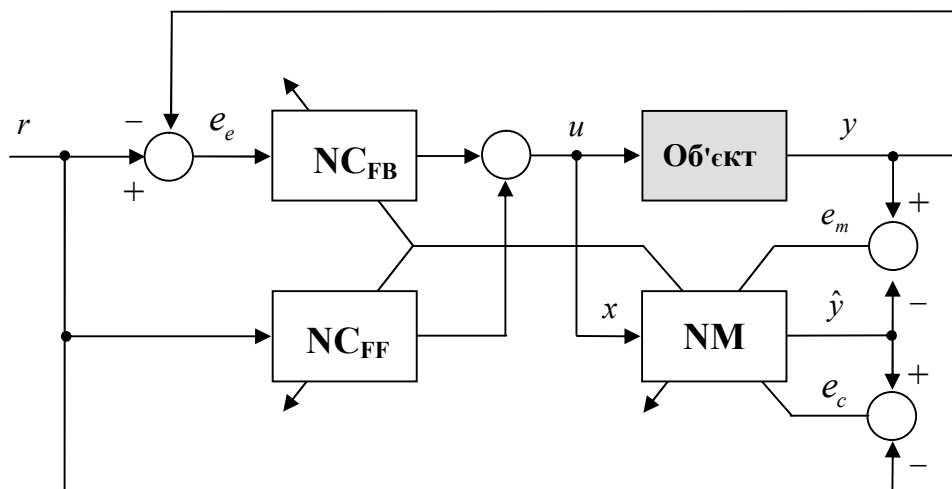


Рисунок 1.6 – Feedforward-Feedback-Error-регулятор

1.2.5 Регулювання з Feedback-Error-Learning.

На рис. 1.7 представлена структура системи регулювання з «*Feedback-Error-Learning*» (FEL) Кавато і ін. [36]. По суті, дана структура аналогічна прямому інверсному регулюванню з класичним, наприклад, ПД-регулятором.

Проте, на відміну від всіх раніше згаданих прямих методів нейрорегулювання, метод FEL використовує вихідний сигнал ПД-регулятора як сигнал помилки, що вимагає мінімізації e_c для підстроювання вагових коефіцієнтів нейрорегулятора. Завдяки цьому відпадає необхідність коректувати помилку за допомогою моделі об'єкту. Іншими словами, тут використовується інший функціонал якості з метою відмови від обчислення матриці Якобі. В результаті online тренування нейрорегулятор NC все більше і більше замінює ПД-регулятор. У результаті $e_c = 0$, що означає, що об'єкт регулюється тільки за до-

помогою нейрорегулятора, що є інверсне об'єкту регулювання ланка. Випадкові відхилення регульованої координати можуть при цьому коректуватися за допомогою ПД-регулятора. Крім того, ПД-регулятор забезпечує стійкість системи в початковій фазі тренування мережі.

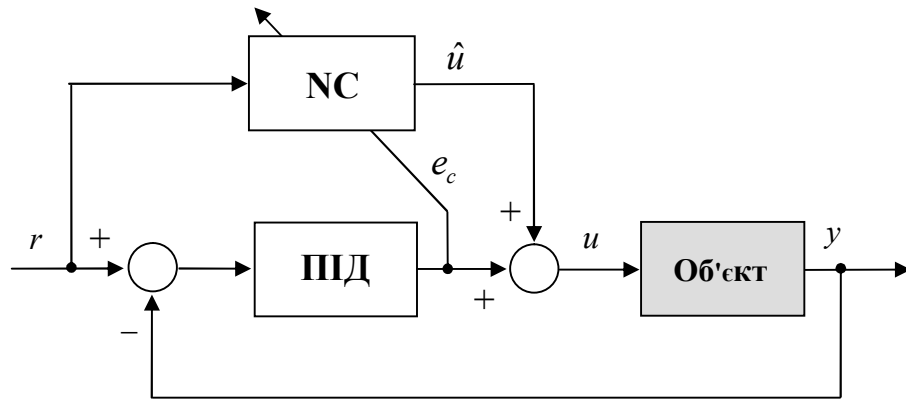


Рисунок 1.7 – Регулювання з Feedback-Error-Learning

До недоліків FEL регулювання відноситься сильна залежність швидкості збіжності від параметрів ПД-регулятора [37]. Слід зазначити, що твердження про те, що нейрорегулятор є інверсною об'єкту ланкою, не буде справедливе, якщо тренувальні дані не є глобальними. Вагові коефіцієнти слід повторно підстроювати, якщо завдання лежатиме в іншому робочому діапазоні.

1.2.6 Регулювання методом компенсації завдання.

Регулювання методом компенсації завдання (*Reference Compensation Technique*, RCT), структурна схема якого представлена на рис. 1.8, була запропонована Юнгом і Хсія [38].

Як і при регулюванні FEL тут також не проводиться обчислення матриці Якобі завдяки тому, що для тренування нейросети використовується спеціально вибраний сигнал помилки e_c (див. таблицю 1.2).

Особливість методу RCT в тому, що компенсація наявних нелінійностей здійснюється на рівні сигналів завдання, а не сигналів дії на об'єкт. Перевага даної техніки – простота вбудовування в готовий контур регулювання. Склад-

ність методу полягає у виборі початкових значень параметрів мережі, оскільки інакше можливе формування небезпечних сигналів завдання.

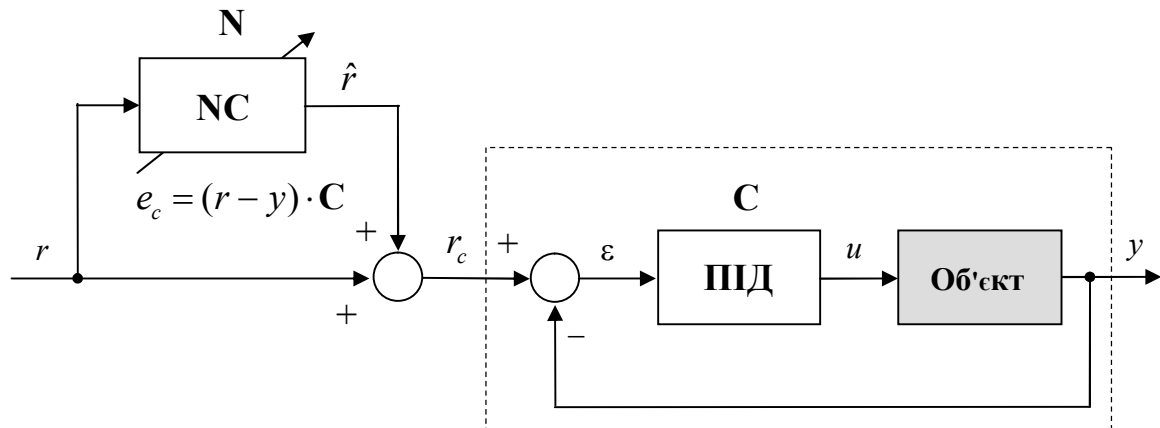


Рисунок 1.8 – Регулювання методом компенсації завдання

Таблиця 1.2 – Обчислення градієнта при FEL регулюванні

Вихідний сигнал мережі	$\hat{r} = \mathbf{N}(r, \mathbf{w})$
Помилка	$e_c = (r - y) \cdot \mathbf{C} = u - \hat{r} \cdot \mathbf{C}$
Градiєнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e_c} \frac{\partial e_c}{\partial \mathbf{w}} = -e_c \frac{\partial \hat{r}}{\partial \mathbf{w}} \mathbf{C}$

1.2.7 Instantaneous linearization.

Головна ідея цього методу полягає в лінеаризації нейронної моделі об'єкту в поточній робочій точці [39].

Хай, наприклад, нелінійний динамічний об'єкт

$$y(k) = \phi(y(k-1) \dots y(k-n), u(k) \dots u(k-m))$$

моделюється за допомогою NARX-нейромоделі

$$\hat{y}(k) = \mathbf{M}(\mathbf{x}(k), \mathbf{w}),$$

де $\mathbf{x}(k) = [y(k-1) \dots y(k-n_y), u(k-n_k) \dots u(k-n_u-n_k+1)]^T$ – вектор стану;

$\mathbf{w}$ – вагові коефіцієнти мережі.

Цей об'єкт може бути лінеаризован в момент $k = \kappa$ наступним чином:

$$\hat{y}(k)|_{k=\kappa} = y(k-1) - a_1 \Delta y(k-1) \dots - a_{n_y} \Delta y(k-n_y) + \\ + b_0 \Delta u(k-n_k) + b_{n_u} \Delta u(k-n_u-n_k+1),$$

де

$$a_i = - \left. \frac{\partial \mathbf{M}(\mathbf{x}(k), \mathbf{w})}{\partial y(k-i)} \right|_{k=\kappa}, \quad b_j = - \left. \frac{\partial \mathbf{M}(\mathbf{x}(k), \mathbf{w})}{\partial u(k-j)} \right|_{k=\kappa}, \quad i = 1 \dots n_y, \quad j = 0 \dots n_u + n_k - 1,$$

$$\Delta y(k-i) = y(k-i) - y(k-i-1), \quad \Delta u(k-j) = u(k-j) - u(k-j-1).$$

Приватні похідні можуть бути обчислені з нейромоделі **NM** за допомогою вже відомого правила підстановки. Звідси один крок до синтезу лінійного регулятора. Концепція регулювання представлена на рис. 1.9 і аналогічна лінійним методам регулювання з непрямою адаптацією.

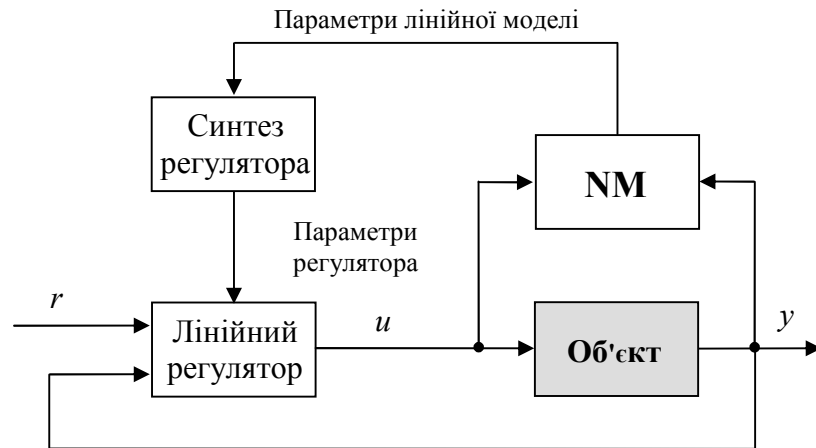


Рисунок 1.9 – Instantaneous linearization і синтез лінійного регулятора

1.2.8 Нелінійне предиктивне регулювання.

Принцип предиктивного методу регулювання на основі моделі (*model predictive control*, MPC) 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. 1.10.

Регулятор в даному випадку є не що інше, як ітераційний алгоритм оптимізації [40]:

- 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 MPC – 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 роботі використовуємо нелінійне предиктивне регулювання.

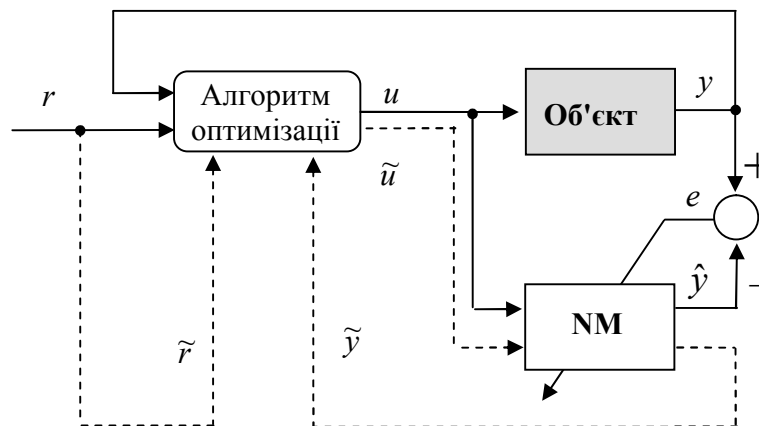


Рисунок 1.10 – Предиктивне регулювання

1.3 Вимоги до систем автоматизації і електроприводів шахтних підйомних машин

Вимоги до системи автоматичного управління. Система автоматичного управління повинна базуватися на принципі підлеглого регулювання параметрів електроприводу, з основним завданням – програмне управління швидкістю. Вона повинна забезпечувати можливість встановлення кінематичних параметрів

рів у межах: швидкість $V_{\max}$ від 5 до 20 м/с, прискорення $a_{\max}$ від 0,5 до 1 м/с², ривок ρ від 0,5 до 2,5 м/с³, при цьому похибка не повинна перевищувати $\pm 1\%$.

Система автоматичного управління швидкістю приводу повинна забезпечувати: статичну точність підтримки не гірше 1,0%, а при використанні точних тахогенераторів – не більше 0,5%; обмеження струму в межах $(1,6 \div 2)I_{\text{стн}}$ на заданому рівні з точністю до 10%; час відгуку при ступінчастій зміні статичного струму $I_{\text{ст}} = (1 \div 1,2)I_{\text{стн}}$ не більше 1 с без суттєвого перерегулювання.

Система управління повинна зберігати зазначену точність управління при зміні напруги живильної мережі змінного струму, зміні умов навколишнього середовища, температури повітря та тепловому дрейфі нуля.

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 погрішністю 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 прискорення (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 Постановка задачі синтезу нейромережевої системи управління

В результаті проведеного аналізу літератури можна зробити висновок, що існує велика кількість підходів до проблеми синтезу систем управління, при цьому універсального і ідеального регулятора поки не існує. При синтезі систе-

ми управління треба враховувати той фактор, що для електроприводу механізму розвороту екскаватора характерно наявність пружно пов'язаних з платформою інерційних мас стріли ротора з роторним колесом і консолі противаги. Це призводить до того, що перехідні процеси мають значні коливання. Традиційні регулятори не в змозі забезпечити задовільні показники якості функціонування системи.

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

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

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

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

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

Сучасні системи управління нелінійними динамічними об'єктами все частіше використовують штучні нейронні мережі, такі як багат шаровий персепт-

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

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

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

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

Для досягнення поставленої мети необхідно вирішити наступні завдання. задач.

1. Виконати вибір системи електроприводу механізмом підйому шахтної-підйимальної 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 застосування методів нейронних мереж для розв'язання задач управління електромеханічними системами показав, що використання штучних нейронних мереж дозволяє створити нейромережеву модель системи управління, яка не тільки формує внутрішні уявлення про зовнішнє середовище та прогнозує поведінку об'єкта управління, але й здатна генерувати оптимальні управлінські рішення відповідно до еталонної моделі. Особлива увага приділяється розвитку методів навчання нейронних мереж, ідентифікації нелінійних нестационарних об'єктів за допомогою нейронних мереж і синтезу нейромережевих систем управління нелінійними об'єктами.

2. Здатність штучних нейронних мереж до самонавчання дозволяє використовувати нейрорегулятори навіть в умовах значних невизначеностей, у той час як традиційні методи адаптивного управління потребують значного обсягу попередньої інформації про об'єкт управління. Простота реалізації нейронних мереж та їх здатність до навчання роблять їх особливо привабливими для управління складними нелінійними об'єктами в реальному часі.

3. Дослідження показали ефективність застосування багат шарового персептрону для створення нейромережевої моделі об'єкта управління. Багат шаровий персептрон є універсальним аппроксиматором і класифікатором, що є одним із найпоширеніших типів нейронних мереж у сучасній практиці. Обговорено методи навчання нейронних мереж та обрано відповідний метод для навчання нейронної мережі регулятора.

4. Проведено аналіз сучасних підходів до опису нелінійних динамічних об'єктів та розглянуто принципи побудови нейромережевих моделей типу NARMAX і NARX. Моделі NARX мають лінійну параметризацію, що дозволяє застосовувати стандартні методи навчання мереж. Саме тому цей тип моделей часто використовується для нейронної ідентифікації систем.

5. Для оптимізації роботи об'єкта необхідно корегувати параметри регуляторів відповідно до зміни умов роботи. Тому доцільно розробляти системи

управління на основі адаптивного підходу, інтегрованого з методами теорії штучних нейронних мереж (ШНМ). Це дозволить створити прості нейромереві моделі і одночасно адаптивно налаштовувати параметри як моделей, так і регулятора відповідно до змінюваних умов. У результаті аналізу сучасних систем нейрорегулювання було обрано принцип нелінійного предиктивного регулювання для побудови нейромережевого регулятора системи.

6. Проведено аналіз основних вимог до систем автоматизації та електроприводів шахтних підйомних машин. Показано, що через кінцеву жорсткість підйомного канату традиційні регулятори не можуть забезпечити належну якість функціонування системи. Тому недоліки класичних підходів до синтезу таких систем можна значною мірою компенсувати завдяки використанню нейромеревих технологій.

7. Сформульовано задачу дослідження, яка полягає у розробці нейромережевої системи управління механізмом підйому шахтної підйомної установки з урахуванням пружних властивостей підйомного 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
Вантажопідйомність судин $Q_{\text{п}}$	кг	$15,2 \cdot 10^3$
Продуктивність A	т/год	600
Висота підйому, H	м	580
Маса порожнього скипа $m_{\text{ск}}$	кг	$9,8 \cdot 10^3$
Діаметр барабана $D_{\text{б}}$	м	5
Шлях дотягування $S_{\text{дот}}$	м	2
Швидкість дотягування $V_{\text{дот}}$	м/с	0,5
Маса погонного метра каната $m_{1\text{к}}$	кг/м	11,6
Тривалість паузи між циклами $t_{\text{п}}$	с	11

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 шахтної підйимальної 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 клітях 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 здійсненним системам при-

воду. 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 ТП-Д.

2.3 Розрахунок швидкісної і навантажувальної діаграм електроприводу

Вибір основних елементів, розрахунок характеристик і параметрів електроприводу проводиться по методиці [52], яка в даний час є однією з найбільш довершених методик розрахунку тиристорного електроприводу підйомної установки.

Розрахунок параметрів режиму роботи підйомної установки виконується в такій послідовності.

Визначаємо тривалість циклу руху підйомних судин, с

$$T_{\text{ц}} = \frac{Q_{\text{п}} \cdot 3600}{A},$$

де $Q_{\text{п}}$ – вантажопідйомність судин, т;

A – задана продуктивність, т/год

$$T_{\text{ц}} = \frac{15,2 \cdot 3600}{600} = 91,1 \text{ с}.$$

Час руху підйомних судин

$$T = T_{\text{ц}} - t_{\text{п}};$$

де $t_{\text{п}}$ – тривалість паузи між циклами підйому (час завантаження скипа), с.

$$T = 91,2 - 11 = 80,1 \text{ с}.$$

Інтервал часу руху із швидкістю дотягування (час руху в розвантажувальних кривих), с

$$t_{\text{дот}} = \frac{S_{\text{дот}}}{V_{\text{дот}}};$$

$$t_{\text{дот}} = \frac{2}{0,5} = 4 \text{ с}.$$

Значення відносних приведених мас трьохмасової системи визначаються згідно рис 1.2 [52]

$$\mu_1 = 0,73, \quad \mu_2 = 0,2, \quad \mu_k = 0,07.$$

Час руху з обмеженнями ривка рівний

$$t_p = \frac{2\pi H}{4200} \sqrt{\frac{\mu_1 \mu_2}{\mu_k}};$$

де H – висота підйому, $м$.

$$t_p = \frac{2 \cdot 3,14 \cdot 580}{4200} \sqrt{\frac{0,73 \cdot 0,2}{0,07}} = 1,3 \text{ с}.$$

Розрахунковий час руху

$$T_p = T - t_{\text{дот}},$$

$$T_p = 80,1 - 4 = 76,1 \text{ с}.$$

Розрахункова висота підйому

$$H_p = H - V_{\text{дот}} \left(\frac{t_p}{2} + t_{\text{дот}} \right)$$

$$H_p = 580 - 0,5 \left(\frac{1,3}{2} + 4 \right) = 578 \text{ м}.$$

Величина розрахункової середньої швидкості

$$V_{\text{cp}} = \frac{H_p}{T_p},$$

$$V_{\text{cp}} = \frac{578}{76,1} = 7,6 \text{ м / с}.$$

Базова середня швидкість

$$V_6 = V_{\text{cp}} \sqrt{\frac{1000}{H_p}}$$

$$V_6 = 7,6 \sqrt{\frac{1000}{578}} = 10,1 \text{ м / с} .$$

Дана підйомна установка є урівноваженою, тобто коефіцієнт неурівноваженості $\xi = 1$. З графіків рис. 1.10 [52] визначаємо значення множника швидкості $\lambda = 1,21$.

Максимальна швидкість

$$V_{\text{max}} = \lambda \cdot \frac{H_p}{T_p} ,$$

$$V_{\text{max}} = 1,21 \frac{578}{76,1} = 9,2 \text{ м / с} .$$

Максимальне прискорення

$$a_{\text{max}} = \frac{\lambda \cdot \frac{H_p}{T_p^2}}{1 - \frac{1}{\lambda} - \frac{t_p}{T_p}} ,$$

$$a_{\text{max}} = \frac{1,21 \cdot \frac{578}{76,1^2}}{1 - \frac{1}{1,21} - \frac{1,3}{76,1}} = 0,87 \text{ м / с}^2 .$$

Час руху з максимальним прискоренням в період розгону

$$t_a = T_p \cdot \left(1 - \frac{1}{\lambda} - 2 \cdot \frac{t_p}{T_p} \right) ,$$

$$t_a = 76,1 \left(1 - \frac{1}{1,21} - 2 \frac{1,3}{76,1} \right) = 9,84 \text{ с}.$$

Час руху з максимальною швидкістю

$$t_v = T_p \cdot \left(\frac{2}{\lambda} - 1 \right),$$

$$t_v = 76,1 \left(\frac{2}{1,21} - 1 \right) = 50,2 \text{ с}.$$

Час руху за рахунок механічного гальма

$$t_r = \frac{V_{\text{дог}}}{a_r},$$

де a_r – уповільнення, приймається рівним a_{max}

$$t_r = \frac{0,5}{0,87} = 0,57 \text{ с}.$$

Час руху з максимальним прискоренням в період уповільнення

$$t_{a1} = t_a - t_r,$$

$$t_{a1} = 9,84 - 0,57 = 9,27 \text{ с}$$

Сумарна приведена маса визначається з виразу

$$m = \frac{Q_{\text{п}} + m_{\text{ск}} + m_{\text{лк}} \cdot H}{\mu_{\text{к}} + \mu_2},$$

$$m = \frac{15200 + 9800 + 11,6 \cdot 580}{0,07 + 0,2} = 116792 \text{ кг}$$

Значення рушійних зусиль у момент часу зміни кінематичних параметрів

$$F_0 = q(k\zeta Q + m_{1k}H),$$

$$F_0 = 9,81 \cdot (1,1 \cdot 1 \cdot 15200 + 11,6 \cdot 580) = 211874 \text{ кг}$$

де k – коефіцієнт, що враховує додаткові опори руху, згідно [52] $k = 1,1 \div 1,2$.

$$F_1 = F_0 - gm_{1k} \frac{a_{\max} t_p^2}{3} + ma_{\max},$$

$$F_1 = 211874 - 9,81 \cdot 11,6 \frac{0,83 \cdot 1,3^2}{3} + 116792 \cdot 0,87 = 292708 \text{ Н}$$

$$F_2 = F_0 - qm_{1k} \left(\frac{a_{\max} t_p^2}{3} + V_{\max} t_a \right) + ma_{\max}$$

$$F_2 = 211874 - 9,81 \cdot 11,6 \left(\frac{0,87 \cdot 1,3^2}{3} + 9,2 \cdot 9,84 \right) + 116792 \cdot 0,87 = 287236 \text{ Н}$$

$$F_3 = F_0 - qm_{1k} V_{\max} (t_a + 2t_p),$$

$$F_3 = 211874 - 9,81 \cdot 11,6 \cdot 9,38 (9,87 + 2 \cdot 1,3) = 199834 \text{ Н}$$

$$F_4 = F_0 - qm_{1k} V_{\max} (t_a + 2t_p + 2t_v),$$

$$F_4 = 211874 - 9,81 \cdot 11,6 \cdot 9,38 \cdot (9,84 + 2 \cdot 1,3 + 2 \cdot 50,2) = 91408 \text{ Н}$$

$$F_5 = F_0 - qm_{1k} V_{\max} \left(t_a + 4t_p + 2t_v - \frac{a_{\max} t_p^2}{3V_{\max}} \right) - ma_{\max},$$

$$F_5 = 211874 - 9,81 \cdot 11,6 \cdot 9,2 \cdot \left(9,84 + 4 \cdot 1,3 + 2 \cdot 50,2 - \frac{0,87 \cdot 1,3^2}{3 \cdot 9,2} \right) - 116792 \cdot 0,87 = 3512 \text{ Н}$$

$$F_6 = F_0 - qm_{\text{лк}}V_{\text{max}} \left(2t_a + 4t_p + 2t_v - \frac{a_{\text{max}}t_p^2}{3V_{\text{max}}} \right) - ma_{\text{max}} ,$$

$$F_6 = 211874 - 9,81 \cdot 11,6 \cdot 9,2 \cdot \left(2 \cdot 9,84 + 4 \cdot 1,3 + 2 \cdot 50,2 - \frac{0,87 \cdot 1,3^2}{3 \cdot 9,2} \right) -$$

$$-116792 \cdot 0,87 = -6571 \text{ H}$$

$$F_7 = F_0 - 2qm_{\text{лк}}$$

$$F_7 = 211874 - 2 \cdot 9,81 \cdot 11,6 \cdot 580 = 77018 \text{ H}$$

Швидкісна і навантаження діаграми, побудовані за наслідками розрахунків, показані на рис. 2.1.

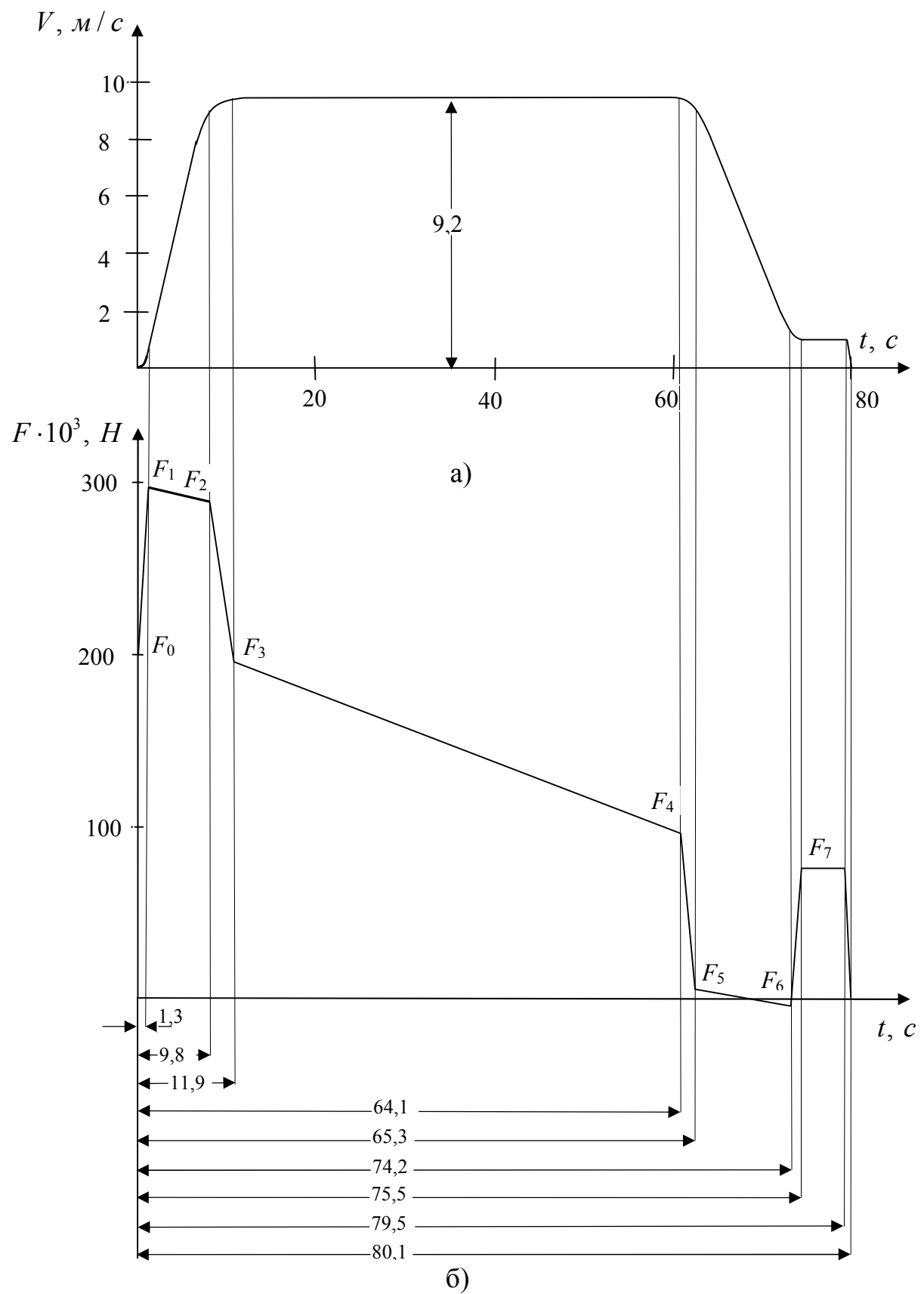


Рисунок 2.1 – Діаграми роботи шахтної підйомної установки: а) швидкісна діаграма; б) діаграма навантаження.

2.4 Вибір електродвигуна по потужності

По графіках рис. 1.10 [52] при, $\lambda = 1,21$ і $V_6 = 10,1 \text{ м / с}$ визначаємо відносну базову потужність

$$P_{\text{еф.б}}^* = 1,89.$$

Відносна ефективна потужність визначається по формулі

$$P_{\text{еф}}^* = [P_{\text{еф.б}}^* - 1,17\lambda] H \cdot 10^{-3} + 1,17\lambda,$$

$$P_{\text{еф}}^* = (1,69 - 1,17 \cdot 1,2) \cdot 580 \cdot 10^{-3} + 1,7 \cdot 1,21 = 1,57.$$

Потужність двигуна обчислюється згідно виразу

$$P_{\text{дв}} = k_{\text{пн}} k_{\text{оп}} P_{\text{еф}}^* \cdot \frac{q \cdot Q_{\text{п}} H}{1040 \cdot T_{\text{р}}},$$

де $k_{\text{пн}} = 1,05$ – коефіцієнт, що враховує пульсуючий характер напруги;

$k_{\text{оп}} = 1,1$ – коефіцієнт, що враховує можливий режим роботи з ослабленням магнітного потоку збудження.

$$P_{\text{дв}} = 1,05 \cdot 1,1 \cdot 1,57 \frac{9,81 \cdot 15200 \cdot 580}{1040 \cdot 76,1} = 1703 \text{ кВт}.$$

Частота обертання електродвигуна

$$n_{\text{дв}} = \frac{30 \cdot V_{\text{max}}}{\pi R_6},$$

$$n_{\text{дв}} = \frac{30 \cdot 9,2}{3,14 \cdot 2,5} = 35,16 \text{ об / хв}.$$

По каталогу вибираємо двигун П2-23/85-2,8УХЛ4. Технічні дані двигуна: номінальна потужність $P_{\text{н}} = 1800 \text{ кВт}$, номінальна частота обертання

$n_n = 32$ об/хв, номінальна напруга $U_n = 600$ В, номінальний струм $I_n = 3180$ А, опір обмотки якоря $R_{\text{я}} = 0,00754$ Ом, опір обмотки додаткових полюсів $R_{\text{дп}} = 0,00114$ Ом, опір компенсаційної обмотки $R_{\text{ко}} = 0,000405$ Ом, число пар полюсів $p = 5$.

Експлуатаційне перевантаження двигуна визначається по виразу

$$\gamma_{\text{експ}} = \frac{(k\xi Q_{\text{п}} + m_{1\text{к}} H)q + ma_{\text{max}}}{F_{\text{дв н}}},$$

де $F_{\text{дв н}}$ – номінальне рушійні зусилля

$$F_{\text{дв н}} = \frac{30P_n}{\pi n_n R_{\text{б}}},$$

$$F_{\text{дв н}} = \frac{30 \cdot 1800 \cdot 10^3}{3,14 \cdot 32 \cdot 2,5} = 201242 \text{ Н}.$$

$$\gamma_{\text{експ}} = \frac{(1,1 \cdot 1 \cdot 15200 + 11,6 \cdot 580) \cdot 9,81 + 116792 \cdot 0,87}{201242} = 1,86.$$

Екстрене перевантаження, що виникає при перестановці барабанів і зміні канатів

$$\gamma_{\text{екст}} = \frac{(Q + m_{\text{ск}} + m_{1\text{к}} H)q}{F_{\text{дв.ном}}},$$

$$\gamma_{\text{екст}} = \frac{(15200 + 9800 + 11,6 \cdot 580) \cdot 9,81}{201242} = 1,71.$$

Таким чином, прийнятий електродвигун по перевантажувальній здатності відповідає вимогам.

2.5 Вибір основного електроустаткування силового кола електроприводу

Для живлення двигуна вибираємо тиристорний перетворювальний агрегат ТПРЗ-4000/660Т-30/ОУ4. Номінальна випрямлена напруга $U_{дн} = 660 \text{ В}$, номінальний випрямлений струм $I_{дн} = 4000 \text{ А}$.

Агрегат реверсивний призначений для тривалого режиму роботи і допускає роботу в повторно-короткочасному режимі. Перевантаження по струму в циклічному режимі дворазове впродовж не більше 15 с. Діапазон регулювання напруги для реверсивних агрегатів $\pm 100\%$. Агрегат виконаний зі зустрічно-паралельним включенням випрямних груп з роздільним управлінням. Використовується шестифазна еквівалентна схема з паралельним з'єднанням мостів.

До складу агрегату входять: перетворювальні секції і щити управління, силовий живлячий трансформатор, згладжуючі реактори, швидкодіючий автоматичний вимикач постійного струму, трансформатори струму і датчики струму на герконах.

Силова частина тиристорного агрегату є набором секцій тиристорів. У секцію входять шафа введення і дві силові шафи. Шафа введення містить елементи зв'язку з трансформатором і двигуном. Силова шафа представляє собою трифазний випрямний міст на номінальний струм 2500 А і є уніфікованим силовим модулем всієї серії агрегатів.

Система імпульсний-фазового управління побудована за вертикальним принципом. Опорний сигнал – синусоїдальний, пов'язаний з мережею. Діапазон зміни фази відмикаючих імпульсів 9-165°. Система управління допускає роботу ТП в двох режимах: одночасного (симетричного) і почергового управління мостовими схемами.

До складу СІФУ входять функціональні блоки: фазового управління БФУ; розмноження імпульсів БРІ; підсилювачів імпульсів БУІ; управління режимами БУР, резисторів, живлення БП1 і БП2; зсуви; регулятора струму БРТ.

Структурна схема реверсивного перетворювального агрегату приведена на рис. 2.2.

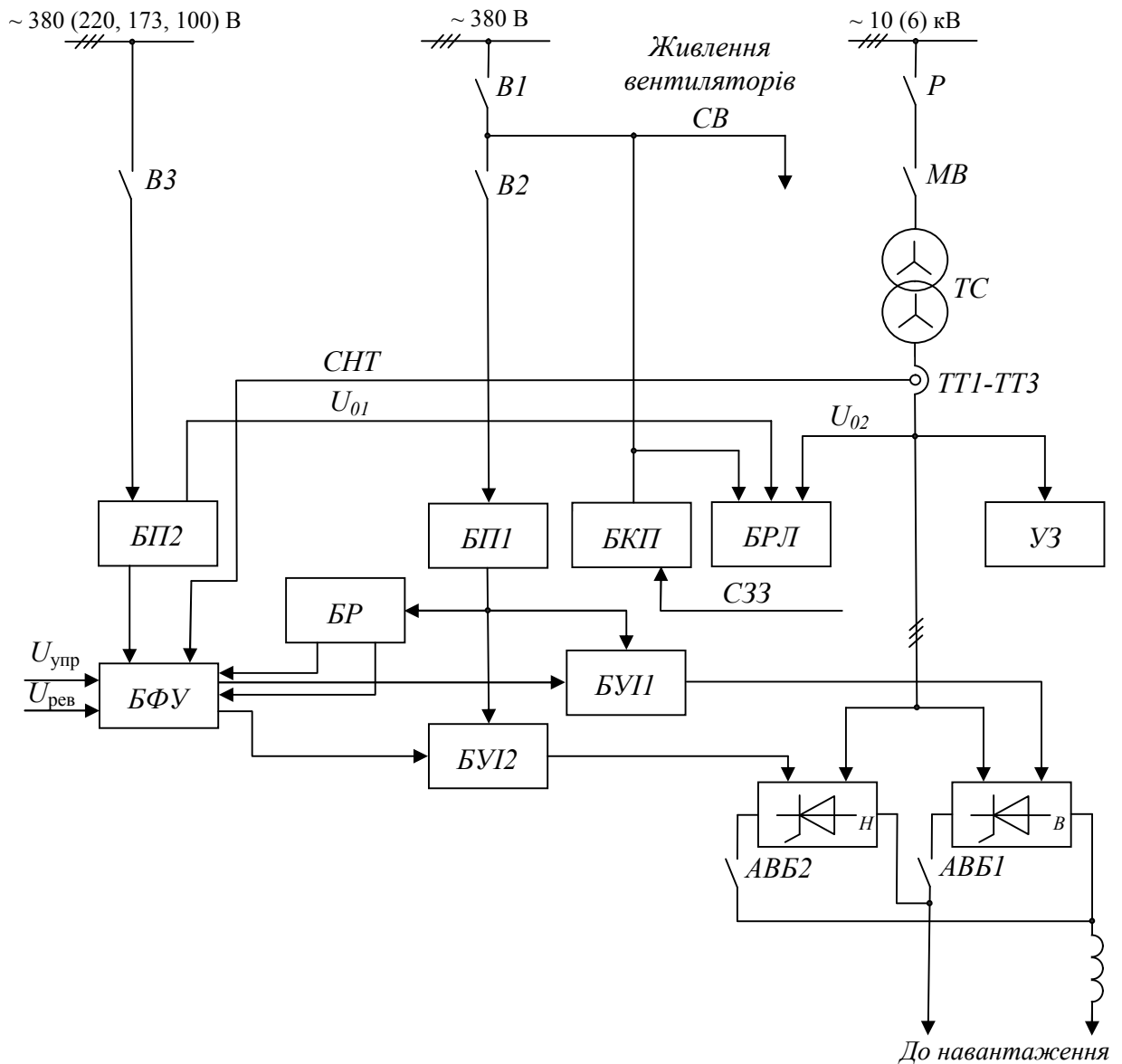


Рисунок 2.2 – Структурна схема реверсивного перетворювального агрегату: Р – роз'єднувач; ТТ1-ТТ3 – трансформатори струму; СНТ – сигнал наявності струму; СЗЗ – сигнал згорання запобіжника у вентильних секціях; U_{01} – контроль опорної напруги; U_{02} – контроль силової напруги.

Силове живлення на випрямні мости В і Н подається через силовий трансформатор ТС від мережі 6 (10) кВ. навантаження підключається до агрегату через швидкодіючі автоматичні вимикачі АВБ1, АВБ2 і згладжуючий реактор і

ін. Оперативна напруга 380В подається через вимикач В1 до вентиляторів силових секцій і через В1 і В2 – до блоку БП1, живлячому БУІ1, БУІ2. Опорна напруга подається через автомат В3 на блок БП2, призначений для живлення блоку БФУ. При появі на вході БУ управляючої напруги $U_{упр}$ у блоці виробляються імпульси управління. Від БФУ імпульси подаються в блоки підсилювачів імпульсів БУІ1 і БУІ2, де вони посилюються до потужності, необхідної для надійного відкриття всіх тиристорів агрегату. У блоці БР розміщені резистори, включені в колекторне навантаження транзисторів осередків підсилювачів імпульсів і призначені для обмеження струму блоків БУІ1 або БУІ2.

Контроль наявності силової, оперативної і опорної напруги, стани масляного вимикача МВ і вимикачів АВБ1, АВБ2 здійснюється блоком БРЛ. Сигналізація стану запобіжників у вентиляльних секціях здійснюється блоком контролю запобіжників БКП. Захист тиристорів агрегату від перенапружень, що виникають при відключенні силового трансформатора на холостому ході, здійснюється вузлом захисту ВЗ, що складається з двох блоків вентилів і блоку конденсаторів.

Для живлення тиристорних агрегатів використовуються трансформатори з перемиканням відгалужень мережевих обмоток без збудження (ПБВ) при відключенні всіх обмоток трансформатора від мережі; з регулюванням напруги під навантаженням – трансформатори з РПН.

Трансформатори з ПБВ перемиканням відгалужень мережевих обмоток регулюють напругу на $\pm 5\%$ для компенсації напруги живлячій мережі. Трансформатори з РПН здійснюють ступінчасте регулювання напруги під навантаженням на 40 % нижче за номінальне значення, число ступенів не меншого 10. Трансформатори з РПН використовуються для регулювання максимальної напруги ТП при зміні V_{max} на тривалий час, а також в тих випадках, коли номінальна напруга двигуна не відповідає номінальній напрузі ТП.

За способом охолодження застосовуються масляні двообмотувальні трансформатори ТПМ і з негорючим заповненням – совтолові трансформатори ТПП. Розширюється номенклатура сухих трансформаторів ТСЗП нової серії, які можуть бути використані натомість совтолових.

Для дванадцятипульсних ТП використовуються масляні трансформатори з двома активними частинами – ТМПД, ДТПД, ТМНПД, ТДНПД. У трансформаторах з двома активними частинами кожна вентиляна обмотка розрахована або на повну напругу і половину номінального струму ТП (2-2000; 2-2500; 2-3150А) для можливості паралельного з'єднання випрямних мостів, або на половинну напругу (2-525; 2-412,5В) і повний струм агрегату для можливості послідовного з'єднання випрямних мостів.

Для вибраного тиристорного агрегату згідно рекомендаціям [51] вибираємо силовий трансформатор ТМПД-3200/10У2. Трансформатор виконаний з двома активними частинами. Кожна вентиляна обмотка розрахована на повну напругу 660 В і половину номінального струму ТП 2×2000 А. Напруга мережевої обмотки 6; 10 кВ. Схема з'єднання обмоток: мережевої Y і Δ ; вентиляною Δ і Δ . Втрати холостого ходу $P_{xx} = 6300$ Вт; втрати короткого замикання $P_{кз} = 34150$ Вт; ток холостого ходу $I_{xx} = 1,2\% = 1,2\%$. Міжфазна напруга вентиляної обмотки $U_2 = 570$ В, напруга короткого замикання $U_{кз} = 6,2\%$.

Для згладжування пульсацій випрямленого струму в головному ланцюзі приводу за системою ТП-Д застосовуються реактори СРОСЗ. Проте в агрегатах з паралельним з'єднанням мостів реактора не потрібний, оскільки при живленні двигунів серії П2 від дванадцятипульсних випрямлячів пульсації I_d не перевищують допустимого значення. У цих агрегатах встановлюються зрівняльні реактори СРОСЗ, які включені в зрівняльний контур. В цьому випадку реактори забезпечують згладжування випрямленого струму якоря електродвигуна, обмеження зрівняльного струму перетворювача і поліпшення використання тиристорів.

Індуктивність реакторів визначається згідно виразу

$$L_{p1} = L_{p2} = 0,003 \cdot \frac{U_{d0}}{I_d},$$

$$L_{p1} = L_{p2} = 0,003 \cdot \frac{660}{4000} = 0,5 \cdot 10^{-3} \text{ Гн}.$$

Вибираємо реактор СРОСЗ-2000У4. Індуктивність $L_p = 0,8 \cdot 10^{-3} \text{ Гн}$, номінальний струм $I_{pн} = 2000 \text{ А}$, активний опір $R_p = 0,33 \cdot 10^{-3} \text{ Ом}$.

Характеристики приводу визначаються також якістю датчика частоти обертання. Нелінійність і асиметрія характеристики викликають помилку у відробітку швидкості. Температурний дрейф приводить до збільшення помилок за швидкістю і зменшенню діапазону регулювання. Наявність низькочастотних пульсацій напруги викликає поява змінної складовою в струмі якоря і частоті обертання двигуна.

Для тиристорних приводів застосовуються тахогенератори ПТ, при розробці яких враховувалися вищезгадані вимоги. Вони мають фланцевих виконань і пристроюються безпосередньо до валу двигуна без сполучних муфт і кінематичних передач, чим обуславлюється прийнятний рівень оборотних пульсацій.

Амплітуда пульсацій вихідної напруги тахогенераторів %: оборотних – 0,25, полюсних – 0,5, зубцових – 1,4, колекторних – 0,6. нелінійність характеристики тахогенератора не перевищує 1 %.

Вибір типу тахогенератора проводиться по відповідності номінальних частот обертання двигуна підйому і тахогенератора.

Вибираємо тахогенератор ПТ-42, номінальна частота обертання $n_{птт} = 100 \text{ об/хв}$, напруга $U_{птт} = 230 \text{ В}$, струм якоря $I_{птт} = 0,1 \text{ А}$, напруга збудження $U_{тгзб} = 55 \text{ В}$, струм збудження $I_{тгзб} = 1 \text{ А}$.

2.6 Розрахунок параметрів силового кола електроприводу

Номінальна кутова частота обертання двигуна

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

$$\omega_n = \frac{3,14 \cdot 32}{30} = 3,35 \text{ с}^{-1}.$$

Індуктивність електродвигуна визначається по формулі Уманського-Лінвіля

$$L_{\text{д}} = \gamma \frac{U_{\text{н}}}{I_{\text{н}} \omega_{\text{н}} p}$$

де γ – коефіцієнт, рівний 0,1 для компенсованих двигунів

$$L_{\text{д}} = 0,1 \cdot \frac{600}{3180 \cdot 3,35 \cdot 5} = 1,125 \cdot 10^{-3} \text{ Гн}.$$

Приведена до кола випрямленого струму індуктивність трансформатора

$$L_{\text{т1}} = L_{\text{т2}} = \frac{U_{\text{кз}} \%}{100} \frac{1}{2mf} \frac{E_{\text{д0}}}{I_{\text{дн}}},$$

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

$E_{\text{д0}}$ – ЕДС холостого ходу перетворювача, для трифазної мостової схеми

$$E_{\text{д0}} = 2,34 U_{2\phi};$$

$U_{2\phi}$ – фазна напруга вентильної обмотки

$$U_{2\phi} = \frac{U_2}{\sqrt{3}} = \frac{570}{1,73} = 330 \text{ В}.$$

$$E_{\text{д0}} = 2,34 \cdot 330 = 770 \text{ В}.$$

$I_{\text{дн}}$ – сила номінального струму навантаження перетворювача, приймається рівній половині номінального струму двигуна (при паралельному включенні мостів)

$$I_{\text{дн}} = \frac{1}{2} I_{\text{н}} = \frac{1}{2} \cdot 3180 = 1590 \text{ А}.$$

$$L_{\text{т1}} = L_{\text{т2}} = \frac{6,2}{100} \cdot \frac{1}{2 \cdot 6 \cdot 50} \cdot \frac{770}{1590} = 0,046 \cdot 10^{-3} \text{ Гн}.$$

Сумарне значення індуктивності силового кола

$$L_{\Sigma} = \frac{(L_{\tau 1} + L_{p1}) \cdot (L_{\tau 2} + L_{p2})}{L_{\tau 1} + L_{p1} + L_{\tau 2} + L_{p2}} + L_{\text{д}};$$

$$L_{\Sigma} = \frac{(0,046 + 0,8) \cdot (0,046 + 0,8) \cdot 10^{-6}}{(0,046 + 0,8 + 0,046 + 0,8) \cdot 10^{-3}} + 1,125 \cdot 10^{-3} =$$
$$= 0,423 \cdot 10^{-3} + 1,125 \cdot 10^{-3} = 1,548 \cdot 10^{-3} \text{ Ом} /$$

Постійна проти-ЕДС двигуну

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

де k – конструктивна постійна двигуна;

$\Phi_{\text{н}}$ – номінальний потік збудження;

$$c = k \cdot \Phi_{\text{н}} = \frac{600 - 3180 \cdot 0,0146}{3,35} = 165.$$

Коефіцієнт посилення двигуна

$$k_{\text{д}} = \frac{1}{c},$$

$$k_{\text{д}} = \frac{1}{165} = 0,0061.$$

Коефіцієнт посилення тиристорного перетворювача

$$k_{\text{тп}} = \frac{U_{\text{дн}}}{U_{\text{вх тп}}},$$

де $U_{\text{дн}}$ – номінальна випрямлена напруга перетворювача;

$U_{\text{вх тп}}$ – вхідна напруга управління, згідно рекомендаціям [51] приймаємо

$$U_{\text{вх тп}} = 10 \text{ В}$$

$$k_{\text{тп}} = \frac{660}{10} = 66.$$

Активний опір якірного кола електродвигуна, приведений до робочої температури

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

де $\beta = 1,36$ – температурний коефіцієнт

$$R_{\text{як}} = 1,36 \cdot (0,00754 + 0,00114 + 0,000405) = 0,0146 \text{ Ом}.$$

Приведений до кола випрямленого струму активний опір обмотки трансформатора

$$R_{\text{т1}} = R_{\text{т2}} = (0,1 \div 0,2) \frac{e_{\text{к}} \% \pi E_{\text{д0}}}{100 m I_{\text{дн}}},$$

$$R_{\text{т1}} = R_{\text{т2}} = 0,15 \cdot \frac{6,2}{100} \cdot \frac{3,14}{6} \cdot \frac{770}{1590} = 0,0023 \text{ Ом}.$$

Опір від комутації тиристорів

$$R_{\text{к1}} = R_{\text{к2}} = m f L_{\text{т1}},$$

$$R_{\text{к1}} = R_{\text{к2}} = 6 \cdot 50 \cdot 0,046 \cdot 10^{-3} = 0,0139 \text{ Ом}.$$

Сумарне значення активного опору якірного кола

$$R_{\Sigma} = \frac{(R_{\text{т1}} + R_{\text{п1}} + R_{\text{к1}})(R_{\text{т2}} + R_{\text{п2}} + R_{\text{к2}})}{R_{\text{т1}} + R_{\text{п1}} + R_{\text{к1}} + R_{\text{т2}} + R_{\text{п2}} + R_{\text{к2}}} + R_{\text{як}};$$

$$R_{\Sigma} = \frac{(0,0023 + 0,00033 + 0,0139)(0,0023 + 0,00033 + 0,0139)}{0,0023 + 0,00033 + 0,0139 + 0,0023 + 0,00033 + 0,0139} + 0,0146 = 0,0082 + 0,0147 = 0,0228 \text{ Ом}$$

Електромагнітна постійна часу якірного кола

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

$$T_e = \frac{0,001548}{0,0228} = 0,069 \text{ с}.$$

Електромеханічна постійна часу електроприводу

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

де J_Σ – сумарний момент інерції, приведений до валу двигуна

$$J_\Sigma = m\rho^2,$$

$$\rho = \frac{V_{\max}}{\omega_d} = \frac{D_6}{2} \text{ – радіус приведення}$$

$$J_\Sigma = 116792 \cdot 2,5^2 = 729823 \text{ кг} \cdot \text{м}^2.$$

$$T_m = 729823 \cdot \frac{0,0228}{165^2} = 0,62 \text{ с}.$$

2.7 Функціональна схема системи автоматичного управління

Зростання вимог до якості управління та розширення діапазону регулювання швидкості обумовлюють доцільність застосування систем автоматичного управління (САУ) для шахтних підйомних машин, що базуються на принципах часткової або повної компенсації зовнішніх впливів. Сучасні модульні системи регуляторів, що використовують операційні підсилювачі, дають змогу реалізувати САУ підлеглого управління параметрами приводу з послідовною корекцією.

Система автоматичного управління швидкістю забезпечує необхідні статичні та динамічні характеристики приводу, які залежать від структури САУ

та виду вхідного сигналу. Для приводу підйомних машин найбільш відповідною є двократно-інтегруюча система підлеглого управління з задаючим сигналом від задатчика інтенсивності, який формує сигнал завдання швидкості згідно з потрібною діаграмою швидкості підйомної установки, з урахуванням обмежень на прискорення та його похідну - ривок.

Функціональна схема САУ електроприводу за системою ТП-Д з силовим реверсивним ТП зображена на рис.2.3.

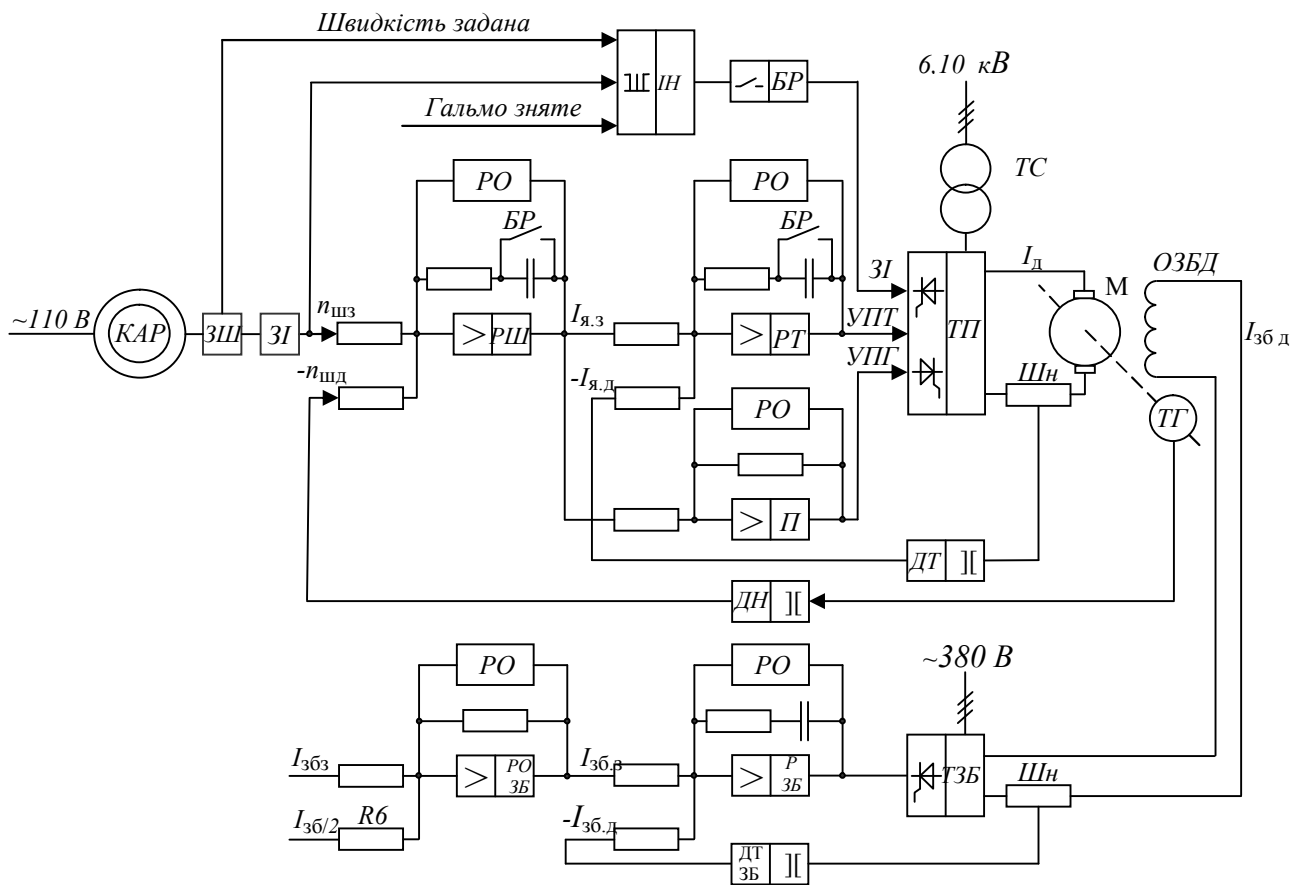


Рисунок 2.3 – Функціональна схема САУ:

$ТС$ – трансформатор силовий; $ЗІ$ – зрив імпульсів; $УПГ$ – управління перемиканням груп перетворювача; $УПТ$ – управління фазозміщенням перетворювача; $I_{зб\ д}$ – струм збудження двигуна; $РО$ – блоки обмеження; $n_{шз}$, $n_{шд}$ – сигнали заданої і дійсної швидкості; $I_{яз}$, $I_{яд}$ – сигнали заданого і дійсного якінного струму; $I_{зб\ з}$, $I_{зб\ д}$ – сигнали заданого і дійсного струму збудження; $ДТЗБ$ – датчик струму збудження.

Завдання рівня швидкості, що необхідно досягти, при ручному та автоматичному управлінні здійснюється командоапаратом ручного управління (КАР) через вузол завдання швидкості (ЗШ) і задатчик інтенсивності (ЗІ). При автоматичному виконанні циклу руху машини ЗШ забезпечує задання кількох ступенів (рівнів) швидкості за сигналами з технологічної автоматики. Задатчик ЗІ формує розгортку сигналу завдання певного рівня швидкості в часі, відповідно до допустимих значень прискорення та ривка.

Сигнал, що надходить з виходу ЗІ, подається на вхід регулятора швидкості (РШ), де він алгебраїчно підсумовується із сигналом негативного зворотного зв'язку за швидкістю. Вихідна напруга РШ є сигналом для регулювання струму в силовому ланцюгу двигуна. На вході регулятора струму (РТ) цей сигнал підсумовується з сигналом зворотного зв'язку по струму.

Напруга, що вимірюється за допомогою тахогенератора (ТГ) і шунта силового кола (Шн), подається на входи РШ і РТ через датчики напруги (ДН) і струму (ДТ), що забезпечують потенційне розділення САУ і силової схеми. Завдяки структурі регуляторів РШ і РТ як пропорційно-інтегральних, САУ має астатичність другого порядку: при обуреннях за завданням і зміні навантаження статична швидкісна помилка зводиться до нуля. Вихідний сигнал РТ є управляючим сигналом для ТП. Зміна величини та знаку цього сигналу визначає фазозміщення відмикаючих імпульсів СІФУ в діапазоні регулювання напруги випрямляча та інвертора ТП. Управління перемиканням груп ТП для зміни знаку моменту двигуна здійснюється підсилювачем (П), вихідна напруга якого змінює свій знак при зміні знаку напруги РШ.

При зупинці приводу для того, щоб забезпечити відсутність струму в двигуні, з ТП знімаються управляючі імпульси. Для цього використовується нуль-індикатор (ІН), який має три входи і функціонує за схемою АБО. Індикатор активується при появі сигналу 0,3-0,5 В на будь-якому з його входів. На входи ІН подаються сигнали: від ЗШ про включення приводу, від ЗІ про появу напруги завдання швидкості та від схеми технологічної автоматики про зняття робочого гальма. При зупинці приводу всі сигнали на вході ІН відсутні,

що активує безконтактне реле (БР), яке подає сигнал на зняття імпульсів обох груп ТП і шунтує конденсатори в колах зворотного зв'язку РШ і РТ, встановлюючи нульові початкові умови в САУ. Коли сигнал з'являється на одному з входів ІН, безконтактне реле відключається (його ключі замкнуті), і схема готова до нормальної роботи.

Схема управління збудженням двигуна забезпечує підтримку струму збудження на заданому рівні з високою точністю, а також форсоване наростання струму до номінального значення при пуску. При включенні схеми ще до початку руху на вхід підсилювача завдання величини струму збудження (РОЗБ) подається сигнал завдання половини номінального струму збудження. Під час пуску, за сигналом з автоматики, на вхід підсилювача РОЗБ подається напруга, яка додається до вже наявної напруги на вході РОЗБ і форсовано збільшується до необхідного рівня.

Регулятор струму збудження (РЗБ) є пропорційно-інтегральним і забезпечує відсутність помилки після досягнення завдання. Вихідний сигнал регулятора є управляючим сигналом для неререверсивного тиристорного збудника (ТЗБ), який живить обмотку збудження двигуна (ОЗБД).

Напруга регуляторів швидкості та струму може обмежуватись схемами регульованого обмеження (РО). Обмеження РШ задає максимальну допустиму величину струму в силовому ланцюгу двигуна. Обмеження РТ і РБ запобігають подачі на вхід СІФУ силового перетворювача та СІФУ збудника управлінської напруги, що перевищує необхідне для їх повного відкриття.

Конструктивно САУ є уніфікованою блочною системою регуляторів, зібраною з безконтактних аналогових елементів, що стандартизовані.

2.8 Рівняння динаміки і передавальні функції об'єкту управління

Об'єктом управління в досліджуваній системі є тиристорний перетворювач (разом із СІФУ) та електродвигун, що керується через ланцюг якоря. Тиристорний перетворювач є дискретним нелінійним пристроєм, який перетворює

безперервний вхідний сигнал управління $U_y(t)$ у моменти відкриття тиристорів $\alpha_i(t)$, що мають змінний інтервал, а потім здійснює зворотне перетворення цих дискретних значень у шматково-безперервний сигнал вхідної координати – ЕДС $E_d(t)$. Дискретний характер процесів у тиристорному перетворювачі можна детально врахувати, подаючи його у вигляді елемента з широтно-імпульсною модуляцією другого роду (ШІМ-II), де піки імпульсів представлені у вигляді відрізків синусоїдальної лінійної напруги. Як дискретний пристрій, тиристорний перетворювач підпорядковується теоремі Котельникова, що означає, що він пропускає без втрат інформації лише ті сигнали, частоти яких менші за певну граничну частоту.

$$\omega_{\text{гр}} = 2\pi f \frac{m}{2},$$

де m - число фаз перетворювача;

f - частота живлячої мережі

Умова допустимості зведення імпульсної системи до безперервної полягає в тому, щоб частота квантування імпульсного елемента $\omega = \frac{1}{T}$, де T - період квантування, була б більше або рівна потрібному значенню частоти сигналу $\omega_{\text{гр}}$, тобто

$$\omega \leq 3\omega_{\text{гр}},$$

Питання про опис динамічних властивостей тиристорного перетворювача з урахуванням дискретного процесу управління і неповної керованості тиристорів трактується по-різному. Частіше всього ТП в режимі безперервного струму описується ланкою, що складається з послідовного з'єднання лінійної безінерційної ланки з коефіцієнтом посилення $k_{\text{тп}}$ і ланки чистого запізнювання з постійною часу (середньостатистичного запізнювання) $\tau = (m \cdot f)^{-1}$, тобто

$$W_{\text{тп}}(p) = k_{\text{тп}} e^{-\tau p}. \quad (2.1)$$

Зазвичай $e^{-\tau p}$ розкладається в степеневий ряд і враховуються тільки два перші члени цього ряду. Тоді передатна функція (2.1) приймає вид:

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

Враховуючи фільтруючі властивості вхідних ланцюгів СІФУ, тиристорний перетворювач представляється у вигляді, представленому на рис. 2.4.

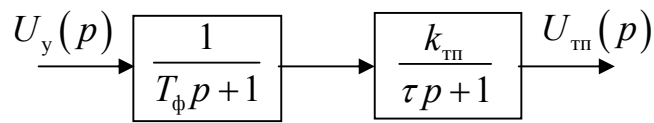


Рисунок 2.4 – Алгоритмічна схема тиристорного перетворювача

Рівняння динаміки двигуна визначимо для управління по ланцюгу якоря. Алгоритмічна схема ДПТ з незалежним збудженням представлена на рис.2.5.

Як видно з схеми – двигун є складною багатовимірною системою, що має 2 входи – управляючу дію $U_d(t)$ і збуджуючу дію $M_{\text{ст}}(I_{\text{ст}})(t)$ і три виходи – частота обертання $\omega_d(t)$; струм силового кола якоря $I_d(t)$ і кут повороту валу $\beta_d(t)$. При дослідженні системи регулювання швидкості остання вихідна величина не представляє інтересу, і надалі враховуватися не буде.

Динамічні властивості двигуна описуються передавальними функціями:

$$W_{I_d U_d}(p) = \left. \frac{I_d(p)}{U_d(p)} \right|_{I_{\text{ст}}=0} = \frac{T_m \cdot p}{R_{\Sigma} (1 + T_m p + T_{\epsilon} T_m p^2)};$$

$$W_{\omega_d U_d}(p) = \left. \frac{\omega_d(p)}{U_d(p)} \right|_{I_{\text{ст}}=0} = \frac{k_d}{1 + T_m p + T_{\epsilon} T_m p^2};$$

$$W_{I_d I_{\text{ст}}}(p) = \left. \frac{I_d(p)}{I_{\text{ст}}(p)} \right|_{\Delta U=0} = \frac{1}{1 + T_m p + T_{\epsilon} T_m p^2};$$

$$W_{\omega_d I_{ст}}(p) = \left. \frac{\omega_d(p)}{I_{ст}(p)} \right|_{\Delta U_d=0} = \frac{R_{\Sigma} k_d (1 + T_{\epsilon} p)}{1 + T_m p + T_{\epsilon} T_m p^2}.$$

Значення параметрів, що входять в передатні функції, будуть наведені нижче.

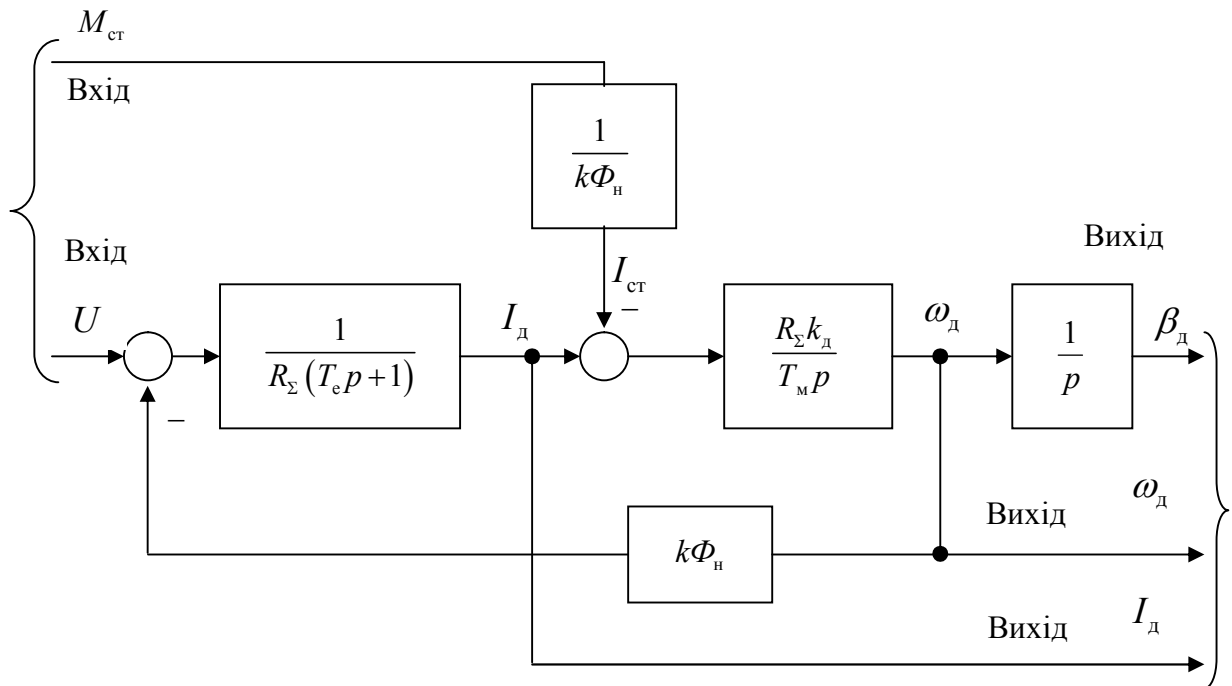


Рисунок 2.5 – Алгоритмічна схема двигуна постійного струму з управлінням по колу якоря

Таким чином, можна зробити висновок – що двигун є астатичною системою першого порядку по управляючій дії, $U_d(t)$ і виходу $\omega_d(t)$ і статичною системою по збурюючій дії $I_{ст}(t)$. Система має негативний зворотний зв'язок за швидкістю.

2.9 Синтез послідовного коректуючого пристрою контура струму

Розрахунок і оптимізація контура струму проводиться при наступних допущеннях:

1. Дана реверсивна схема управління електричним приводом з роздільним управлінням вентильними групами. Наявність зрівняльних струмів виключається.

2. Наявність фільтру на вході датчика струму з постійною часу $T_{\text{дт}}$ забезпечує згладжування напруги зворотного зв'язку і постійний коефіцієнт посилення аналогових підсилювачів схеми.

3. На вході СІФУ встановлений фільтр з постійною часу $T_{\text{ф}}$.

4. Вплив протити ЕДС двигуна враховується.

Алгоритмічна схема контура струму приведена на рис.2.6 а.

Для того, щоб можна було скористатися загальноприйнятою методикою синтезу послідовних коректуючих пристроїв, перетворимо алгоритмічну схему до схеми з одиничним зв'язком. Одержимо схему, показану на рис.2.6 б.

Некомпенсована мала постійна часу контура струму рівна сумі малих постійних часу ланок системи, тобто $T_{\text{мт}} = T_{\text{ф}} + T_{\text{дт}} + \tau$. З урахуванням цього, схема, приведена на рис. 2.6 б, перетвориться до вигляду, показаному на рис.2.6 в. Формуюча ланка $(1 + T_{\text{дт}}p)$ на вході системи згідно загальноприйнятій методиці не враховується.

Величиною $T_{\text{мт}}$ задамося у встановлених практикою межах $(4 \div 10) \text{мс}$. Приймаємо $T_{\text{мт}} = 5 \text{мс}$.

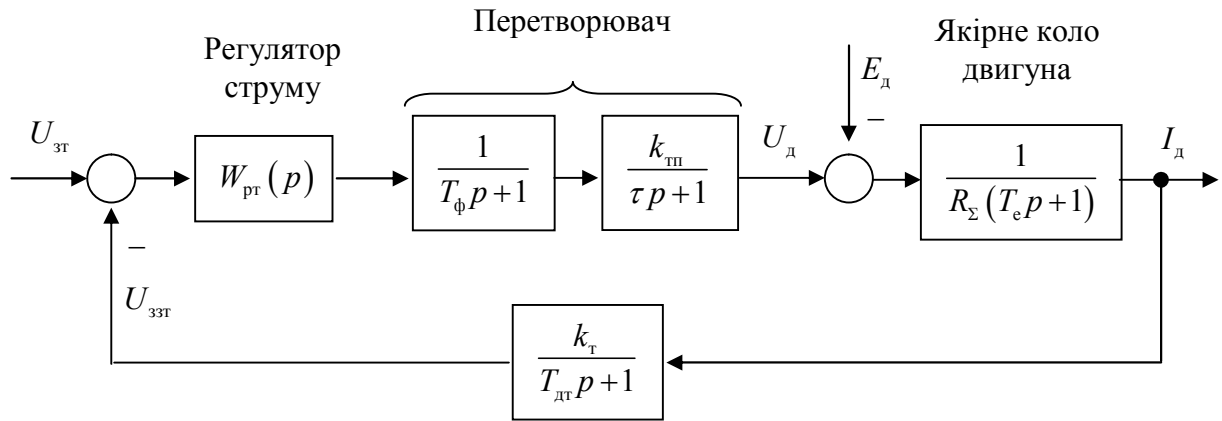
Оптимізацію контура струму виконаємо на основі використання модульного критерію, згідно якому передавальна функція оптимізованого контура має вид:

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

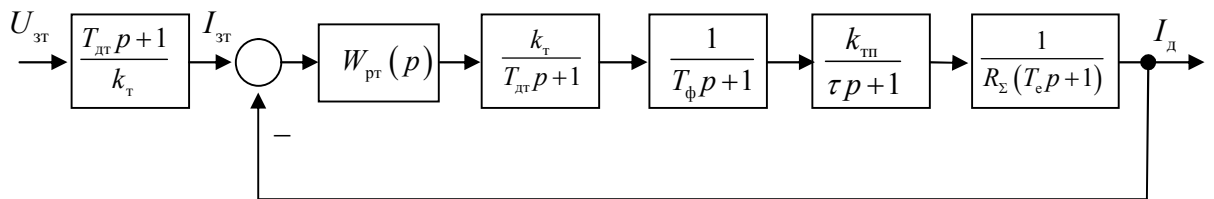
Відповідно до рис.2.6 в передавальна функція об'єкту управління

$$W_{\text{обг}}(p) = \frac{k_{\text{тп}} \cdot k_{\text{т}}}{R_{\Sigma}(1 + T_{\text{мт}}p)(1 + T_{\text{е}}p)},$$

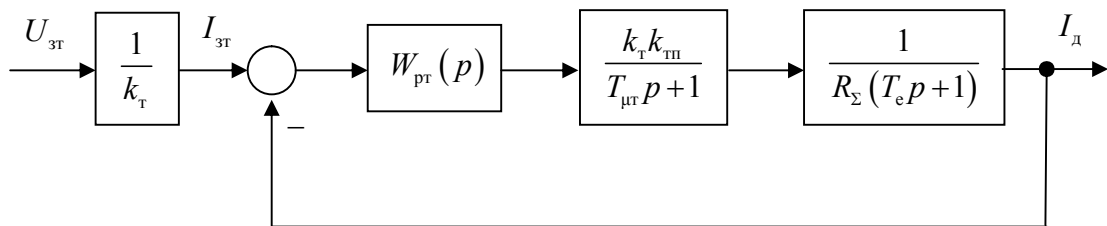
де $k_{\text{т}}$ - коефіцієнт посилення кола зворотного зв'язку по струму.



а)



б)



в)

Рисунок 2.6 - Алгоритмічні схеми контура струму:

- а) – вихідна схема; б) – схема, перетворена до одиничного зворотного зв'язку;
в) - розрахункова алгоритмічна схема

Оскільки

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

то, передавальна функція регулятора струму

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

де $k_{\text{рт}}$ - коефіцієнт посилення регулятора струму

$$k_{\text{рт}} = \frac{R_{\Sigma}}{2T_{\text{мт}} \cdot k_{\text{тп}} \cdot k_{\text{т}}}.$$

Отриману передавальну функцію регулятора струму $W_{\text{рт}}(p)$ можна реалізувати за допомогою типового пропорційно-інтегрального регулятора. Принципова схема ПІ- регулятора приведена на рис. 2.7.

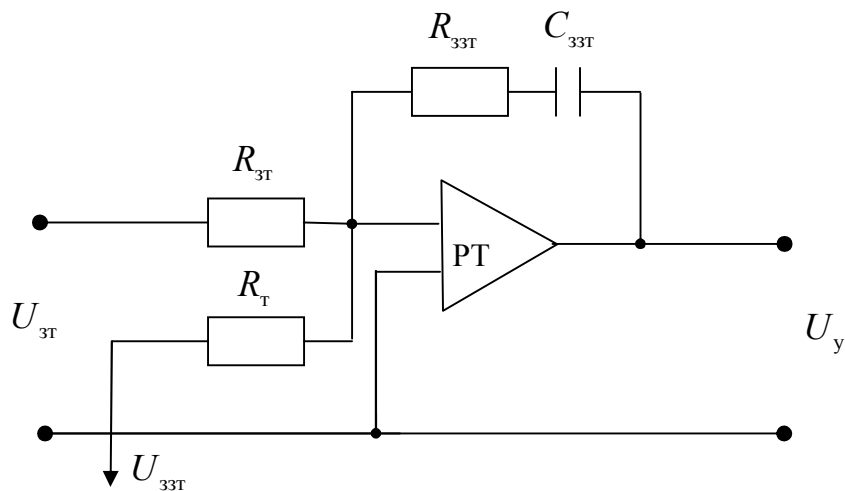


Рисунок 2.7 - Принципова схема регулятора струму на базі операційного підсилювача

Величину максимального струму якірного кола визначимо з рекомендованих співвідношень $I_{\text{н ш}} = I_{\text{max}} = 2I_{\text{н}} = 2 \cdot 3180 = 6360\text{A}$. Величину максимальної

напруги завдання струму $U_{\text{зт max}}$ (вхідна напруга регулятора швидкості) прийmemo рівною $U_{\text{зт max}} = 24 \text{ В}$.

Для отримання сигналу, пропорційного струму якiрного кола використовуємо шунт нормалізованої серії з номінальним струмом $I_{\text{н ш}} = I_{\text{max}} = 6360 \text{ А}$ і номінальною напругою $U_{\text{шн}} = 75 \text{ мВ}$.

Коефіцієнт посилення шунта

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

$$k_{\text{ш}} = \frac{75 \cdot 10^{-3}}{6360} = 0,0118 \cdot 10^{-3} \text{ В / А}.$$

Для вимірювання і гальванічного розділення ланцюгів управління від силових ланцюгів вибираємо датчик струму. Коефіцієнт посилення датчика струму визначуваний таким чином

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

$$k_{\text{дт}} \leq \frac{24}{0,0118 \cdot 10^{-3} \cdot 6360} = 319,8 \text{ В/А}$$

Приймаємо коефіцієнт посилення датчика струму $k_{\text{дт}} = 320$.

Коефіцієнт посилення кола зворотного зв'язку за струмом визначається виразом:

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

де $\frac{R_{\text{зт}}}{R_{\text{т}}}$ – коефіцієнт приведення напруги зворотного зв'язку по струму до кола задаючого сигналу регулятора струму;

$R_{зт}, R_{т}$ – вихідні опори регулятора по каналам завдання і зворотного зв'язку за струмом. Обмеження вихідного сигналу регулятора швидкості здійснюється шляхом використання регульованого джерела опорної напруги, тому приймаємо $R_{зт} = R_{т}$, тоді

$$k_{т} = k_{дт} \cdot k_{ш},$$

$$k_{т} = 320 \cdot 0,00118 \cdot 10^{-3} = 0,038 \cdot 10^{-3} \text{ В / А}$$

$$k_{т} = 320 \cdot 0,0118 \cdot 10^{-3} = 0,0038 \text{ В/А}$$

Виконаємо розрахунок елементів регулятора. Задавшись величиною вхідних опорів $R_{т} = R_{зт} = 10 \text{ кОм}$ визначимо значення $C_{зт}$ із співвідношення

$$\frac{1}{C_{зт} R_{т}} = k_{пт} = \frac{R_{\Sigma}}{2T_{\mu т} k_{тп} k_{т}},$$

звідки

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

$$C_{зт} = \frac{2 \cdot 0,005 \cdot 66 \cdot 0,038 \cdot 10^{-3}}{0,0228 \cdot 10 \cdot 10^3} = 1,1 \cdot 10^{-6} \text{ Ф}.$$

Постійна часу ПІ-регулятора

$$T_{пт} = T_{е} = R_{зт} C_{зт},$$

звідки

$$R_{зт} = \frac{T_{е}}{C_{зт}} = \frac{0,069}{1,1 \cdot 10^{-6}} = 62,7 \cdot 10^3 \text{ Ом}.$$

Передавальна функція замкнутого оптимізованого контура струму має вид:

$$W_{зт}(p) = \frac{1}{k_{т}} \cdot \frac{1}{2T_{\mu т}^2 p^2 + 2T_{\mu т} p + 1}. \quad (2.2)$$

2.10 Синтез послідовного коректуючого пристрою контура швидкості

У регуляторах швидкості контур швидкості виступає як зовнішній, тому динамічні характеристики такого регулятора повністю визначаються динамікою контуру швидкості.

До складу об'єкта регулювання контуру швидкості входять: механічна частина електроприводу, яка є ідеальною інтегруючою ланкою з коефіцієнтом посилення $R_{\Sigma} k_d / T_m$, оптимізованого контуром струму, властивості якого визначаються передавальною функцією (2.2), а також датчик швидкості, який зазвичай реалізується за допомогою тахогенератора постійного струму. На виході датчика швидкості розташований фільтр у вигляді аперіодичної ланки з постійною часу $T_{дш}$. Алгоритмічна схема системи представлена на рис. 2.8 а.

Оптимізацію контура швидкості виконаємо із застосуванням симетричного критерію.

Перетворена до схеми з одиничним зворотним зв'язком алгоритмічна схема показана на рис. 2.8 б.

Мала некомпенсуєма постійна часу контура швидкості

$$T_{\mu ш} = 2T_{\mu т} + T_{дш} .$$

Приймаємо $T_{дш} = 0$, тоді

$$T_{\mu ш} = 2 \cdot 5 \cdot 10^{-3} = 0,01 \text{ с} .$$

Бажана передавальна функція контура швидкості, оптимізованого на підставі симетричного критерію оптимізації

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

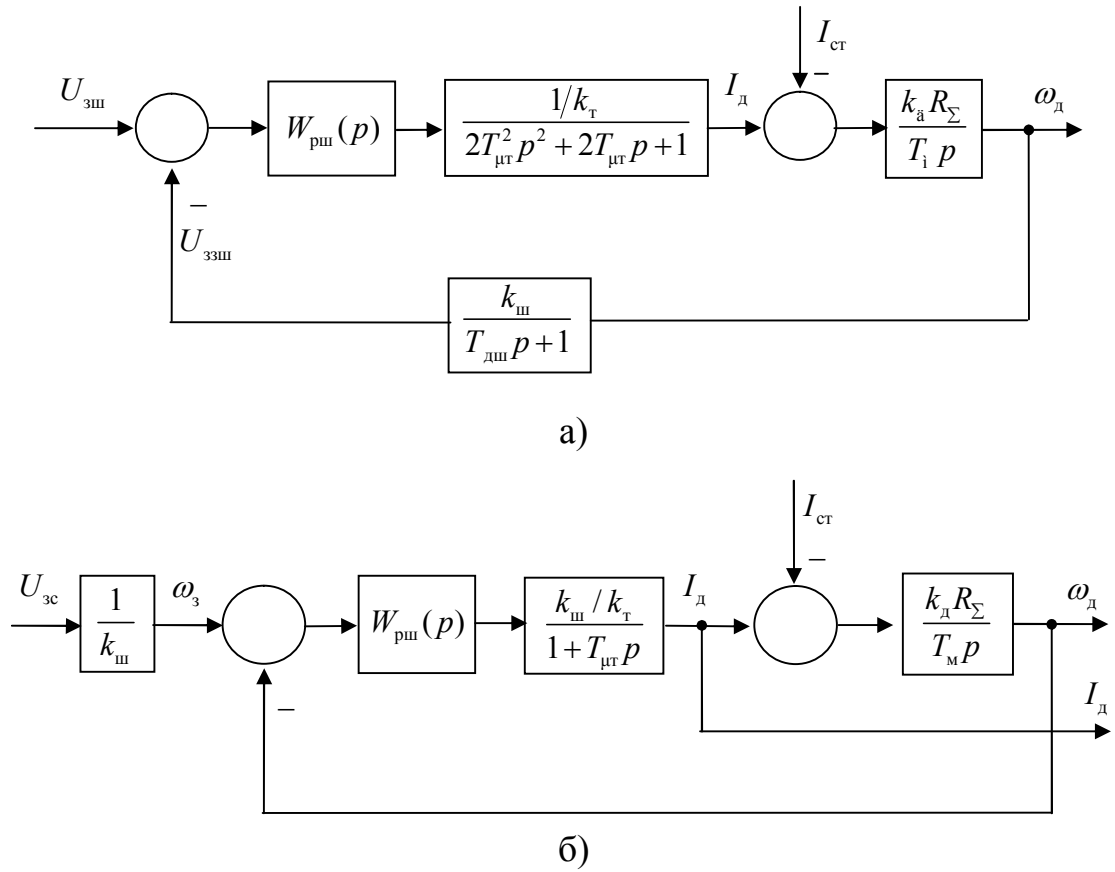


Рисунок 2.8 - Алгоритмічні схеми контура швидкості:
а) – вихідна схема; б) - розрахункова алгоритмічна схема

Розділивши бажану передавальну функцію на передавальну об'єкту регулювання отримаємо передавальну функцію регулятора контура швидкості

$$\begin{aligned}
 W_{рш}(p) &= \frac{W_{бш}(p)}{W_{обш}(p)} = \frac{\frac{1 + 4T_{\muш}p}{8T_{\muш}^2 p^2 (1 + T_{\muш}p)}}{\frac{k_{ш}k_d R_{\Sigma}}{k_t T_m} \cdot \frac{1}{p(1 + T_{\muш}p)}} = \\
 &= \frac{T_m k_t}{8T_{\muш}^2 k_d k_{ш} R_{\Sigma}} \cdot \frac{1 + 4T_{\muш}p}{p} = k_{рш} \cdot \frac{1 + 4T_{\muш}p}{p},
 \end{aligned} \tag{2.3}$$

де $k_{рш} = \frac{T_m k_t}{8T_{\muш}^2 k_d k_{ш} R_{\Sigma}}$ - коефіцієнт посилення регулятора контура швидкості.

Передатна функція (2.3) реалізується за допомогою типового ПІ – регулятора, принципова схема якого показана на рис.2.9.

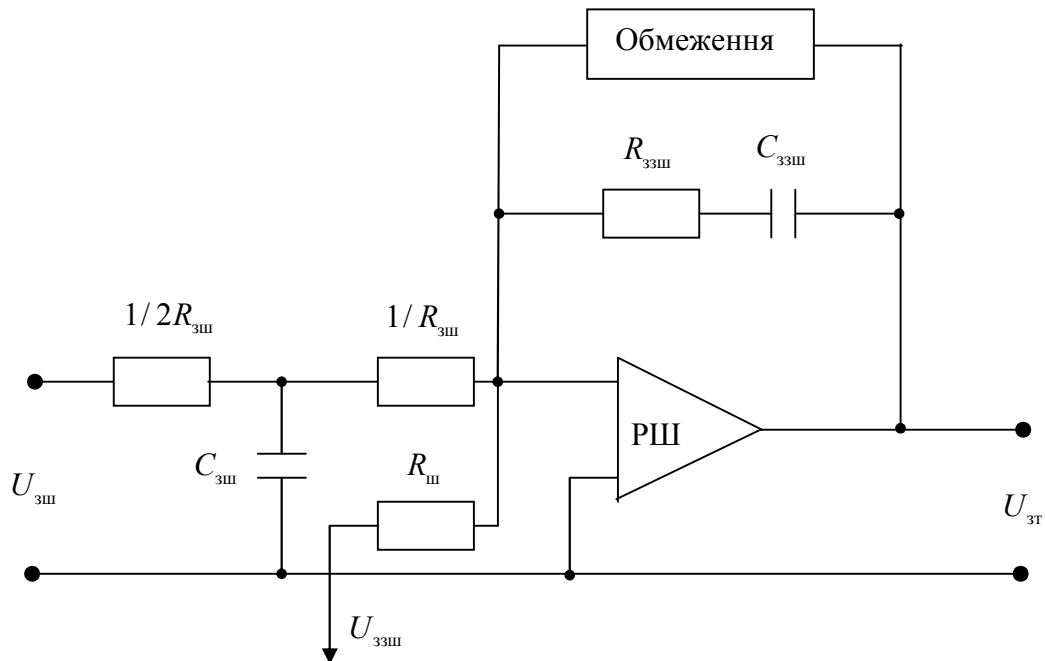


Рисунок 2.9 – ПІ-регулятор контура швидкості

Зіставляючи передавальну функцію (2.3) з передавальною функцією типового ПІ-регулятора, отримуємо наступні рівняння

$$R_{зш} \cdot C_{зш} = 4T_{\mu ш} , \quad (2.4)$$

$$\frac{R_{зш}}{R_{ш}} = \frac{T_{\mu} \cdot k_{\tau}}{2T_{\mu ш} \cdot k_{ш} \cdot k_{\text{д}} \cdot R_{\Sigma}} . \quad (2.5)$$

Коефіцієнт посилення зворотного зв'язку по швидкості $k_{ш}$ визначається з виразу

$$k_{ш} = k_{\tau} \frac{R_{зш}}{R_{ш}} ,$$

де k_{τ} – коефіцієнт посилення тахогенератора

$$k_{\text{тг}} = \frac{U_{\text{н тг}}}{\omega_{\text{н тг}}},$$

де $\omega_{\text{н тг}}$ - номінальна швидкість обертання тахогенератора c^{-1}

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

$$\omega_{\text{н тг}} = \frac{3,14 \cdot 100}{30} = 10,5 \text{ } c^{-1},$$

$$k_{\text{тг}} = \frac{230}{10,5} = 22.$$

Відношення опорів $\frac{R_{\text{зш}}}{R_{\text{ш}}}$ знаходимо по рівнянню

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

звідки

$$\frac{R_{\text{зш}}}{R_{\text{ш}}} = \frac{U_{\text{зш max}}}{k_{\text{тг}} \cdot \omega_{\text{max}}}, \quad (2.6)$$

де ω_{max} – задана максимальна величина швидкості обертання двигуна, приймаємо $\omega_{\text{max}} = \omega_0$, де ω_0 – швидкість ідеального холостого ходу двигуна

$$\omega_0 = \frac{U_{\text{н}}}{c},$$

$$\omega_0 = \frac{600}{165} = 3,6 \text{ } c^{-1}.$$

$U_{\text{зш max}}$ – вихідна напруга задаючого елемента, відповідна ω_{max} . Приймаємо $U_{\text{зш max}} = 24 \text{ В}$.

Задаємося величиною $R_{\text{зш}} = 20 \cdot 10^3 \text{ Ом}$, тоді з (2.6) знаходимо

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

$$R_{ш} = \frac{22 \cdot 3,6}{24} \cdot 20 \cdot 10^3 = 66 \cdot 10^3 \text{ Ом}.$$

Із співвідношення (2.5) визначаємо опір $R_{3ш}$

$$R_{3ш} = \frac{T_{\mu} k_{т}}{8 T_{\mu ш}^2 k_{тг} k_{д} R_{\Sigma}} \cdot R_{ш},$$

$$R_{3ш} = \frac{0,62 \cdot 0,038 \cdot 10^{-3}}{8 \cdot 0,01^2 \cdot 22 \cdot 0,0061 \cdot 0,0228} \cdot 66 \cdot 10^3 = 9,6 \cdot 10^3 \text{ Ом}.$$

Із співвідношення (2.4) визначаємо

$$C_{3ш} = \frac{4 T_{\mu ш}}{R_{3ш}}$$

$$C_{3ш} = \frac{4 \cdot 0,01}{9,6 \cdot 10^3} = 4,16 \cdot 10^{-6} \text{ Ф}.$$

Як відомо, при настроюванні контура швидкості на симетричний оптимум маємо перерегулювання швидкості при ступінчастій задаючій дії рівне 43%, Для зниження перерегулювання на вхід системи ставимо фільтр, що є аперіодичною ланкою з постійною часу $4T_{\mu ш}$. Включення фільтру показане на рис. 2.9.

Величина ємності $C_{3ш}$ визначається з рівняння:

$$\frac{R_{зс}}{4} C_{3ш} = 4 T_{\mu ш},$$

звідки

$$C_{3ш} = \frac{4 \cdot 4 \cdot T_{\mu ш}}{R_{зс}}.$$

$$C_{\text{зш}} = \frac{16 \cdot 0,01}{20 \cdot 10^3} = 8 \cdot 10^{-6} \Phi.$$

З алгоритмічної контура швидкості, зображеного на рис. 2.10 шляхом її структурного перетворення, визначасмо наступні вирази для передавальних функцій оптимізованої системи по задаючій $U_{\text{зш}}(t)$ і збурюючій $I_{\text{ст}}(t)$ діям

$$W_{\omega_{\text{д}} U_{\text{зш}}}(p) = \left. \frac{\omega_{\text{д}}(p)}{U_{\text{зш}}(p)} \right|_{\substack{I_{\text{ст}} = 0 \\ \Phi_{\text{н}} = \text{const}}} = \frac{1}{k_{\text{ш}}} \cdot \frac{1 + 4T_{\mu\text{ш}}p}{8T_{\mu\text{ш}}^3 p^3 + 8T_{\mu\text{ш}}^2 p^2 + 4T_{\mu\text{ш}}p + 1},$$

$$W_{\omega_{\text{д}} I_{\text{ст}}}(p) = \left. \frac{\omega_{\text{д}}(p)}{I_{\text{ст}}(p)} \right|_{\substack{\Phi_{\text{н}} = \text{const} \\ U_{\text{зш}} = \text{const}}} = - \frac{k_{\text{д}} \cdot R_{\Sigma} \cdot 8T_{\mu\text{ш}}^2 p \cdot (1 + T_{\mu\text{с}}p) / T_{\text{м}}}{8T_{\mu\text{ш}}^3 p^3 + 8T_{\mu\text{ш}}^2 p^2 + 4T_{\mu\text{ш}}p + 1},$$

$$W_{I_{\text{д}} U_{\text{зш}}}(p) = \left. \frac{I_{\text{д}}(p)}{U_{\text{зш}}(p)} \right|_{\substack{I_{\text{ст}} = 0 \\ \Phi_{\text{н}} = \text{const}}} = \frac{T_{\text{м}}}{k_{\text{д}} R_{\Sigma} k_{\text{ш}}} \cdot \frac{p(1 + 4T_{\mu\text{ш}}p)}{8T_{\mu\text{ш}}^3 p^3 + 8T_{\mu\text{ш}}^2 p^2 + 4T_{\mu\text{ш}}p + 1},$$

$$W_{I_{\text{д}} I_{\text{ст}}}(p) = \left. \frac{I_{\text{д}}(p)}{I_{\text{ст}}(p)} \right|_{\substack{\Phi_{\text{н}} = \text{const} \\ U_{\text{зш}} = \text{const}}} = \frac{1 + 4T_{\mu\text{ш}}p}{8T_{\mu\text{ш}}^3 p^3 + 8T_{\mu\text{ш}}^2 p^2 + 4T_{\mu\text{ш}}p + 1}.$$

Розгорнута алгоритмічна схема системи показана на рис. 2.11.

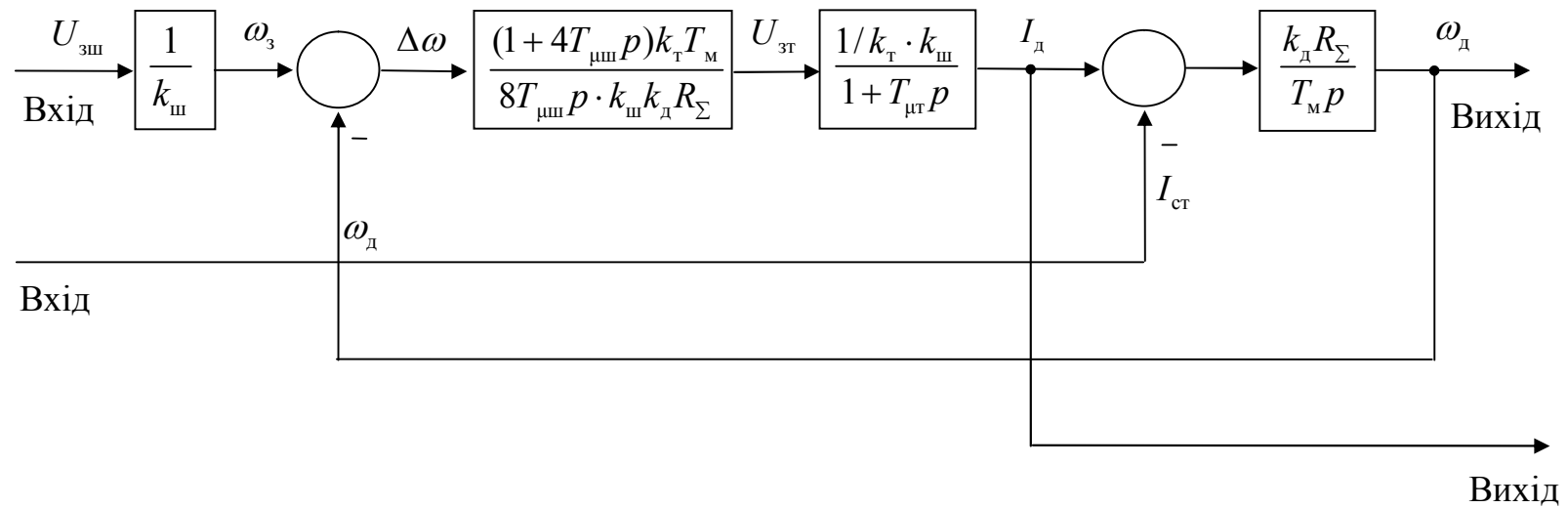


Рисунок 2.10 - Алгоритмічна контура швидкості

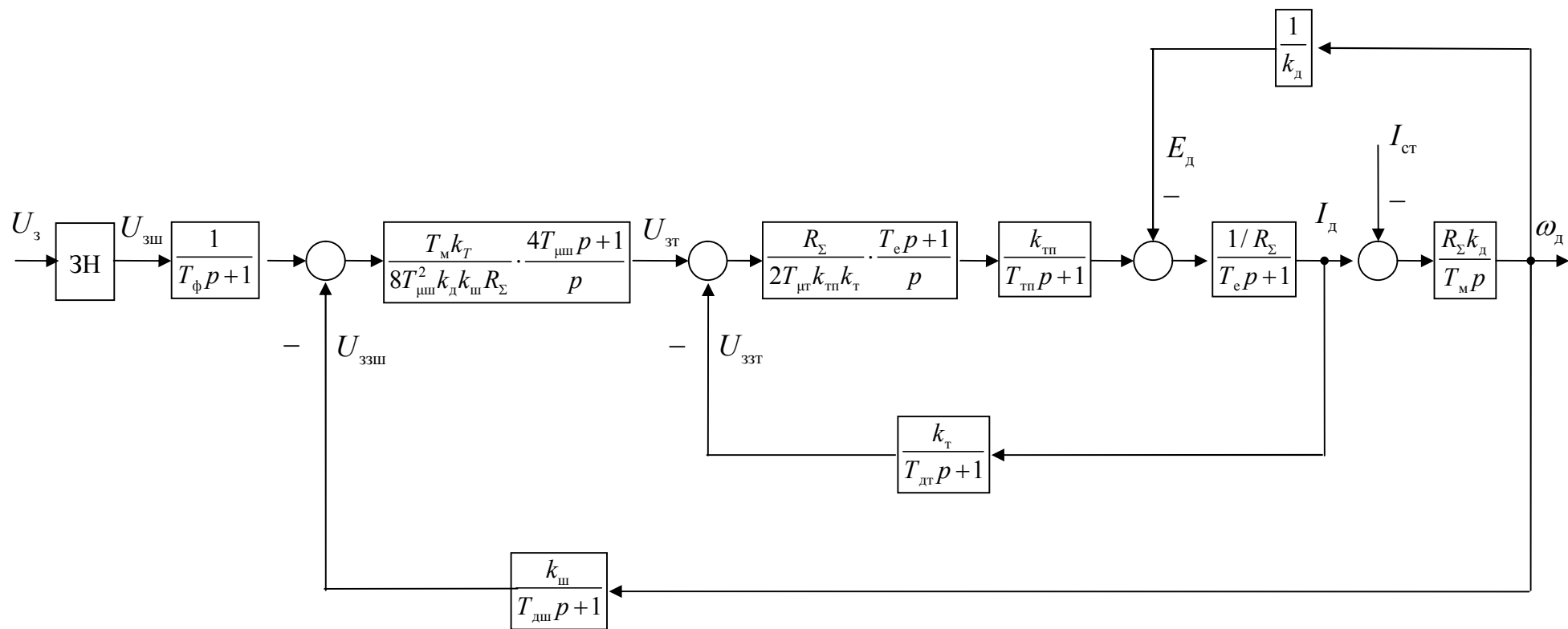
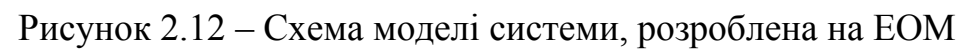


Рисунок 2.11– Алгоритмічна схема оптимізованої системи

2.11 Дослідження динамічних характеристик системи

Важливим етапом синтезу системи автоматичного управління є дослідження її динамічних характеристик. На рис. 2.13 і рис. 2.14 приведені графіки перехідних процесів змінних стану системи по задавальній і збурюючій діях. Розрахунок проведений на ЕОМ з використанням системи MATLAB. Для розрахунків використовувалися як отримані вище передатні функції, так і схема моделі системи, побудована в режимі SIMULINK. Схема моделі системи приведена на рис.2.12. Як вказувалось вище, при настройці контура швидкості по симетричному критерію є значне перерегулювання швидкості, рівне 43%. Для зменшення перерегулювання на вхід системи включено фільтр з постійною часу $4T_{\mu 0}$, при цьому перерегулювання системи знизилось до 8,1%. Графіки перехідних процесів без фільтру і з фільтром позначені на рис. 2.13 відповідно 1 і 2.



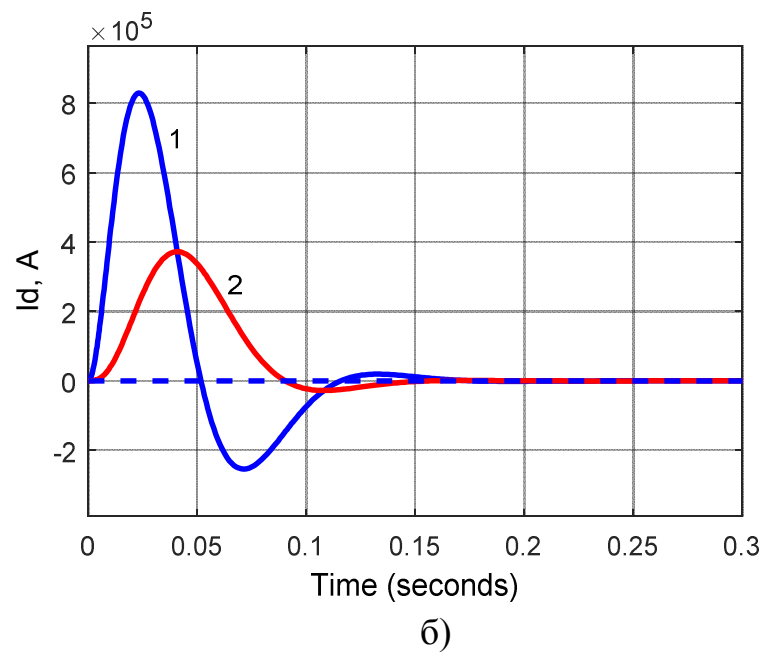
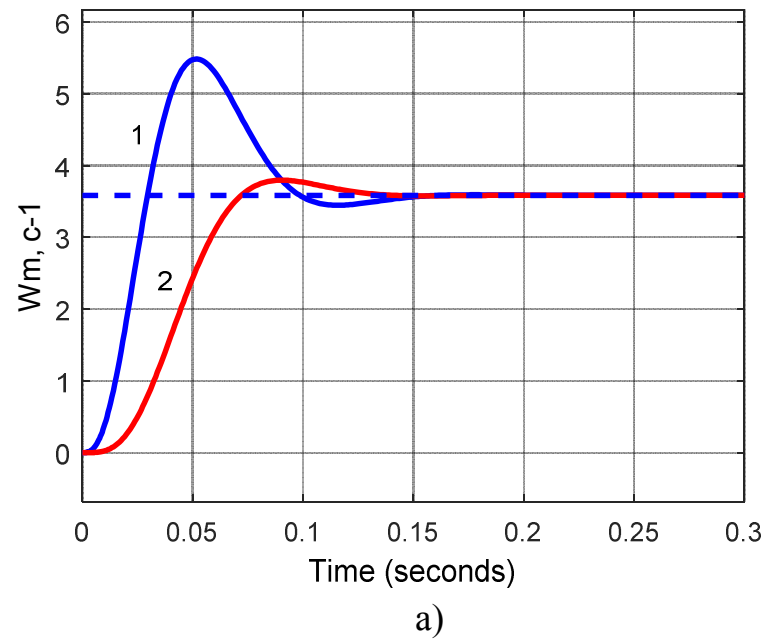


Рисунок 2.13 – Графіки перехідних процесів змінних стану системи по заляючій дії:
а) – графік швидкості $\omega_d = f(t)$; б) – графік струму $I_d = f(t)$

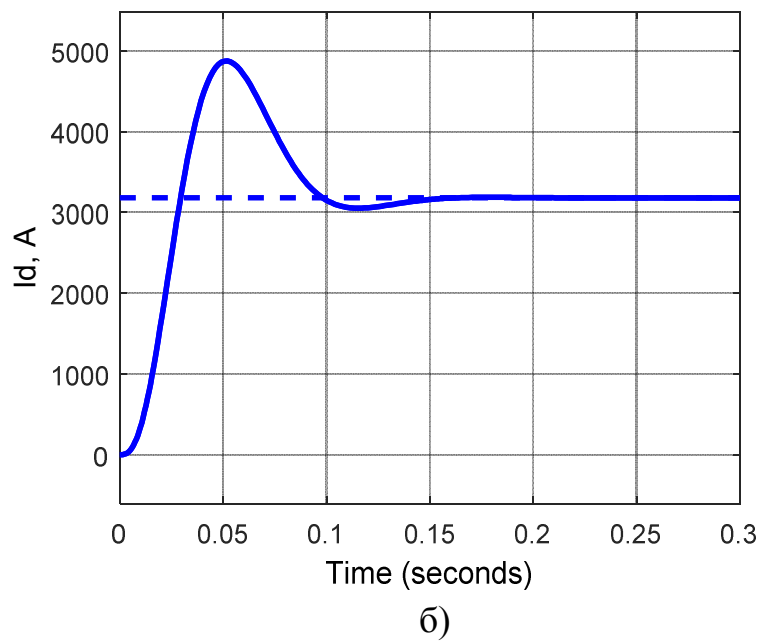
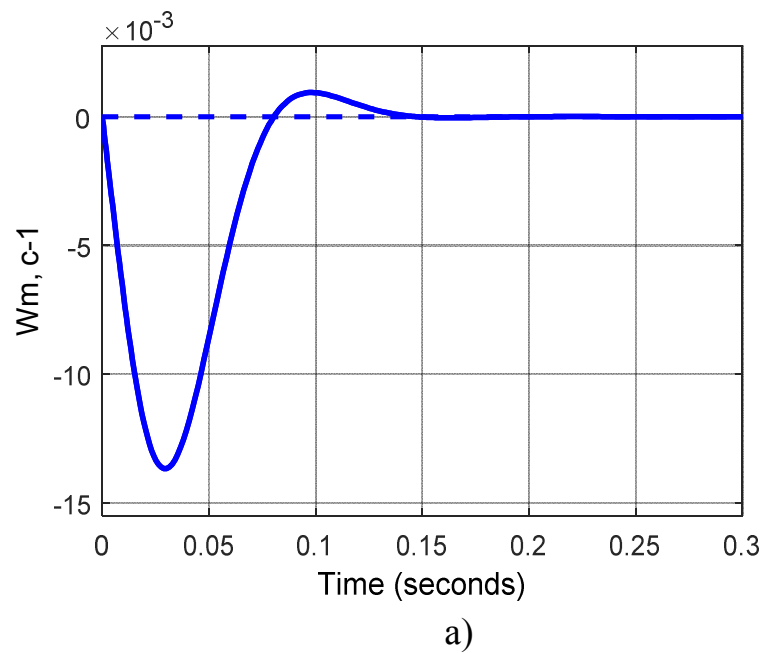


Рисунок 2.14 – Графіки перехідних процесів змінних стану системи по збурюючій дії:

а) – графік швидкості $\omega_d = f(t)$; б) – графік струму $I_d = f(t)$;

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 ТП–Д.

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

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

4. Проведено вибір структури системи автоматичного управління. Система регулювання повинна пояснювати необхідні механічні характеристики електроприводу в статиці 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 підйомальної установки вибрано двоконтурну систему підлеглого регулювання з контуром струму і 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 струму і швидкості синтезовані ПІ-регулятори, унаслідок чого контур струму оптимізований 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 $4T_{\text{шп}}$, унаслідок чого перерегулювання знижене до 8,1%.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ВЛАСТИВОСТЕЙ ПІДЙОМНОГО КАНАТА

3.1 Математична моделі трьохмасової системи

Розрахунок системи підлеглого регулювання здійснений за припущенням нескінченної жорсткості всіх елементів механічної частини, включаючи підйомний канат, тобто для одномасової системи. У реальних системах усі зв'язки між елементами мають кінцеву жорсткість, і підйомний канат є пружним елементом з розподіленими параметрами.

При розгляді каната як пружного елементу з розподіленою масою можна ігнорувати ряд незначних факторів. Наприклад, канат можна вважати консервативною ланкою, оскільки демпфування, що здійснюється двигуном і тертям підйомної судини об ті, значно перевищує сили внутрішнього тертя в канаті. Також не суттєво нерівномірне розподілення маси по довжині каната, яке викликане його деформацією від власної ваги.

В урахуванні даних допущень основне значення мають подовжні коливання каната, тоді як поперечні і крутильні коливання є другорядними. Найбільший вплив на процеси в системі має навантажена частина каната, коли судно знаходиться в крайньому нижньому положенні, а тривалість перехідного режиму під час розгону незначна, тому зміна довжини каната через його навивку на барабан є малою у відношенні до загальної довжини каната. Тому можна припустити, що довжина пружного елемента і його жорсткість залишаються незмінними в період розгону.

З урахуванням зазначених припущень розрахункову схему механічної підйомної установки можна зобразити як дві зосереджені маси m_1 і m_2 , сполучені між собою ідеально пружним стрижнем (рис. 3.1).

У [52] приводиться методика розрахунку вказаного завдання. Проте для практичних розрахунків ця методика достатньо складна. В цілях спрощення

можна розглядати механічну частину підйомної установки як трьохмасову систему, в якій підйомний канат зображається зосередженою масою, пов'язаною з двома іншими масами еквівалентними невагомими пружними зв'язками, а також двомасової системи, в якій канат приведений до зосереджених мас (див. рис. 3.1).

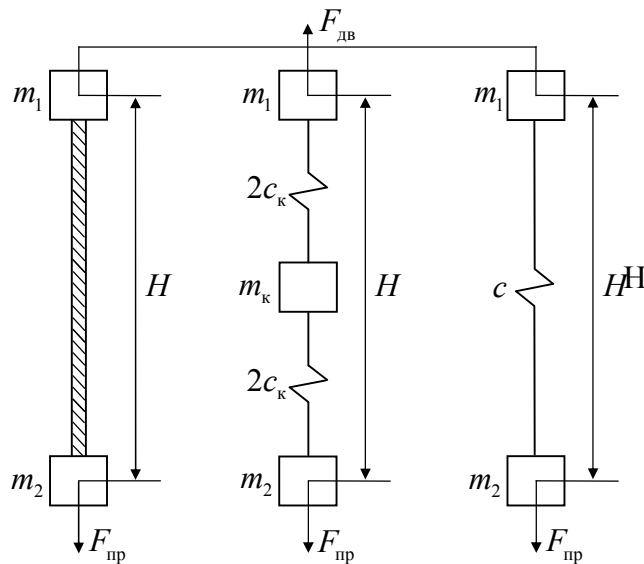


Рисунок 3.1 – Розрахункові схеми механічної частоти підйомної установки

У попередньому розділі по кривим рис.1.2 [52] були вибрані відносні приведені маси μ_1 , μ_2 і μ_k (залежно від глибини підйому).

Зосереджена відома m_1 , ведуча m_2 маси і маса каната m_k рівні:

$$m_1 = \mu_1 m \quad m_2 = \mu_2 m \quad m_k = \mu_k m,$$

де $m = m_1 + m_2 + m_{1k} \cdot H$ - сумарна приведена маса

m_{1k} - маса одиниці довжини каната.

У інженерних розрахунках розглядають підйомний канат, що володіє розподіленою масою, пов'язаною з масами m_1 і m_2 еквівалентними невагомими пружними зв'язками з жорсткістю c_1 і c_2 , тобто розглядають трьохмасову сис-

тому. В цілях спрощення досліджень трьохмасова система може бути замінена двомасовою з приведеними масами $m'_1 = m_1 + m_k / 2$ і $m'_2 = m_2 + m_k / 2$, сполученими невагомим канатом з жорсткістю c_{12} .

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

$$\left\{ \begin{array}{l} J_{\Sigma} \frac{d\omega_{\Sigma}}{dt} = k\Phi I_{\Sigma} - M_{\Sigma 1}; \\ J_k \frac{d\omega_k}{dt} = M_{\Sigma 1} - M_{\Sigma 2}; \\ J_m \frac{d\omega_m}{dt} = M_{\Sigma 2} - M_{\Sigma 3}; \\ \frac{dM_{\Sigma 1}}{dt} = c_1 (\omega_{\Sigma} - \omega_k); \\ \frac{dM_{\Sigma 2}}{dt} = c_2 (\omega_k - \omega_m), \end{array} \right. \quad (3.1)$$

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

$$M_{\Sigma 1} = M_{\Sigma 1p} + M_{\Sigma 1v}, \quad M_{\Sigma 2} = M_{\Sigma 2p} + M_{\Sigma 2v}, \quad (3.2)$$

$M_{\Sigma 1p}$, $M_{\Sigma 2p}$ - моменти пружної взаємодії між масами трьохмасової системи;

$M_{\Sigma 1v}$, $M_{\Sigma 2v}$ - моменти в'язкого тертя

$$M_{\Sigma 1v} = \beta_1 (\omega_{\Sigma} - \omega_k); \quad M_{\Sigma 2v} = \beta_2 (\omega_k - \omega_m), \quad (3.3)$$

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

J_{Σ} - приведений момент інерції провідної маси: $J_{\Sigma} = m_1 \rho^2$;

$J_k = m_k H \rho^2$ - приведений момент інерції каната (другої маси);

$m_{\kappa 1}$ - маса одного погонного метра каната;

H - висота підйому;

$J_{\text{м}}$ - приведений момент інерції навантаженого скипа: $J_{\text{м}} = m_2 \rho^2$;

$\rho = \frac{V_{\text{п}}}{\omega_{\text{д}}} = \frac{D_6}{2}$ - радіус приведення поступальної швидкості до обертальної;

c_1, c_2 - приведені жорсткості ділянок між скипами визначаються таким чином: $c_1 = c_2 = 2c_{\kappa} \rho^2$;

c_{κ} - жорсткість каната: $c_{\kappa} = c_{1\kappa} / H$;

$c_{1\kappa}$ - жорсткість одного метра підйомного каната;

Для двомасової системи рівняння руху мають вид:

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

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

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

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

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

Розглянемо трьохмасову систему підлеглого регулювання швидкості двигуна з регулятором швидкості, настроєним по симетричному критерію з урахуванням контура струму, налаштованим на модульний оптимум.

Запишемо систему (3.1) з урахуванням (3.2) і (3.3) у формі Коші

$$\left\{ \begin{aligned} \frac{d\omega_{\text{м}}}{dt} &= \frac{1}{J_{\text{м}}} M_{\text{пп2}} + \frac{\beta_2}{J_{\text{м}}} \omega_{\text{к}} - \frac{\beta_2}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пп1}}}{dt} &= c_1 \omega_{\text{д}} - c_1 \omega_{\text{к}}; \\ \frac{d\omega_{\text{к}}}{dt} &= \frac{1}{J_{\text{к}}} M_{\text{пп1}} - \frac{\beta_1}{J_{\text{к}}} \omega_{\text{д}} + \frac{\beta_1}{J_{\text{к}}} \omega_{\text{к}} - \frac{1}{J_{\text{к}}} M_{\text{пп2}} - \frac{\beta_2}{J_{\text{к}}} \omega_{\text{к}} + \frac{\beta_2}{J_{\text{к}}} \omega_{\text{м}}; \\ \frac{dM_{\text{пп2}}}{dt} &= c_2 \omega_{\text{к}} - c_2 \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} &= \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пп1}} - \frac{\beta_1}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta_1}{J_{\text{д}\Sigma}} \omega_{\text{к}}. \end{aligned} \right. \quad (3.4)$$

Розглянемо алгоритмічну схему одномасової системи, описану в попередньому розділі, і запишемо рівняння стану системи регулювання без урахування пружних властивостей елементів механічної частини. Далі, додавши рівняння трьохмасової системи, отримаємо диференціальні рівняння стану системи управління з урахуванням пружного зв'язку.

Рівняння стану одномасової системи:

$$\left\{ \begin{aligned} \frac{d\omega_{\text{д}}}{dt} &= \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dI_{\text{д}}}{dt} &= -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma}T_{\text{е}}} U_{\text{тп}}; \\ \frac{dU_{\text{тп}}}{dt} &= -\frac{k_{\text{пш}}k_{\text{пт}}k_{\text{тп}}k_{\text{ш}}}{T_{\text{мт}}} \omega_{\text{д}} - \frac{k_{\text{т}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мт}}} I_{\text{д}} - \frac{1}{T_{\text{мт}}} U_{\text{тп}} - \\ &\quad - \frac{k_{\text{тп}}}{T_{\text{мт}}} U_{\text{рт}} - \frac{k_{\text{тп}}k_{\text{пт}}}{T_{\text{мт}}} U_{\text{рш}} + \frac{k_{\text{пш}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мт}}} U_{\text{з}}; \\ \frac{dU_{\text{рт}}}{dt} &= -k_{\text{ш}}k_{\text{пш}}k_{\text{іт}} \omega_{\text{д}} - k_{\text{іт}}k_{\text{т}} I_{\text{д}} + k_{\text{іт}} U_{\text{рш}} + k_{\text{пш}}k_{\text{іт}} U_{\text{з}}; \\ \frac{dU_{\text{рш}}}{dt} &= -k_{\text{ш}}k_{\text{іш}} \omega_{\text{д}} + k_{\text{іш}} U_{\text{з}}. \end{aligned} \right. \quad (3.5)$$

При записі системи (3.5) прийняті наступні позначення.

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

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

$k_{\text{пш}}, k_{\text{ш}}, k_{\text{пт}}, k_{\text{ит}}$ – коефіцієнти підсилення пропорційної і інтегральної частин регулятора швидкості і току відповідно; значення указаних параметрів можуть бути отримані з виразів передатних функцій регуляторів:

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

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

$$W_{\text{пт}}(p) = \frac{R_{\Sigma}}{2T_{\text{пт}} k_{\text{тп}} k_{\text{т}}} \frac{1 + T_{\text{е}} p}{p} = \frac{R_{\Sigma} T_{\text{е}}}{2T_{\text{пт}} k_{\text{тп}} k_{\text{т}}} + \frac{R_{\Sigma}}{2T_{\text{пт}} k_{\text{тп}} k_{\text{т}}} \frac{1}{p} = k_{\text{пт}} + \frac{k_{\text{ит}}}{p},$$

$$k_{\text{пт}} = \frac{R_{\Sigma} T_{\text{е}}}{2T_{\text{пт}} k_{\text{тп}} k_{\text{т}}}, \quad k_{\text{ит}} = \frac{R_{\Sigma}}{2T_{\text{пт}} k_{\text{тп}} k_{\text{т}}},$$

При записі системи (3.5) також замість $T_{\text{м}}$ і $k_{\text{д}}$ записані відповідні вирази

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

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

Об'єднаємо системи (3.4) і (3.5), одержимо наступну систему диференціальних рівнянь трьохмасової системи:

$$\left\{ \begin{aligned}
\frac{d\omega_{\text{м}}}{dt} &= \frac{1}{J_{\text{м}}} M_{\text{пп2}} + \frac{\beta_2}{J_{\text{м}}} \omega_{\text{к}} - \frac{\beta_2}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\
\frac{dM_{\text{пп1}}}{dt} &= c_1 \omega_{\text{д}} - c_1 \omega_{\text{к}}; \\
\frac{d\omega_{\text{к}}}{dt} &= \frac{1}{J_{\text{к}}} M_{\text{пп1}} - \frac{\beta_1}{J_{\text{к}}} \omega_{\text{д}} + \frac{\beta_1}{J_{\text{к}}} \omega_{\text{к}} - \frac{1}{J_{\text{к}}} M_{\text{пп2}} - \frac{\beta_2}{J_{\text{к}}} \omega_{\text{к}} + \frac{\beta_2}{J_{\text{к}}} \omega_{\text{м}}; \\
\frac{dM_{\text{пп2}}}{dt} &= c_2 \omega_{\text{к}} - c_2 \omega_{\text{м}}; \\
\frac{d\omega_{\text{д}}}{dt} &= \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пп1}} - \frac{\beta_1}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta_1}{J_{\text{д}\Sigma}} \omega_{\text{к}}; \\
\frac{dI_{\text{д}}}{dt} &= -\frac{k\Phi_{\text{н}}}{R_{\Sigma} T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma} T_{\text{е}}} U_{\text{тп}}; \\
\frac{dU_{\text{тп}}}{dt} &= -\frac{k_{\text{пш}} k_{\text{пт}} k_{\text{тп}} k_{\text{ш}}}{T_{\text{мт}}} \omega_{\text{д}} - \frac{k_{\text{т}} k_{\text{пт}} k_{\text{тп}}}{T_{\text{мт}}} I_{\text{д}} - \frac{1}{T_{\text{мт}}} U_{\text{тп}} - \frac{k_{\text{тп}}}{T_{\text{мт}}} U_{\text{пт}} - \frac{k_{\text{тп}} k_{\text{пт}}}{T_{\text{мт}}} U_{\text{пш}} + \frac{k_{\text{пш}} k_{\text{пт}} k_{\text{тп}}}{T_{\text{мт}}} U_3; \\
\frac{dU_{\text{пт}}}{dt} &= -k_{\text{ш}} k_{\text{пш}} k_{\text{ит}} \omega_{\text{д}} - k_{\text{ит}} k_{\text{т}} I_{\text{д}} + k_{\text{ит}} U_{\text{пш}} + k_{\text{пш}} k_{\text{ит}} U_3; \\
\frac{dU_{\text{пш}}}{dt} &= -k_{\text{ш}} k_{\text{иш}} \omega_{\text{д}} + k_{\text{иш}} U_3.
\end{aligned} \right.$$

Алгоритмічна схема трьохмасової системи показана на рис. 3.2.

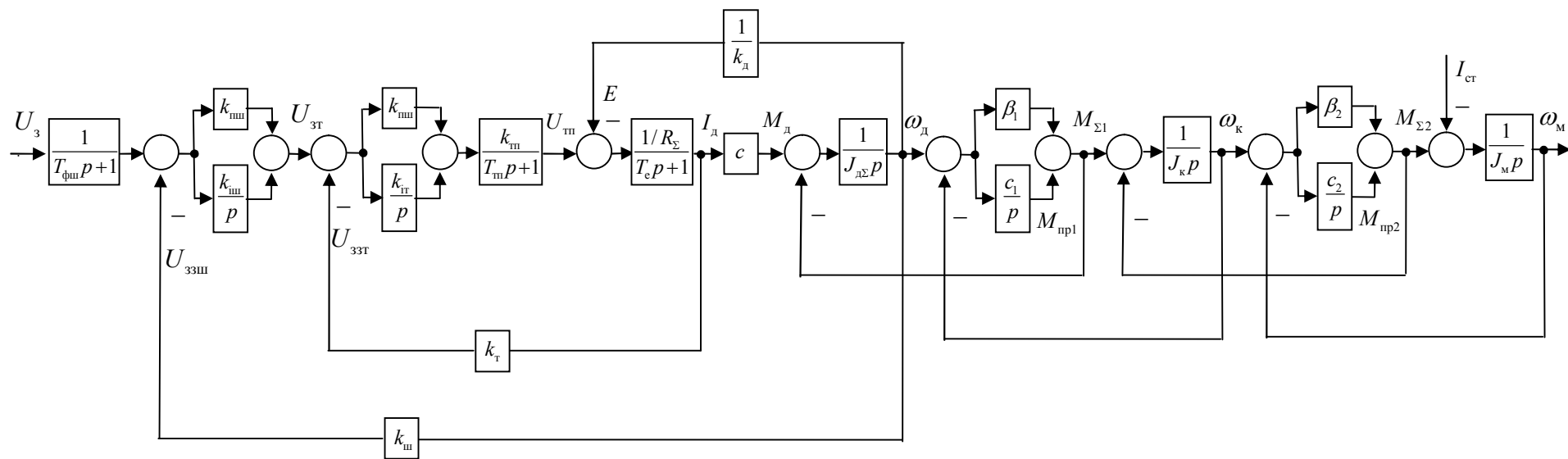


Рисунок 3.2 – Алгоритмічна схема трьохмасової системи

Введемо вектор стану трьохмасової системи:

$$\vec{X}(t) = [\omega_m(t), M_{np2}(t), \omega_k(t), M_{np1}(t), \omega_d(t), I_d(t), U_{тп}(t), U_{рт}(t), U_{рш}(t)]^T$$

і вектор зовнішніх дій

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

Запишемо рівняння стану системи, доповнивши його рівнянням виходу системи

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

де $\mathbf{A}$ – матриця стану;

$\mathbf{B}$ – матриця управління;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ – вектор виходу.

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

Структура матриць **A** і **B** трьохмасової системи

	$\omega_{\text{м}}$	M_{np2}	$\omega_{\text{к}}$	M_{np1}	$\omega_{\text{д}}$	$I_{\text{д}}$	$U_{\text{тп}}$	$U_{\text{рт}}$	$U_{\text{рш}}$		U_3	$M_{\text{ст}}$
A=	$-\frac{\beta_2}{J_{\text{м}}}$	$\frac{1}{J_{\text{м}}}$	$\frac{\beta_2}{J_{\text{м}}}$	0	0	0	0	0	0		0	$-\frac{1}{J_{\text{м}}}$
	$-c_2$		c_2		0	0	0	0	0		0	0
	$\frac{\beta_2}{J_{\text{к}}}$	$-\frac{1}{J_{\text{к}}}$	$-\frac{\beta_1}{J_{\text{к}}} - \frac{\beta_2}{J_{\text{к}}}$	$\frac{1}{J_{\text{к}}}$	$\frac{\beta_1}{J_{\text{к}}}$	0	0	0	0		0	0
	0	0	$-c_1$	0	c_1	0	0	0	0		0	0
	0	0	$\frac{\beta_1}{J_{\text{д}\Sigma}}$	$-\frac{1}{J_{\text{д}\Sigma}}$	$-\frac{\beta_1}{J_{\text{д}\Sigma}}$	$\frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}}$	0	0	0	B=	0	0
	0	0	0	0	$-\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}}$	$-\frac{1}{T_{\text{е}}}$	$\frac{1}{R_{\Sigma}T_{\text{е}}}$	0	0		0	0
	0	0	0	0	$-k_{\text{шш}}k_{\text{пт}}k_{\text{тп}}\frac{k_{\text{ш}}}{T_{\text{мт}}}$	$-\frac{k_{\text{т}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мт}}}$	$-\frac{1}{T_{\text{мт}}}$	$\frac{k_{\text{тп}}}{T_{\text{мт}}}$	$\frac{k_{\text{тп}}k_{\text{пт}}}{T_{\text{мт}}}$		$\frac{k_{\text{шш}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мт}}}$	0
	0	0	0	0	$-k_{\text{ш}}k_{\text{шш}}k_{\text{ит}}$	$-k_{\text{ит}}k_{\text{т}}$	0	0	$k_{\text{ит}}$		$k_{\text{шш}}k_{\text{ит}}$	0
	0	0	0	0	$-k_{\text{ш}}k_{\text{шш}}$	-0	0	0	0		$k_{\text{шш}}$	0

3.2 Розрахунок перехідних процесів трьохмасової системи

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

Відносні приведені маси: завантаженого скипа μ_2 , каната μ_k і двигуна з підйомним барабаном μ_1

$$\mu_1 = 0,73 \quad \mu_2 = 0,2 \quad \mu_k = 0,07.$$

Сумарна приведена маса

$$m = 116792 \text{ кг}.$$

Приведений до валу двигуна момент інерції механізму

$$J_{\Sigma} = 729823 \text{ кг} \cdot \text{м}^2.$$

Значення приведених мас

$$m_1 = \mu_1 m = 0,73 \cdot 116792 = 85258 \text{ кг}$$

$$m_2 = \mu_2 m = 0,2 \cdot 116792 = 23358 \text{ кг}$$

$$m_k = \mu_k m = 0,07 \cdot 116792 = 8175 \text{ кг}$$

Радіус приведення поступальної ходи до обертальної:

$$\rho = \frac{V_n}{\omega_n} = R_0 = 2,5.$$

Сумарні приведені моменти інерції рівні:

$$J_{\Sigma} = m_1 \rho^2 = 85258 \cdot 2,5^2 = 532864 \text{ кг} \cdot \text{м}^2$$

$$J_{\text{м}} = m_2 \rho^2 = 23358 \cdot 2,5^2 = 145990 \text{ кг} \cdot \text{м}^2$$

$$J_{\text{к}} = m_{\text{к}} \rho^2 = 8175 \cdot 2,5^2 = 51097 \text{ кг} \cdot \text{м}^2$$

Для двомасової системи

$$J'_{\text{д}\Sigma} = J_{\text{д}\Sigma} + \frac{J_{\text{к}}}{2} = 532864 + \frac{51097}{2} = 558412 \text{ кг} \cdot \text{м}^2$$

$$J'_{\text{м}} = J_{\text{м}} + \frac{J_{\text{к}}}{2} = 145990 + \frac{51097}{2} = 171539 \text{ кг} \cdot \text{м}^2$$

Жорсткість каната визначаємо із співвідношення

$$\omega_0 = \sqrt{\frac{c_{\text{к}}}{m'_2}},$$

отже

$$c_{\text{к}} = \omega_0^2 m'_2.$$

де ω_0 – частота власних коливань підйомної судини; з літератури, наприклад [52], 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):

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

$$m'_2 = \left(0,2 + \frac{0,07}{2} \right) \cdot 116792 = 27446 \text{ кг}$$

$$c_{\text{к}} = 3^2 \cdot 27446 = 247015 \text{ Н} \cdot \text{м}$$

Приведена жорсткість підйомного каната

$$c = c_k \rho^2 = 247015 \cdot 2,5^2 = 1543844 \text{ Н} \cdot \text{м}.$$

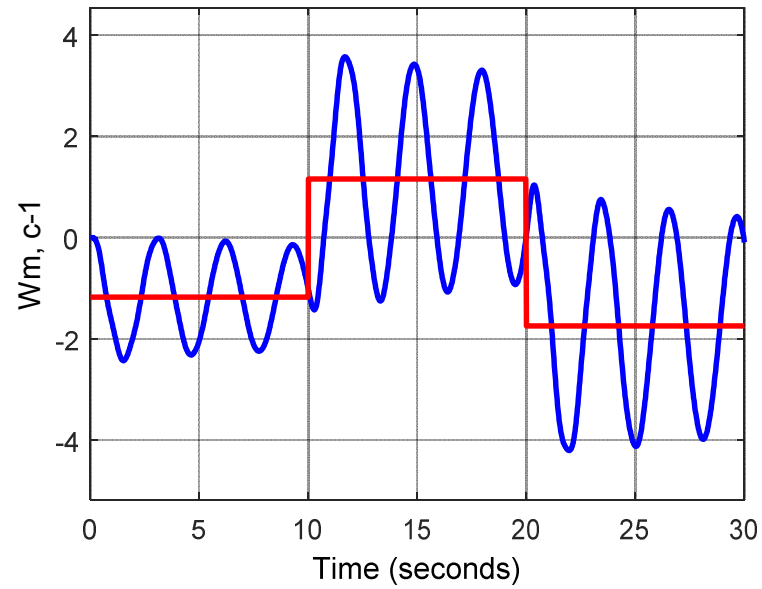
Приведені жорсткості ділянок каната між масами для трьохмасової системи

$$c_1 = c_2 = 2c_k = 2 \cdot 1543844 = 3087689 \text{ Н} \cdot \text{м}.$$

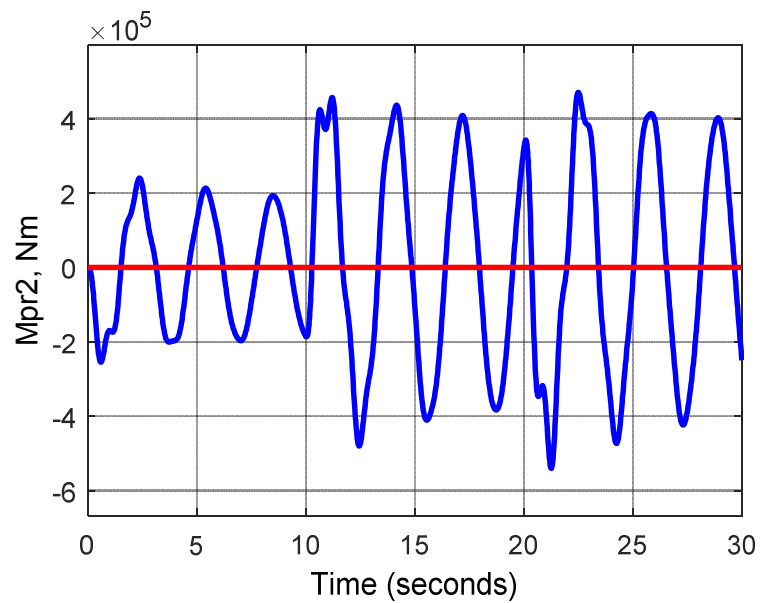
Для дослідження динамічних характеристик трьохмасової системи використовуємо пакет прикладних програм MATLAB. Розрахунок виконуємо як за допомогою моделі, зображеної на рис.3.3, так і по векторно-матричному рівнянню (3.6).

Для зменшення інформації, що виводиться, на екран монітора виводимо перші шість змінних стану системи по задаючій і збурюючій дії. Відповідні графіки приведені на рис.3.4 -3.9. Як видно з графіків, перехідні процеси швидкості другої і третьої маси, а також моментів пружності мають характер слабозатухаючих коливань. По збурюючій дії перехідні процеси всіх змінних стану мають коливальний характер.

У додатку Д приведений текст файлу, що містить початкові дані і команди розрахунку.

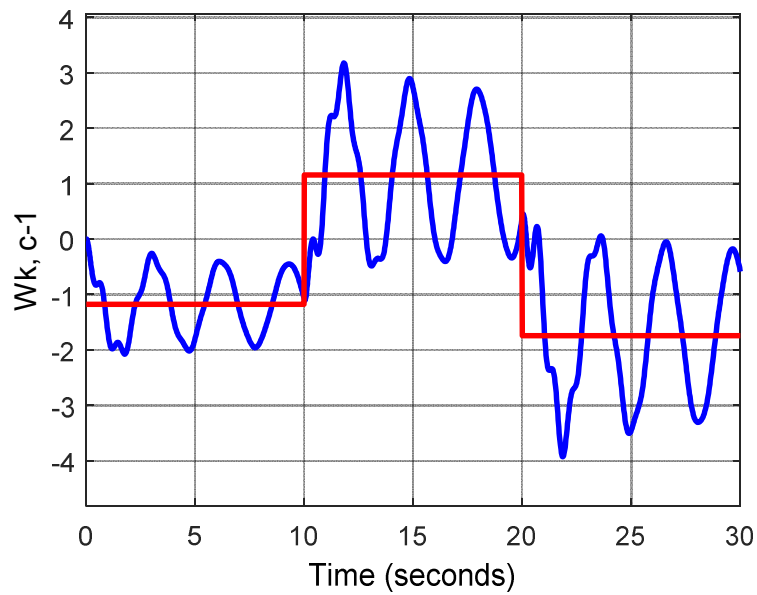


a)

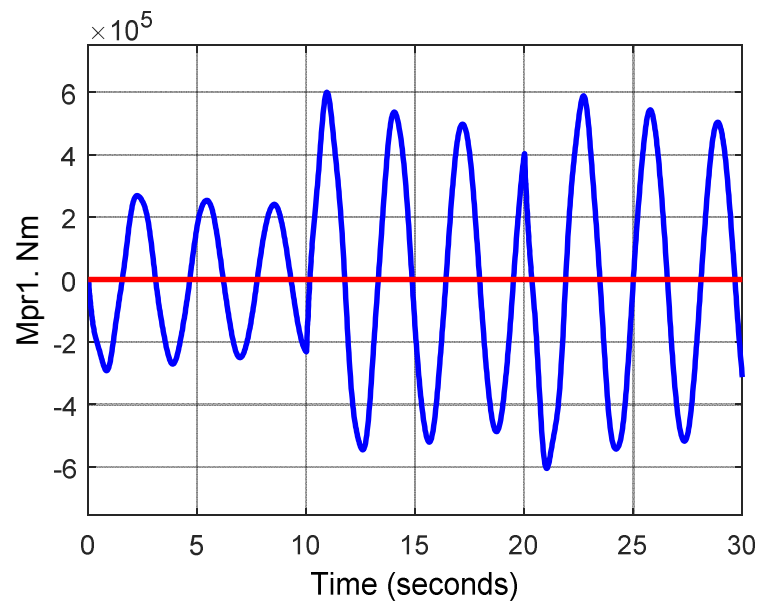


б)

Рисунок 3.4 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр2} = f(t)$ (б) трьохмасової системи по задаючій дії

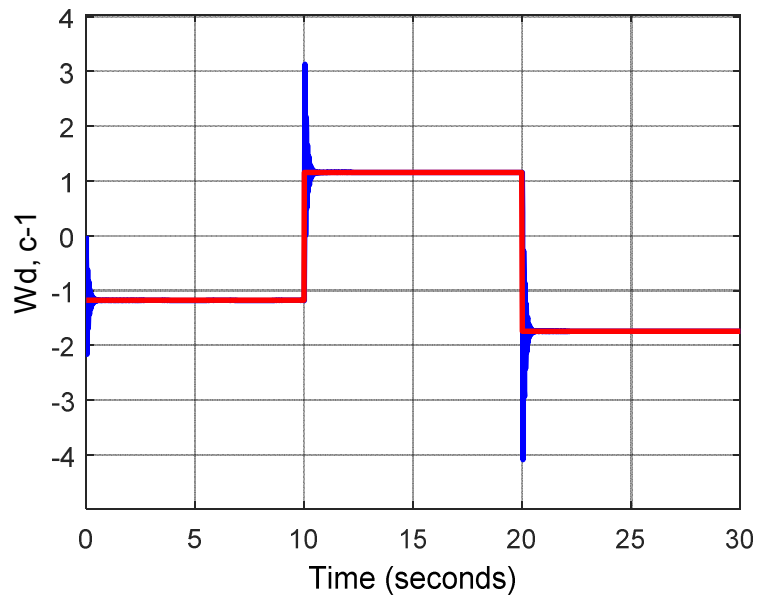


a)

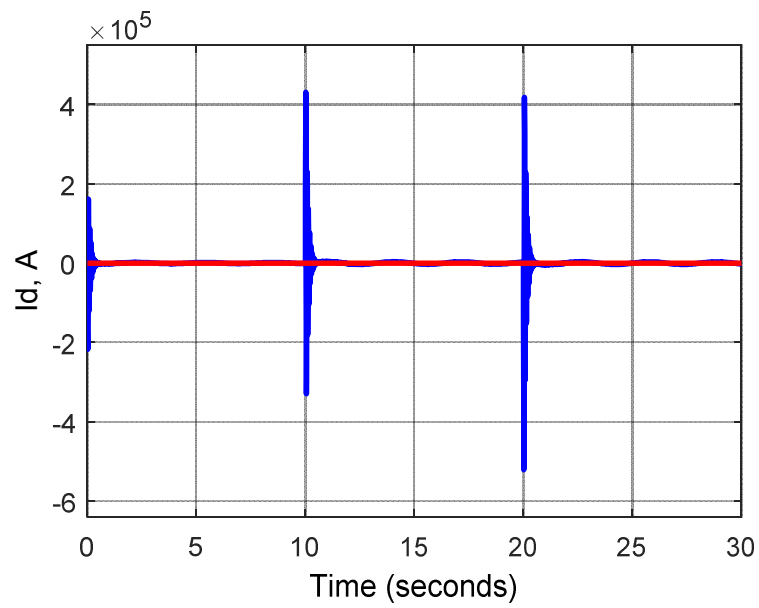


б)

Рисунок 3.5 – Графіки перехідних процесів швидкості каната $\omega_k = f(t)$ (а) і моменту пружності $M_{\text{пр1}} = f(t)$ (б) трьохмасової системи по задаючій дії

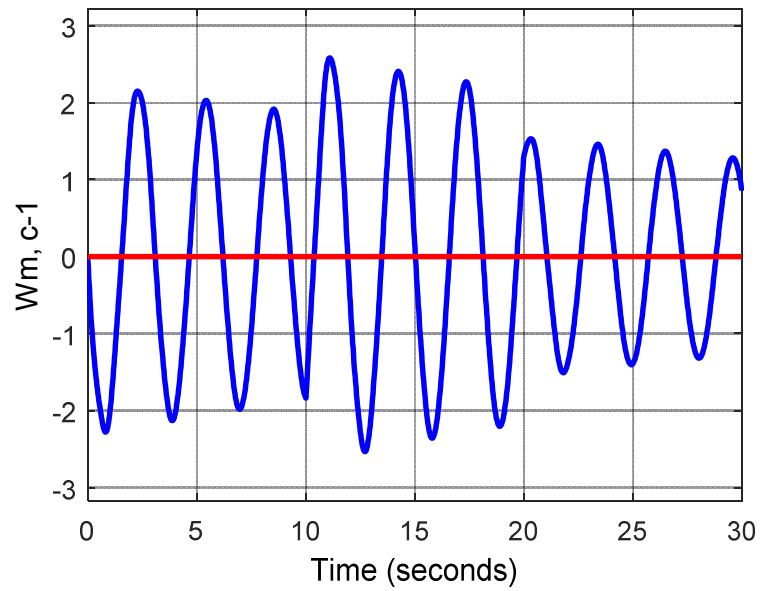


a)

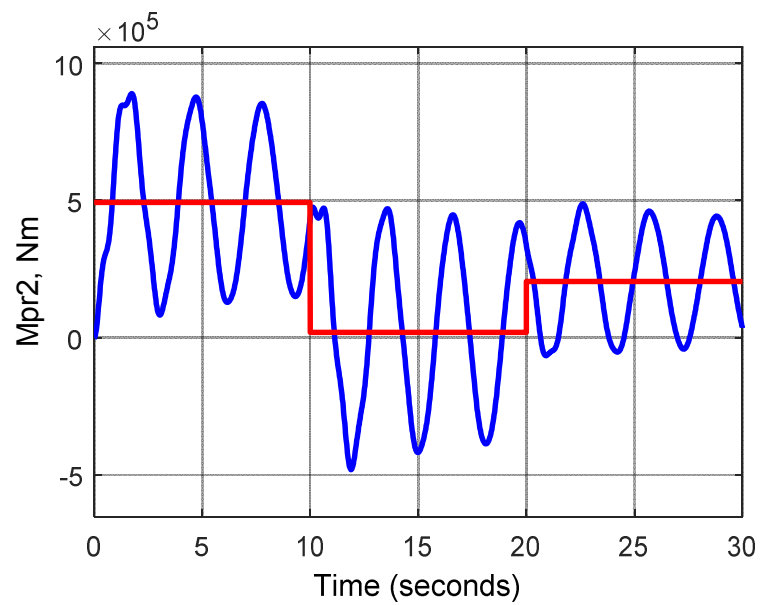


б)

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

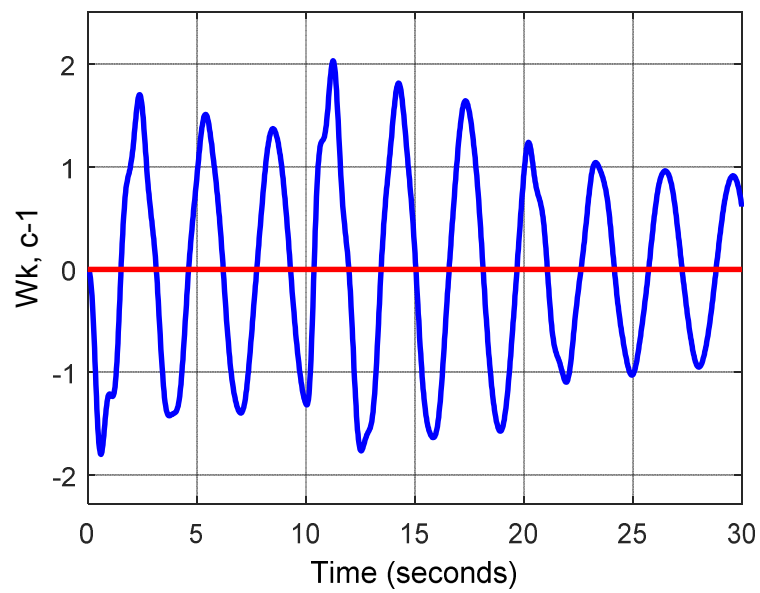


a)

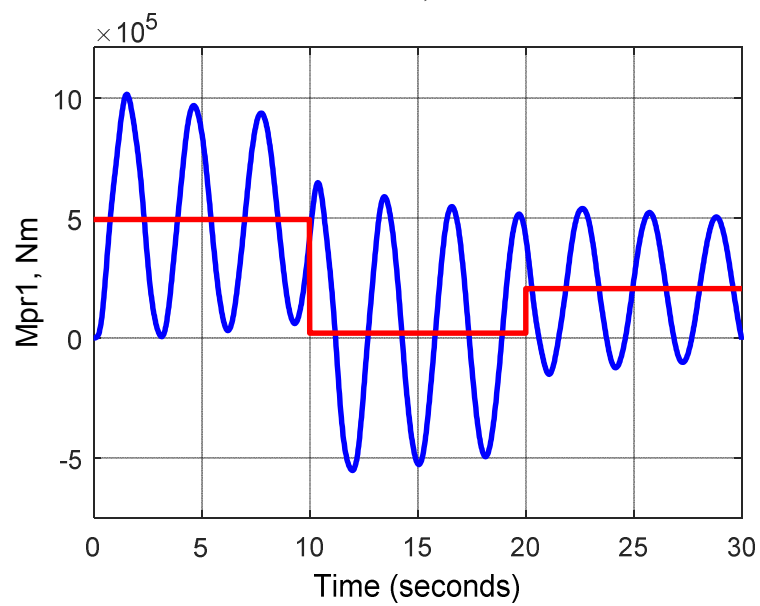


б)

Рисунок 3.7 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр2} = f(t)$ (б) трьохмасової системи по збурюючій дії



a)



б)

Рисунок 3.8 Графіки перехідних процесів швидкості каната $\omega_k = f(t)$ (а) і моменту пружності $M_{\text{пр1}} = f(t)$ (б) трьохмасової системи по збурюючій дії

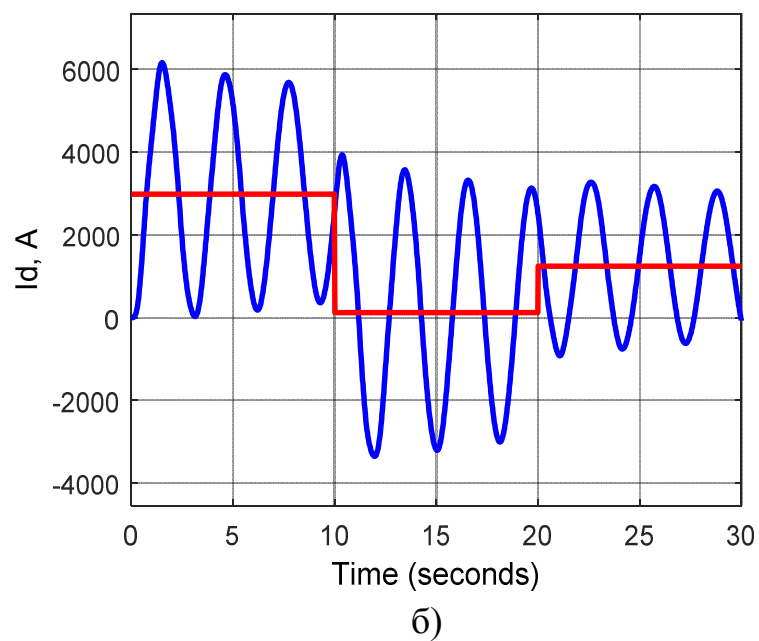
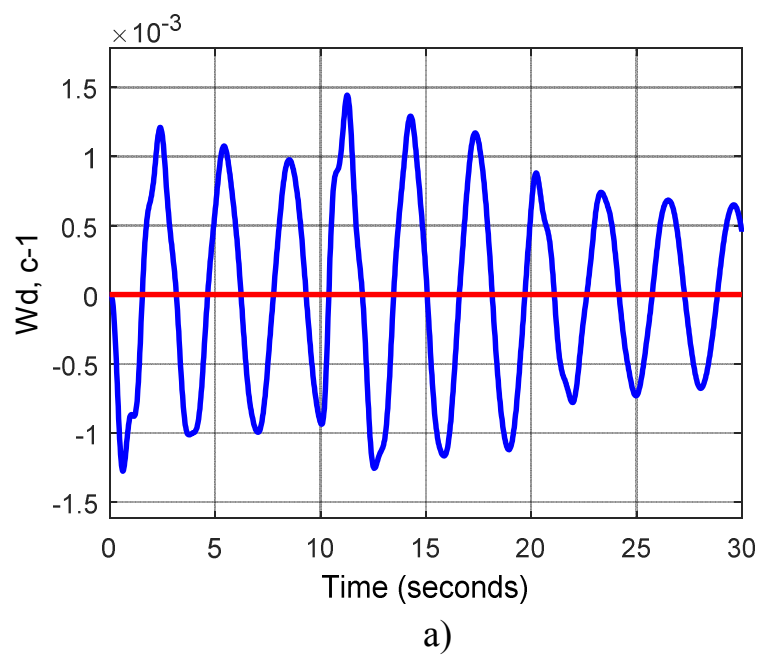


Рисунок 3.9 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) трьохмасової системи по збурюючій дії

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 трьохмасової системи. Отримано рівняння стану трьохмасової системи і рівняння стану трьохмасової 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 трьохмасової системи.

3. Проведено моделювання трьохмасової системи на ЕОМ з використанням пакету прокладних програм MATLAB. В режимі SIMULINK системи MATLAB розроблено схему моделі трьохмасової 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).

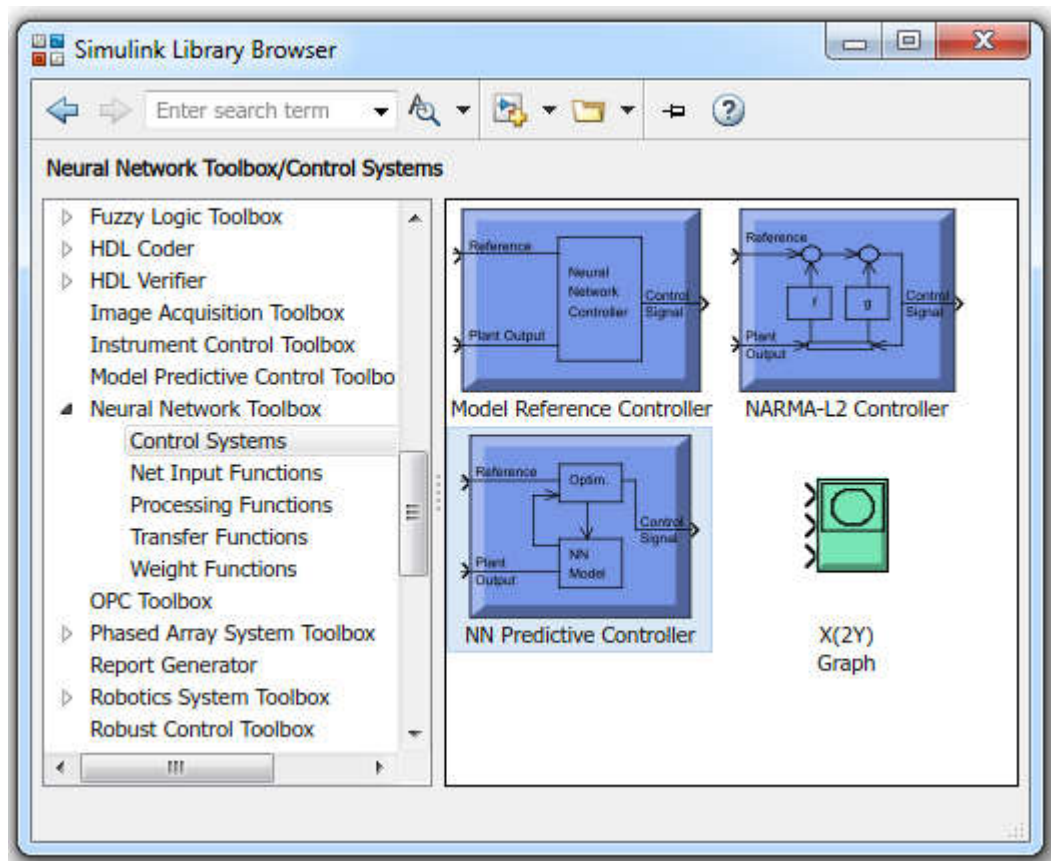


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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.

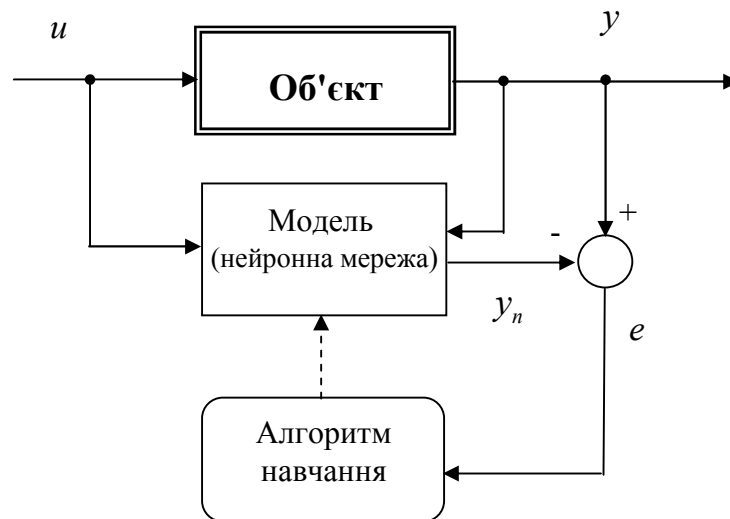


Рисунок 4.2 – Схема підсистеми ідентифікації.

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**.

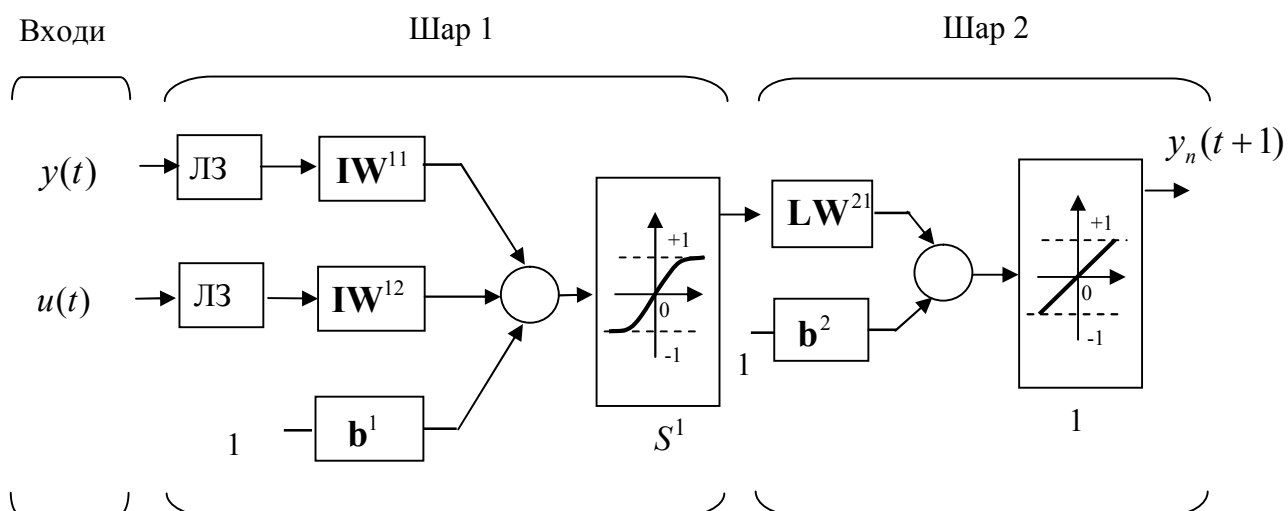


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

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.

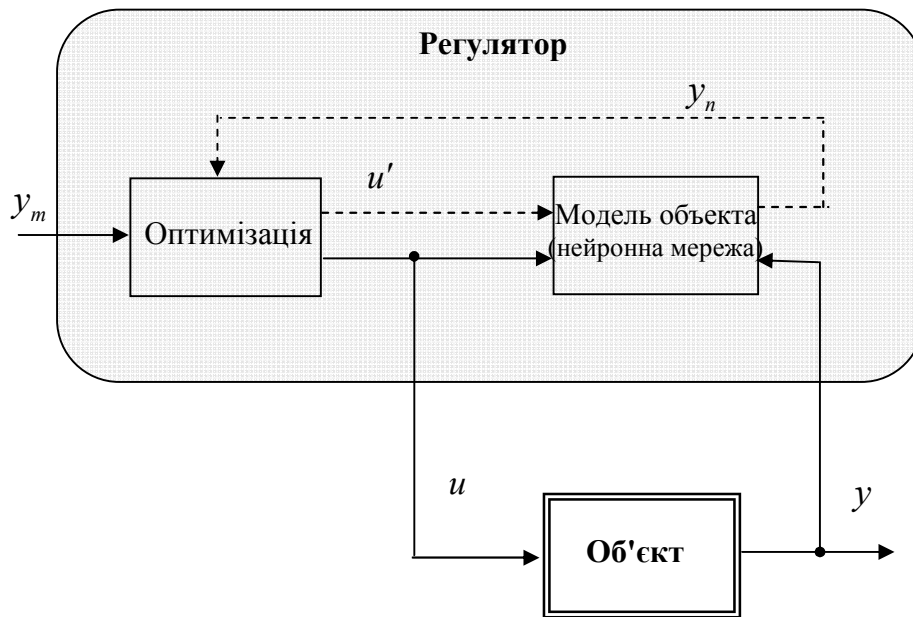


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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.

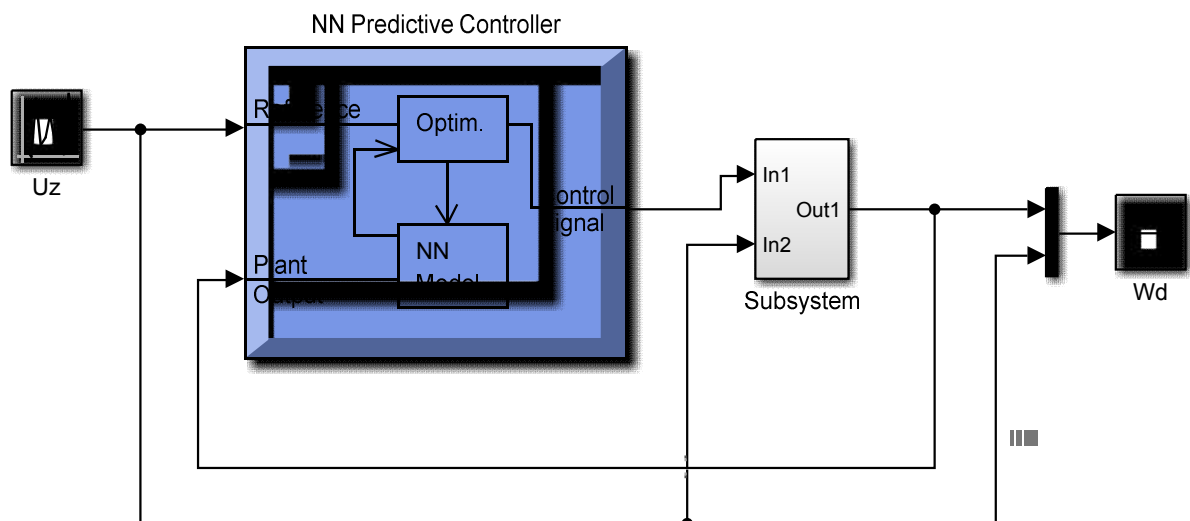


Рисунок 4.5 – Схема системи управління з нейрорегулятором

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-

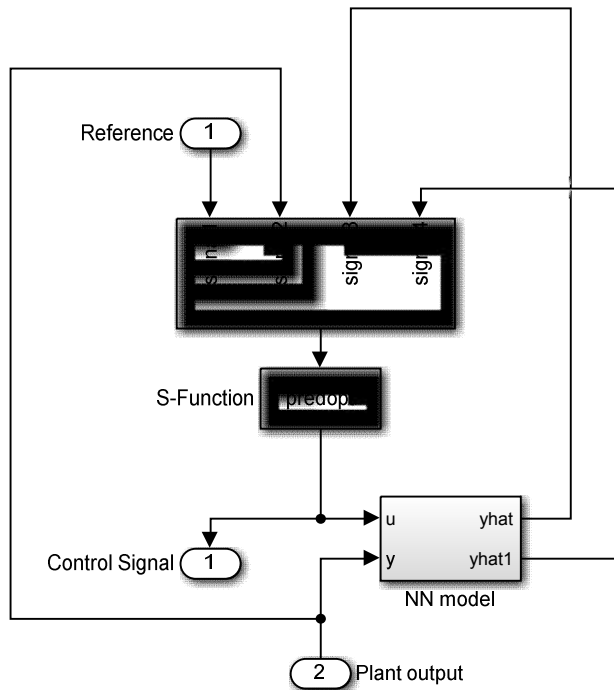


Рисунок 4.6 – Структурна схема нейрорегулятора

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**.

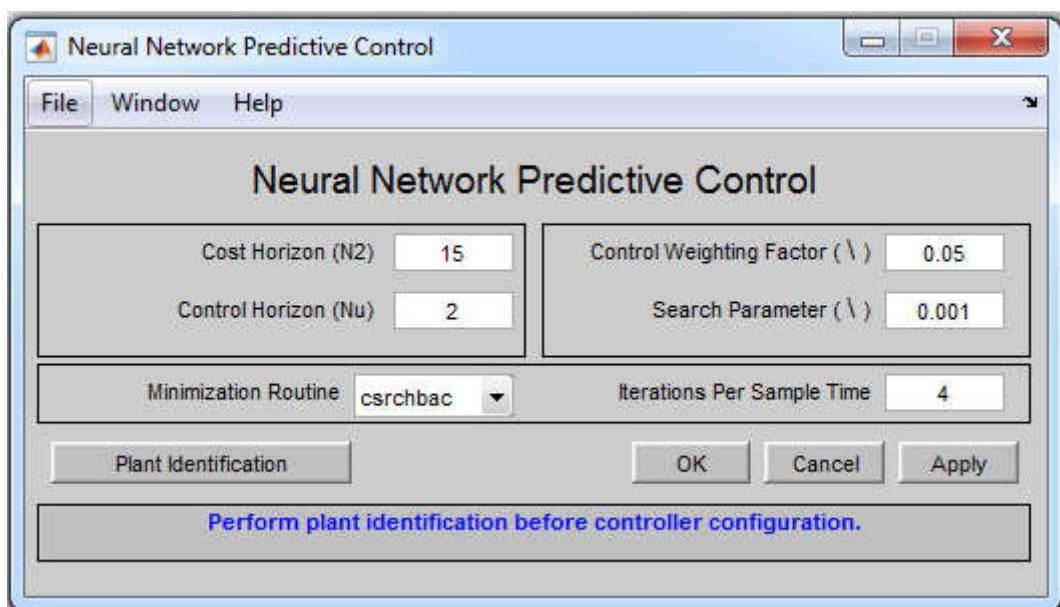


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

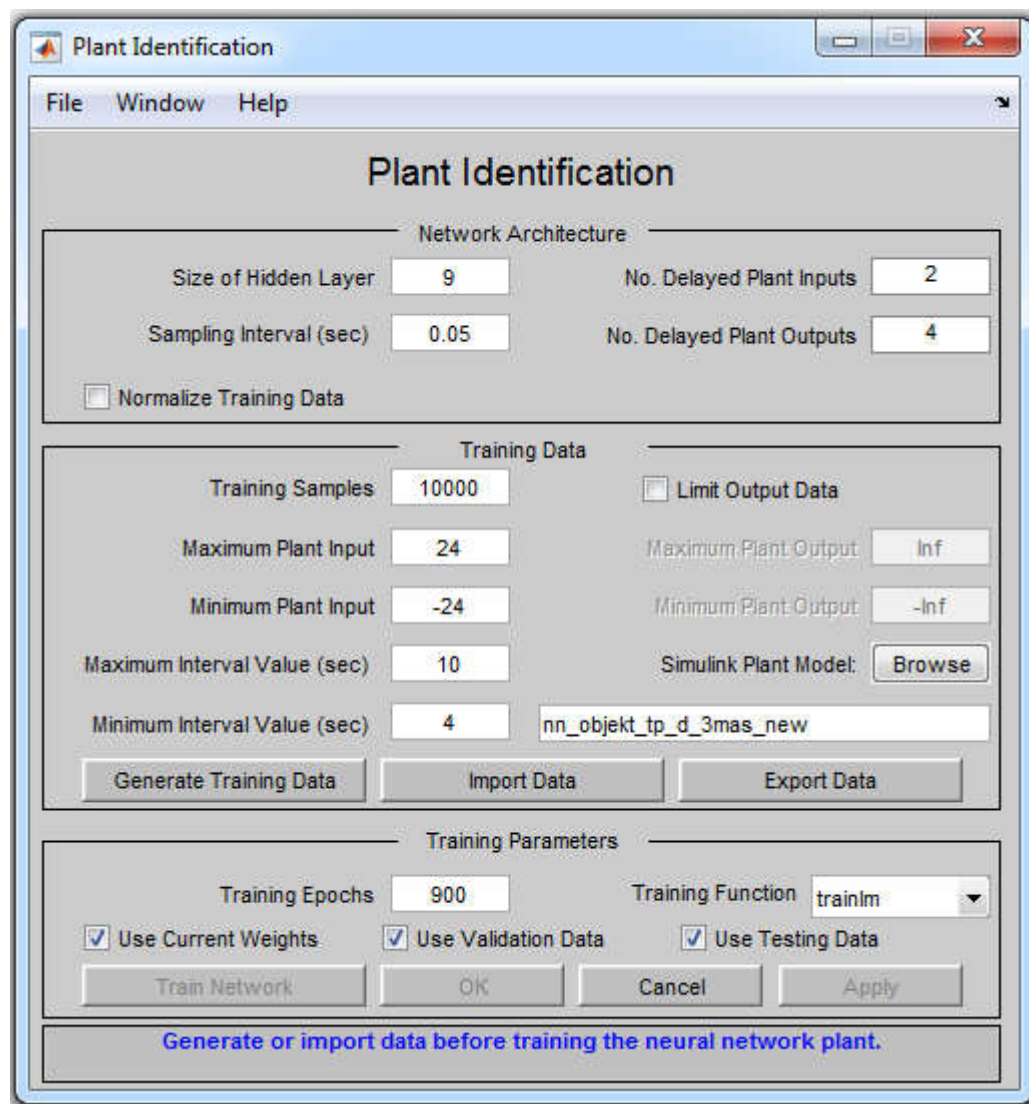


Рисунок 4.8 – Вікно ідентифікації керованої системи

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.

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

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

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

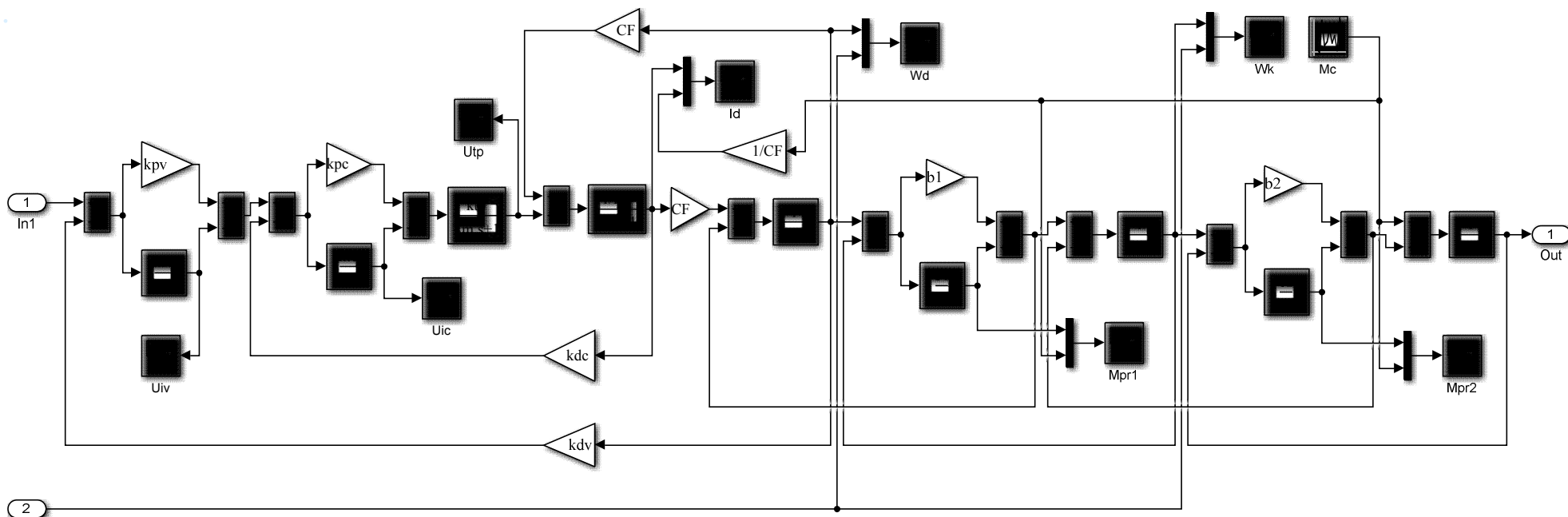


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

Секція Training Parameters (параметри навчання). Training Parameters

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

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

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

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

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

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

найменших квадратів Левенберга-Марквардта; trainbr - байєсівський регулятор.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

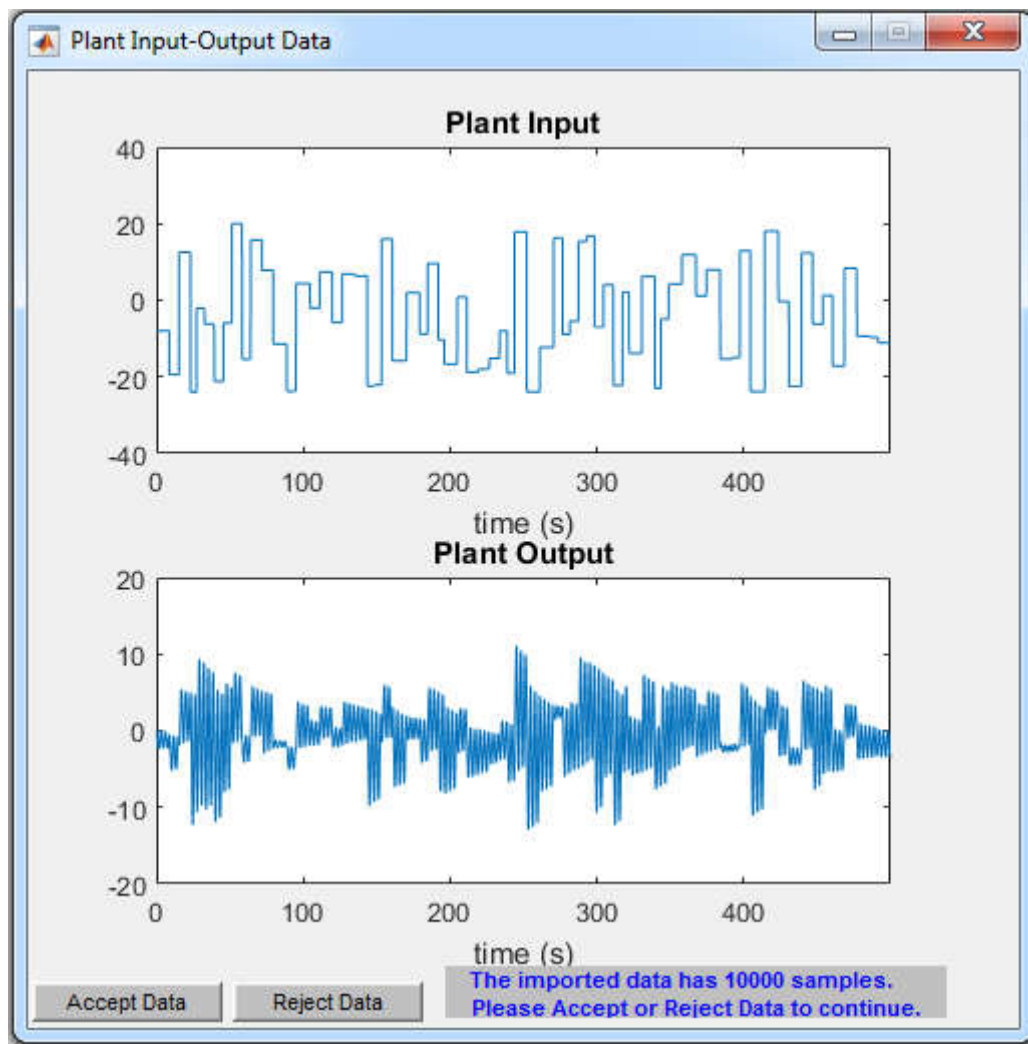


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

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

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

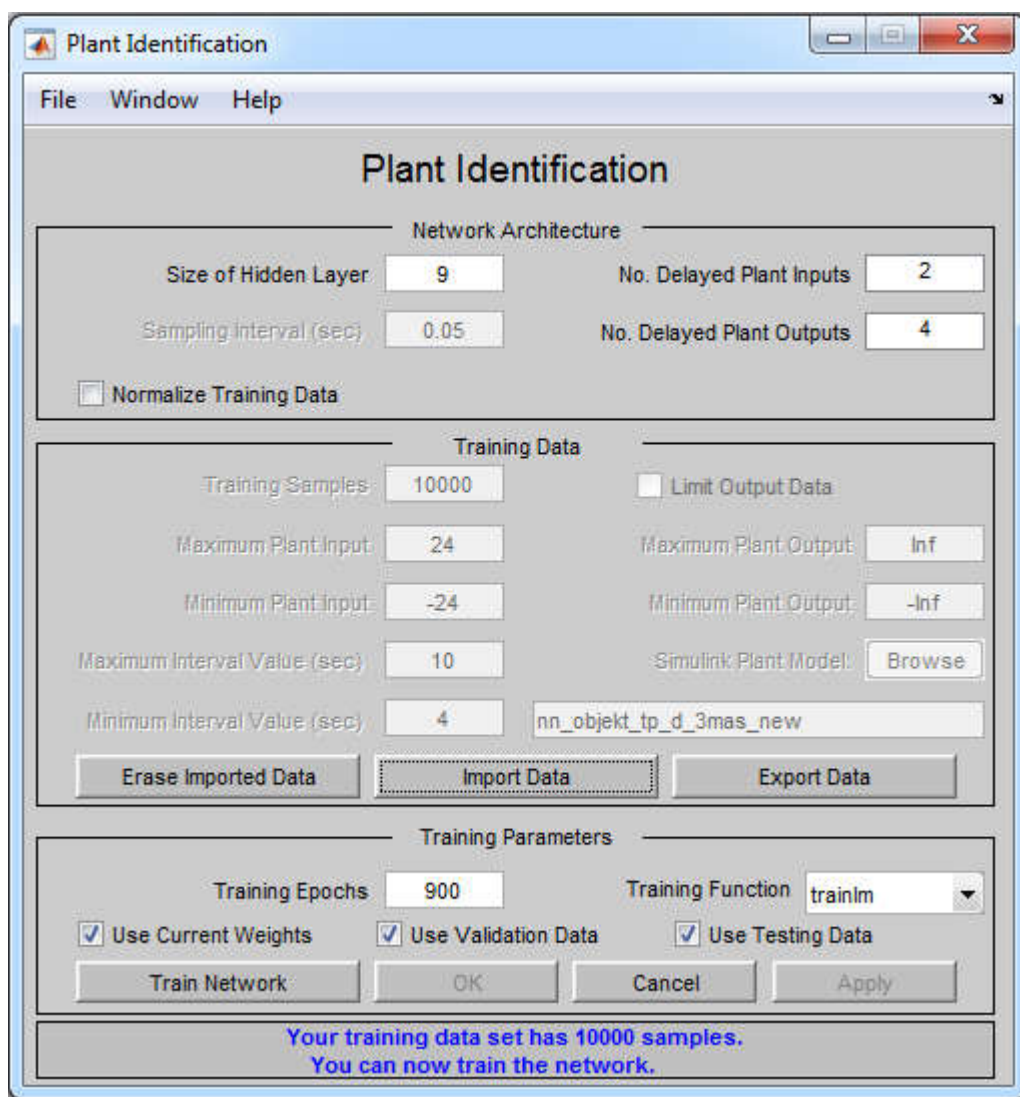


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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

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

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

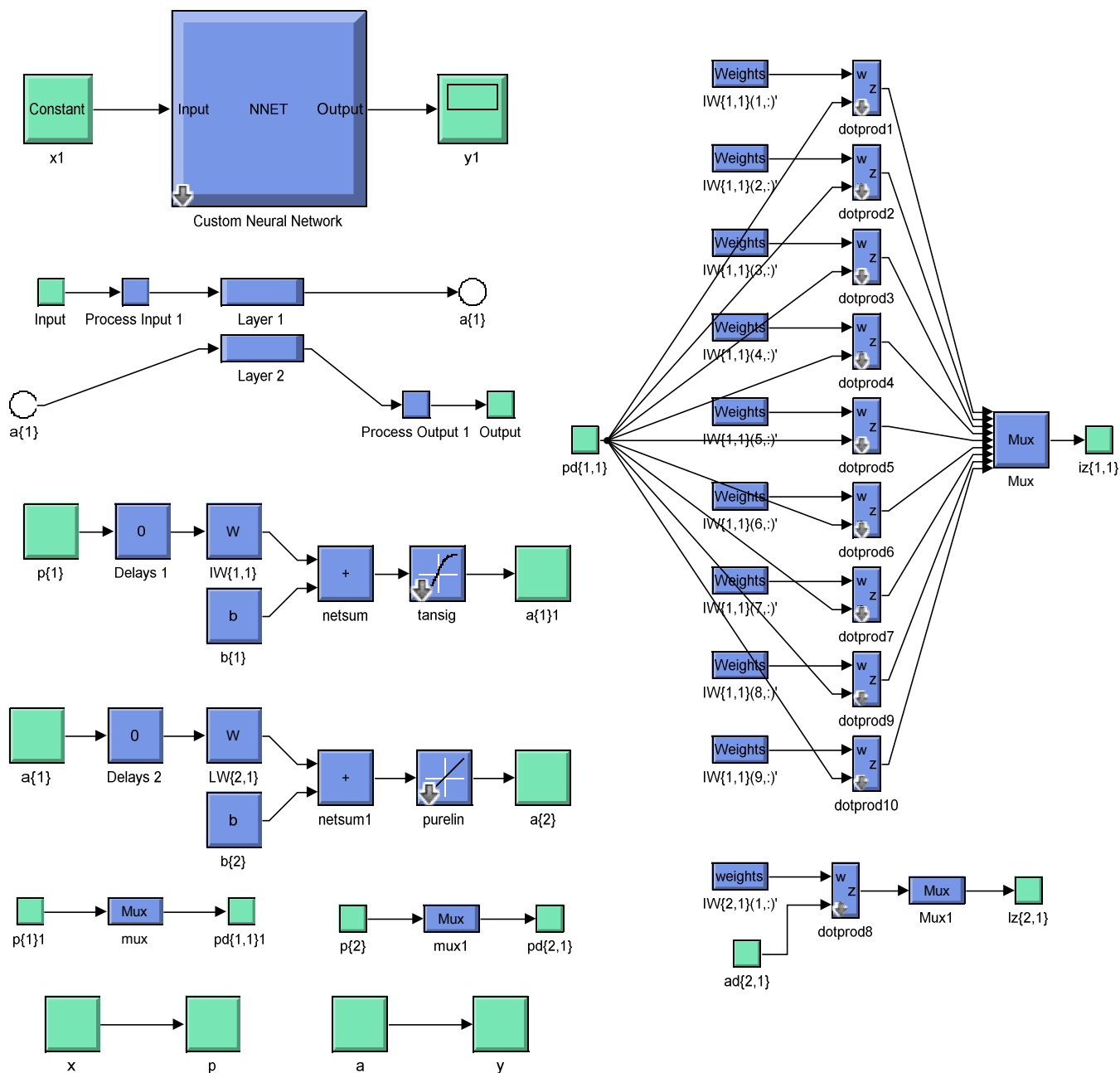


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

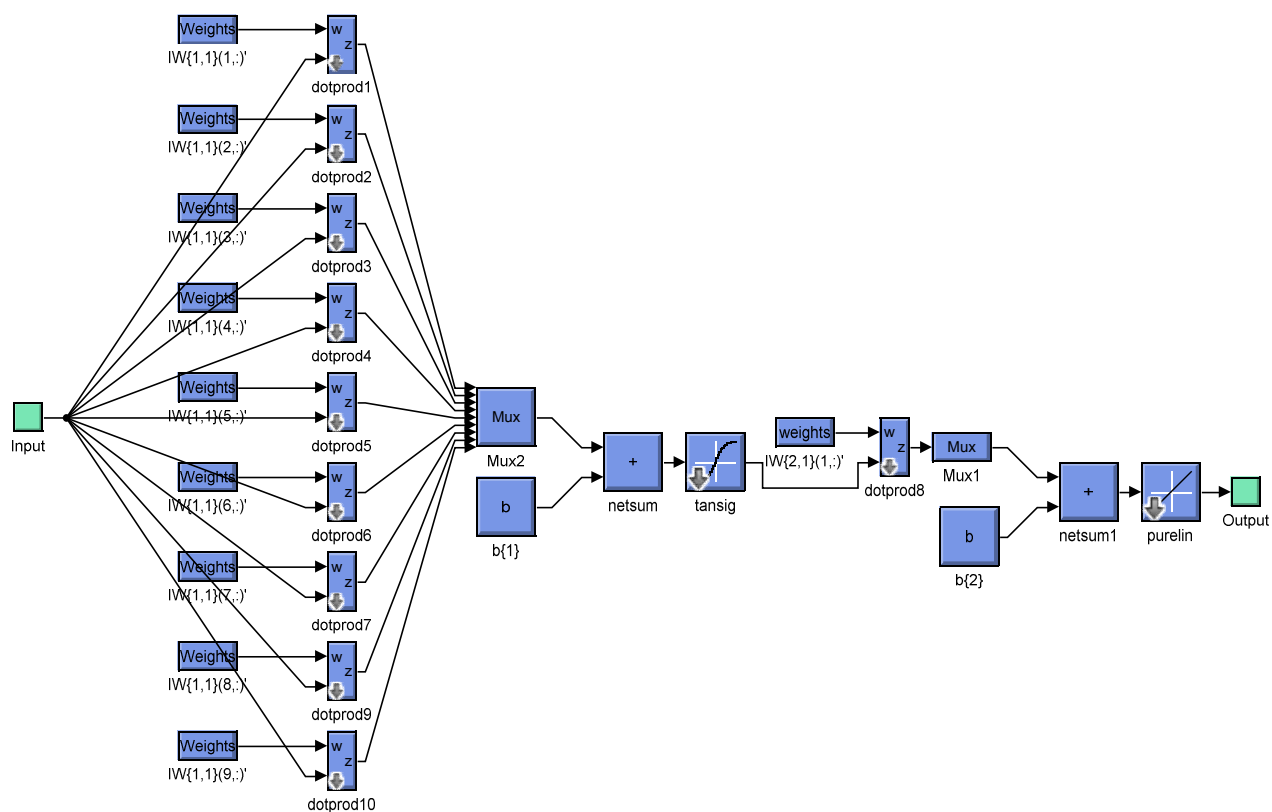


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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.

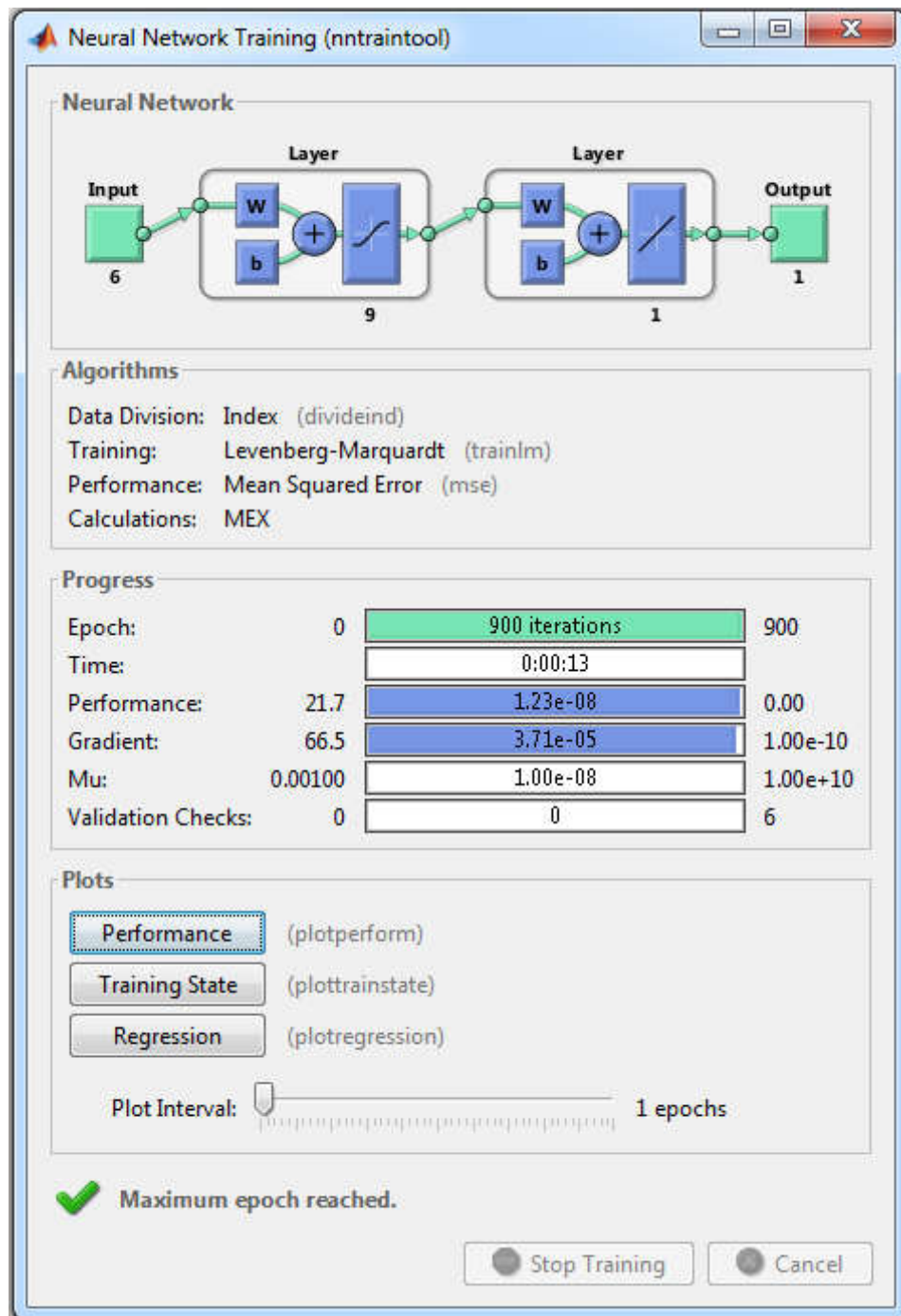


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

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 перестав покращуватися або погіршується, то це може бути індикатором перенавчання, і навчання може бути припинене.

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

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

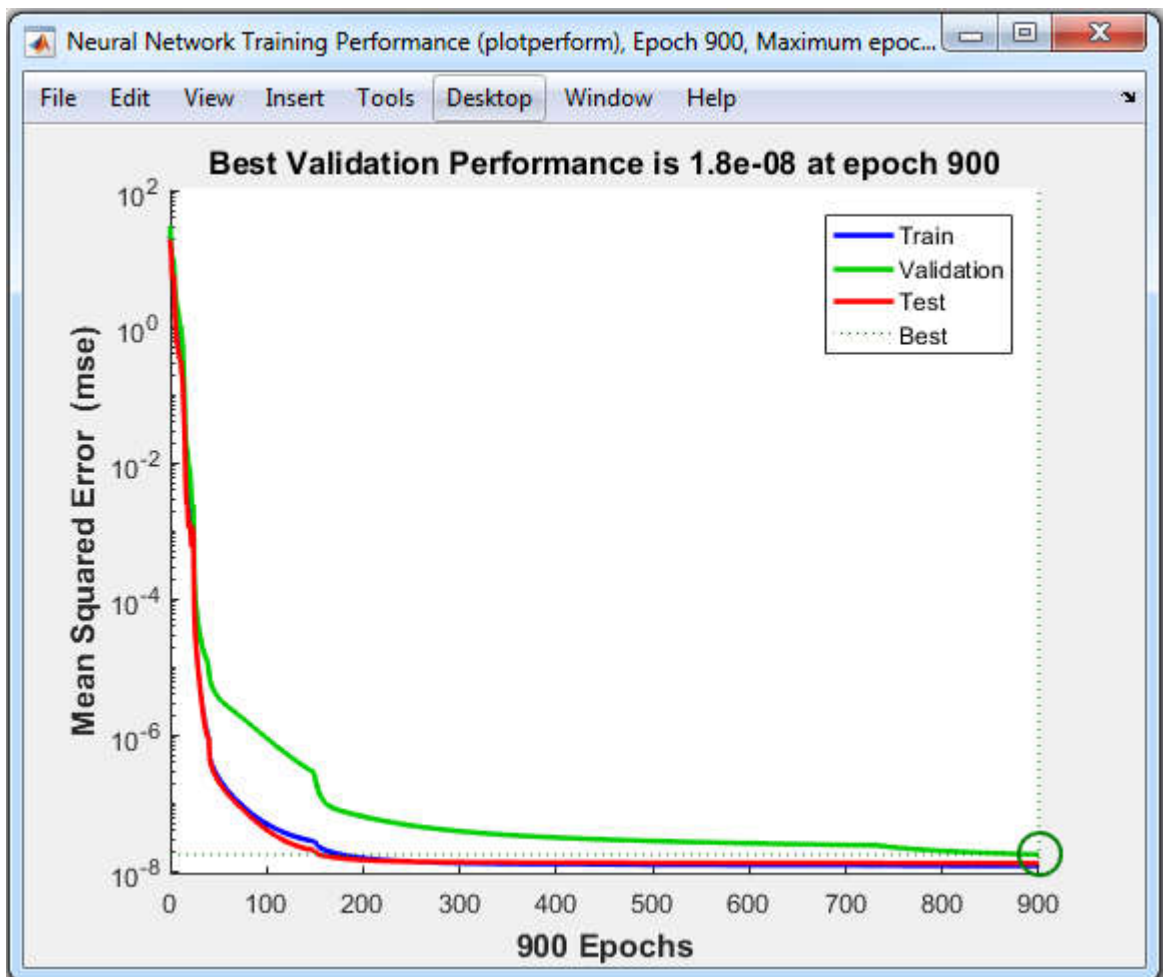


Рисунок 4.15 – Вікно контролю процесу навчання

У вікні Neural Network Training Performance відображаються графіки **Mean Squared Error (mse)**, які показують, як змінюється середньоквадратична

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

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

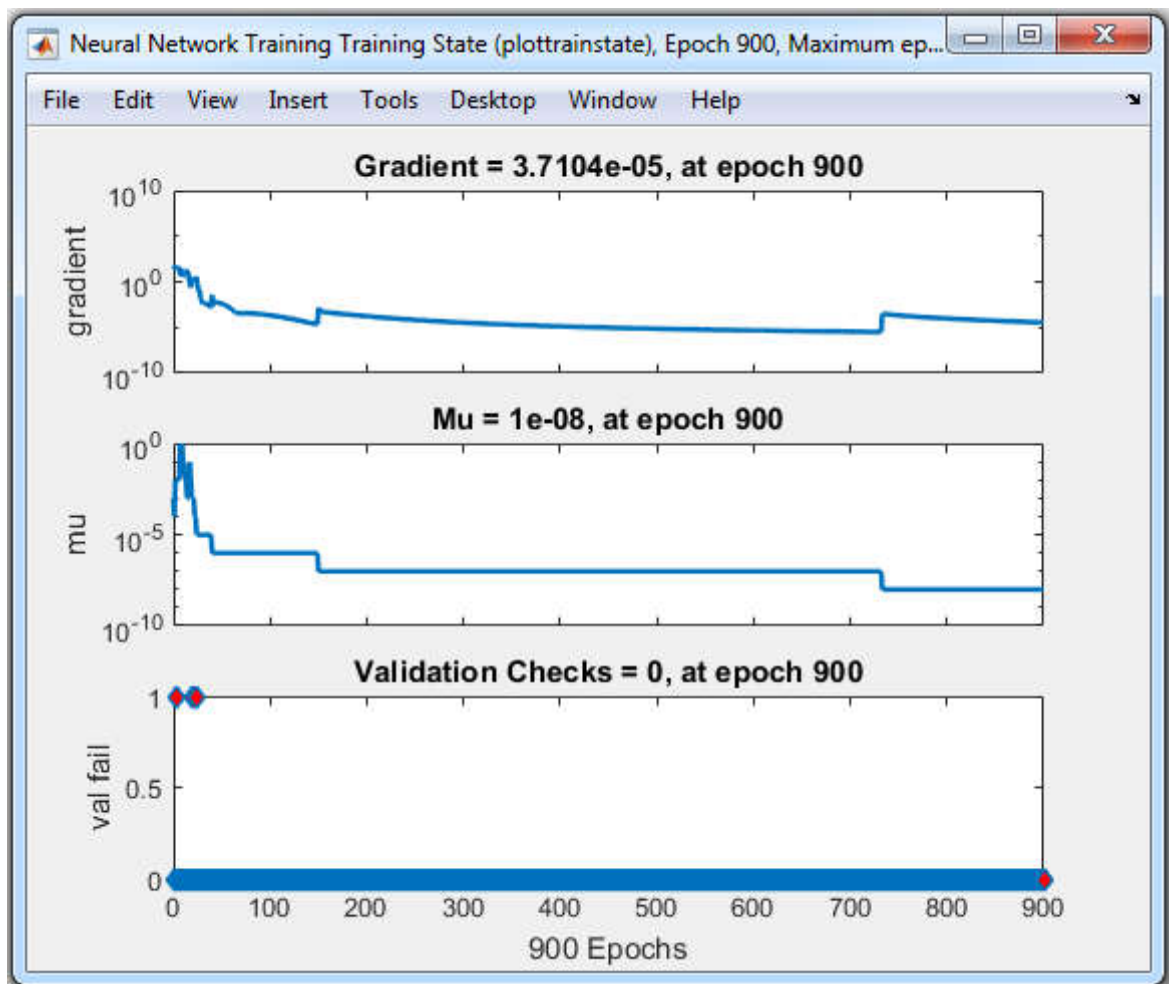


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні Neural Network Training State відображаються наступні графіки.

Графік Gradient показує зміну значення градієнта під час навчання нейронної мережі. Градієнт в контексті навчання нейронних мереж є вектором, який вказує напрямок та міру найшвидшого зростання функції втрати (або помилки) мережі в точці навчання. Градієнт використовується для оновлення вагових коефіцієнтів мережі під час оптимізації.

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

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

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

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

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

Зміна μ показує, як модифікується швидкість навчання під час процесу навчання. Зазвичай, μ може збільшуватися або зменшуватися в залежності від реакції нейронної мережі на дані. Наприклад, велике значення " μ " може вка-

зувати 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 входних даних. У регресійних завданнях модель намагається знайти функціональну залежність між входними ознаками (параметрами) і вихідними числовими значеннями, які зазвичай називаються цільовими або відгуком. .

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

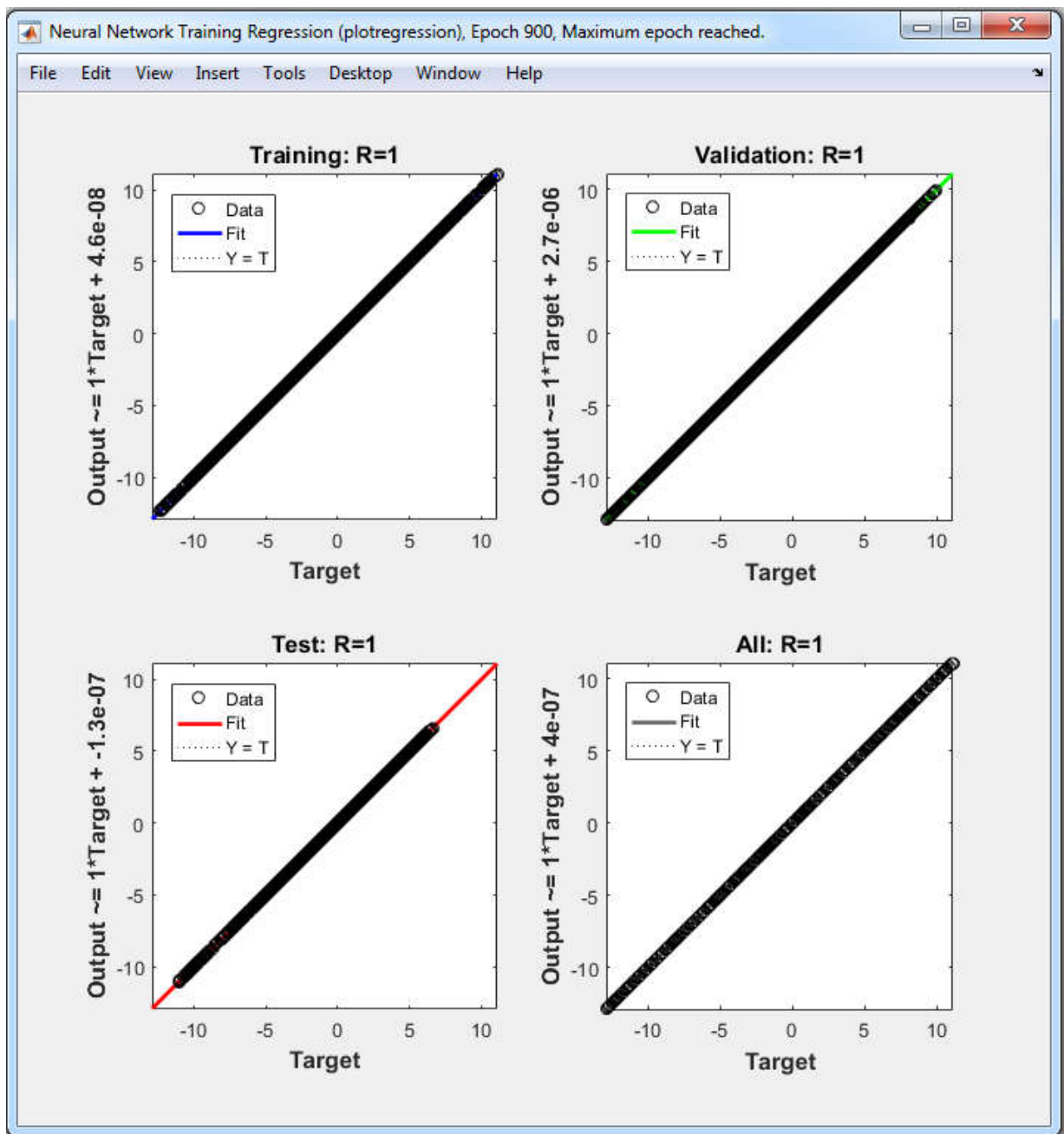


Рисунок 4.17 - Вікно Neural Network Training Regression

Onuc вікна Neural Neural Network Training Regression

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

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

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

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

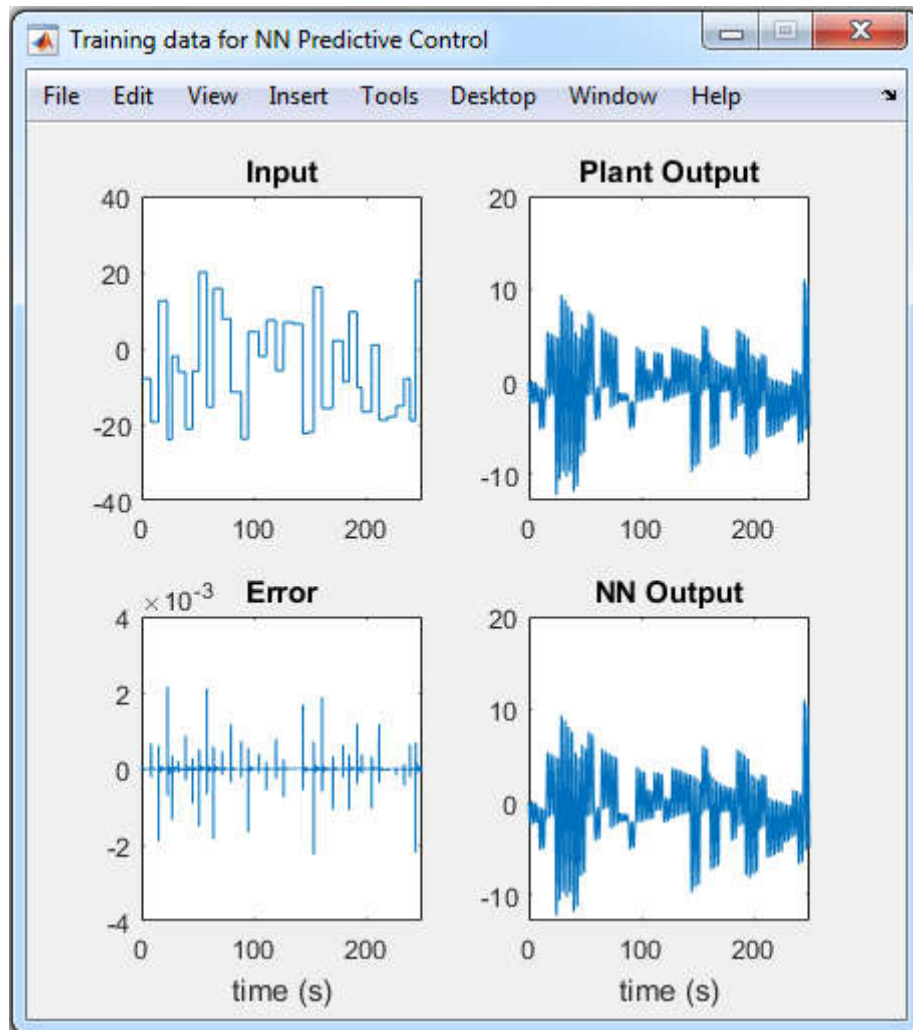


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

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

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

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

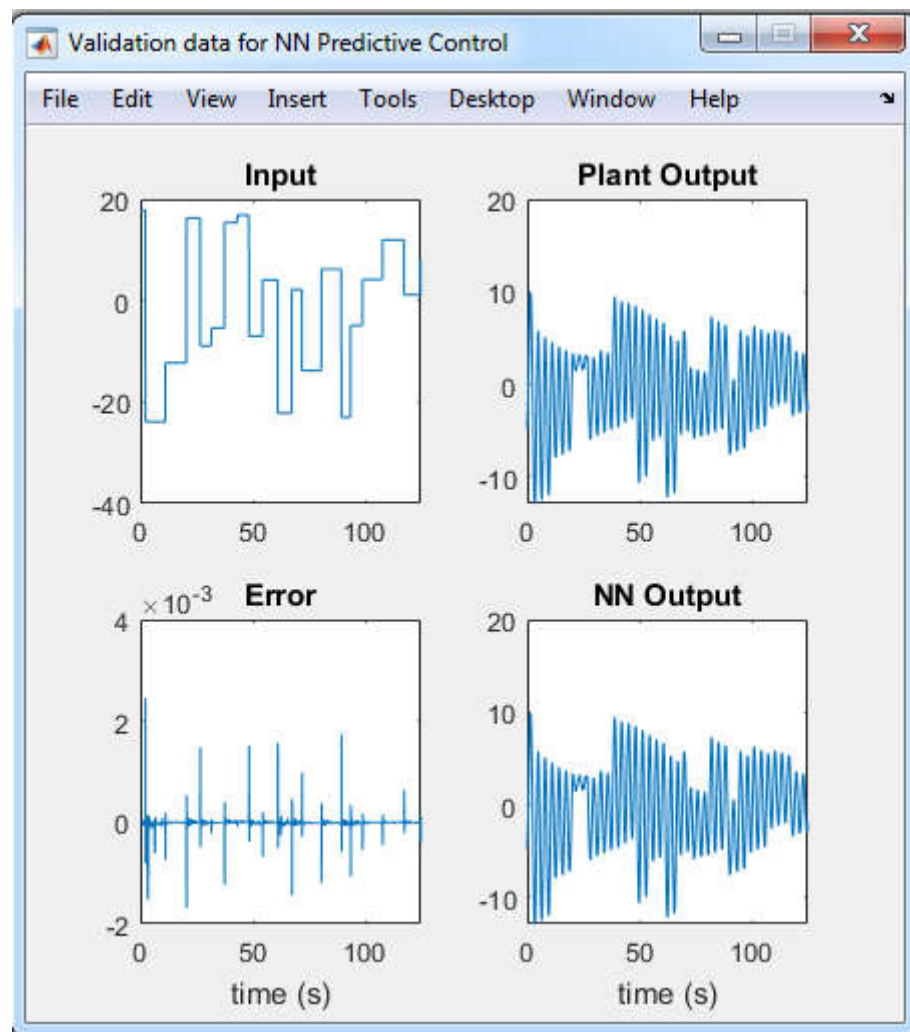


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

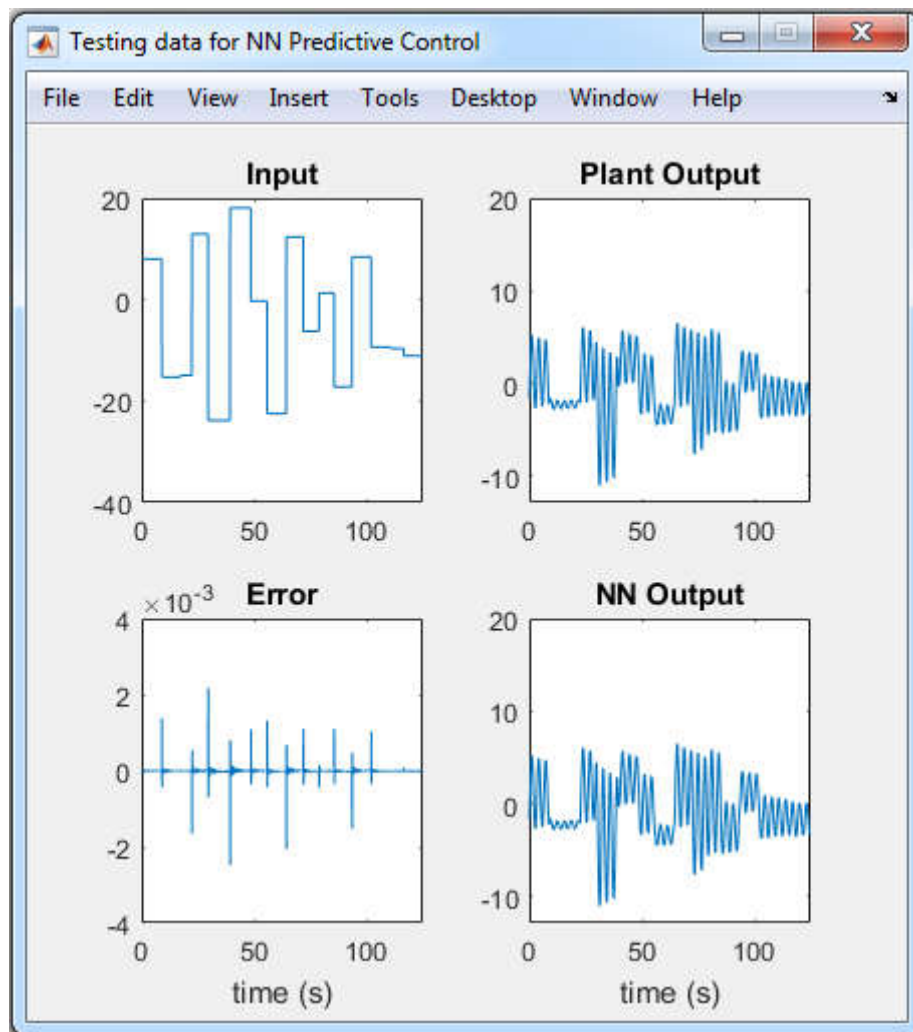


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

Продовження опису вікна Plant Identification.

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

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

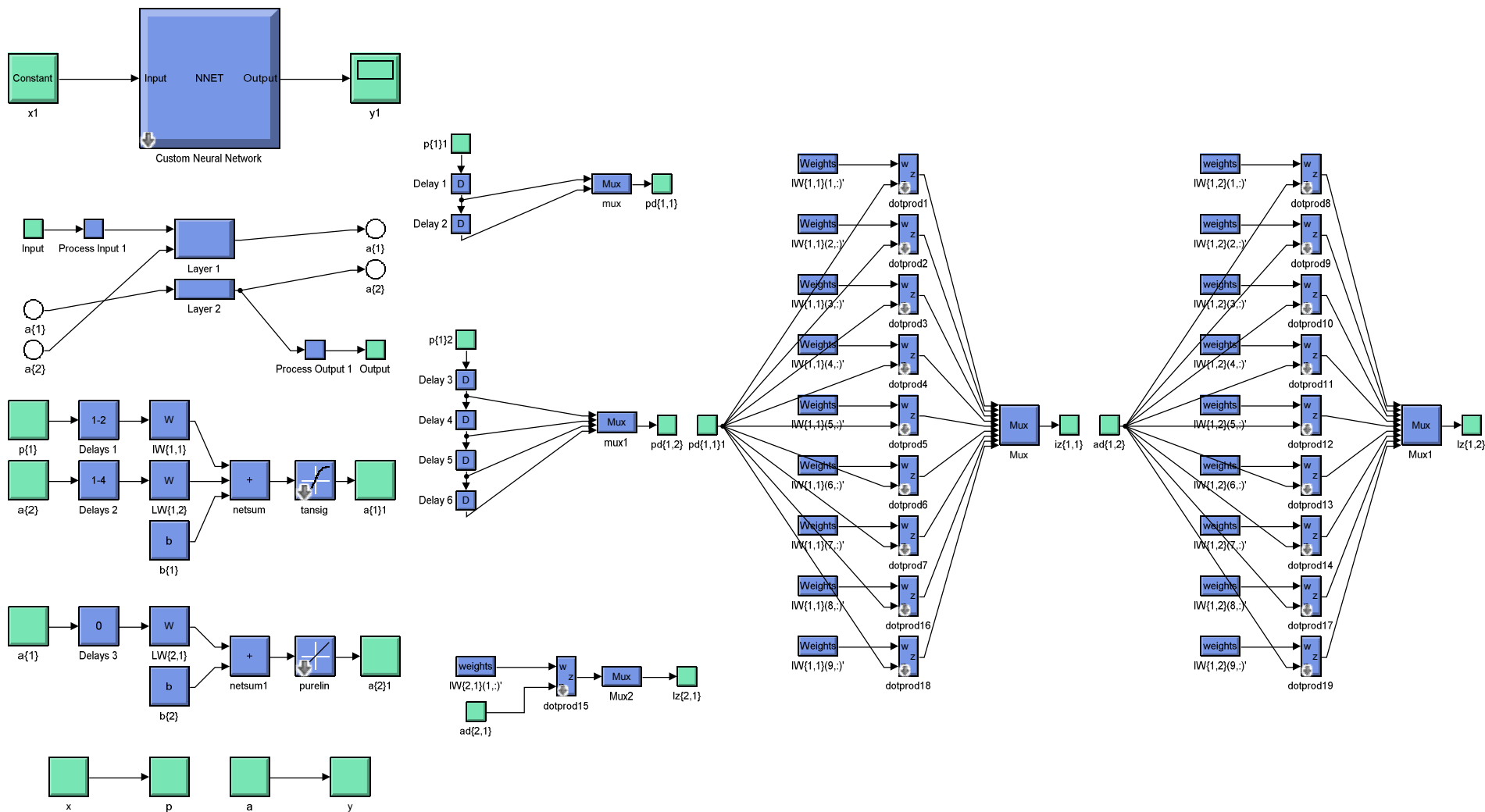


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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

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

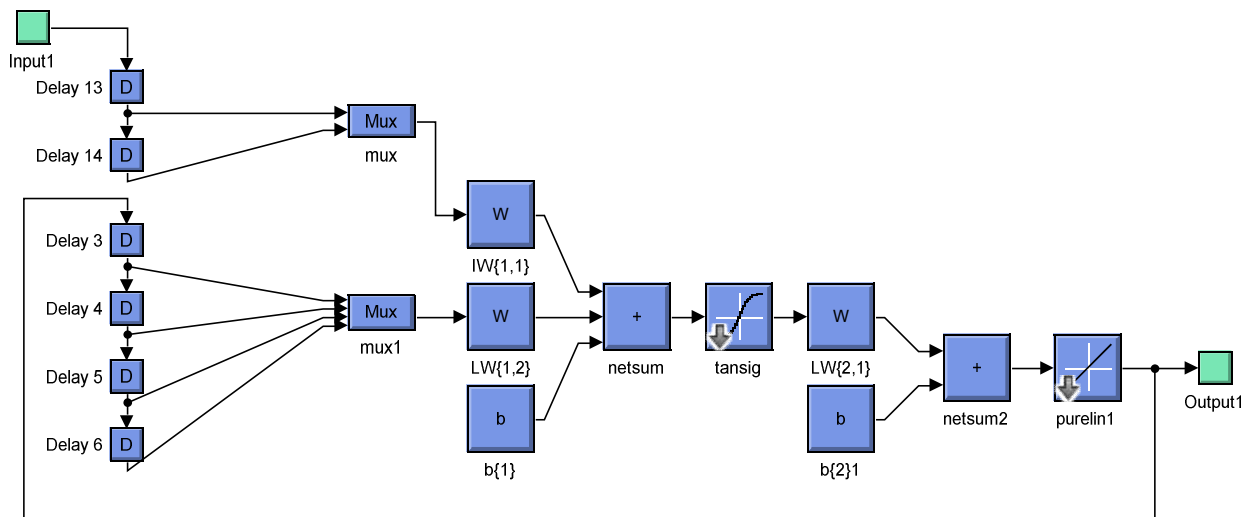


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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

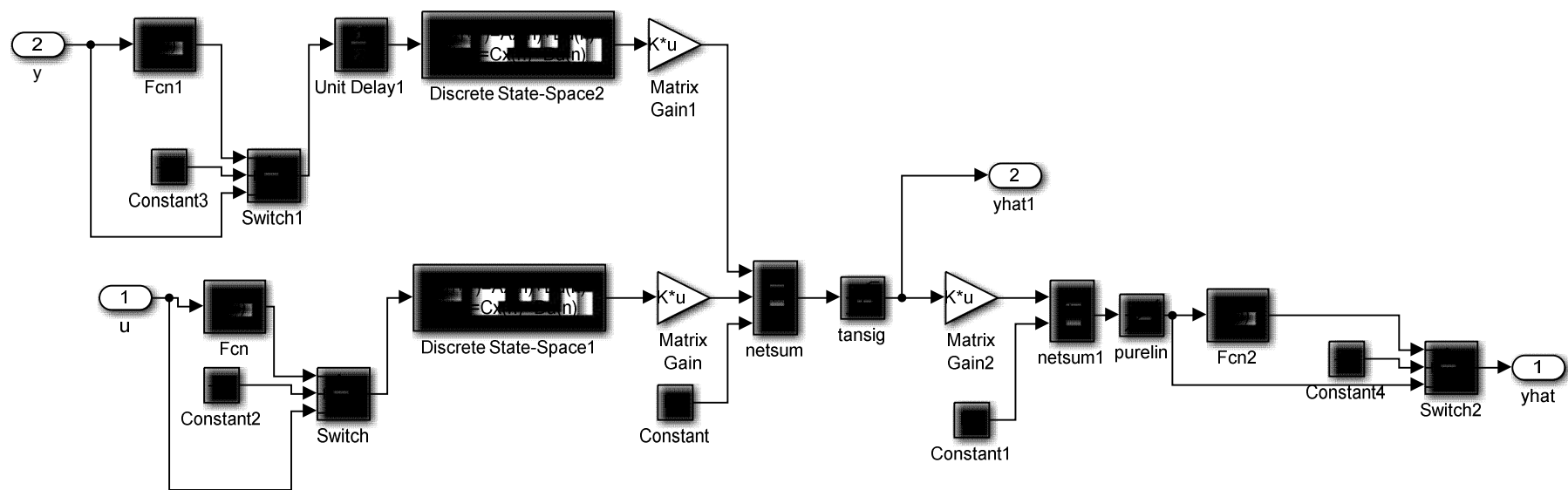


Рисунок 4.23 – Структурна схема нейронмережевої моделі об'єкту регулювання

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

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

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

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

Discrete State Space 1

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

Discrete State Space 2

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

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

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

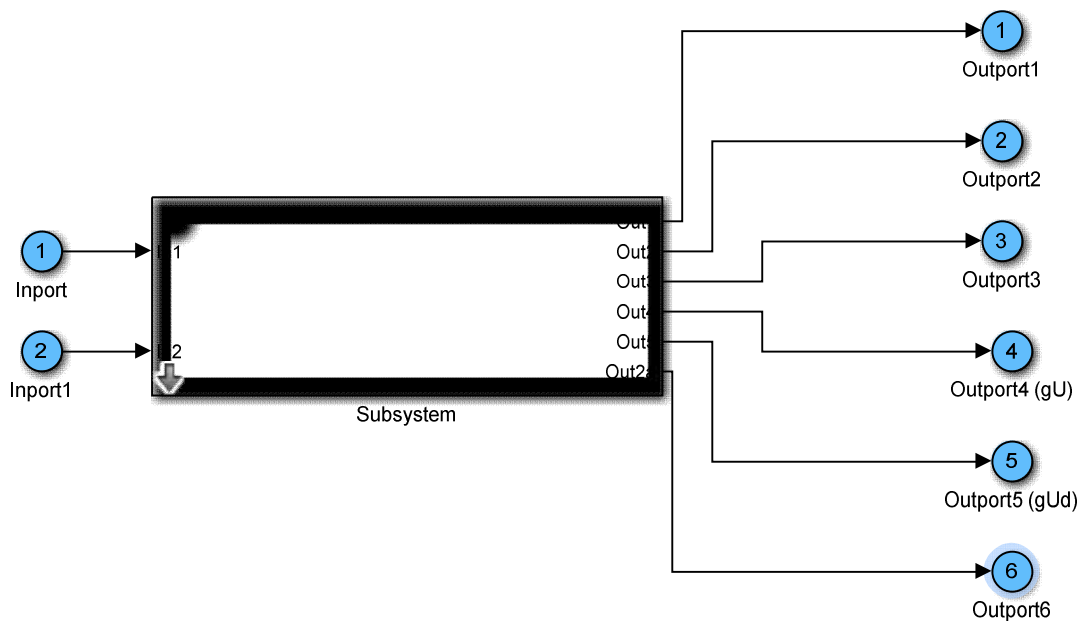


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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

Продовження опису вікна Neural Network Predictive Controller

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

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

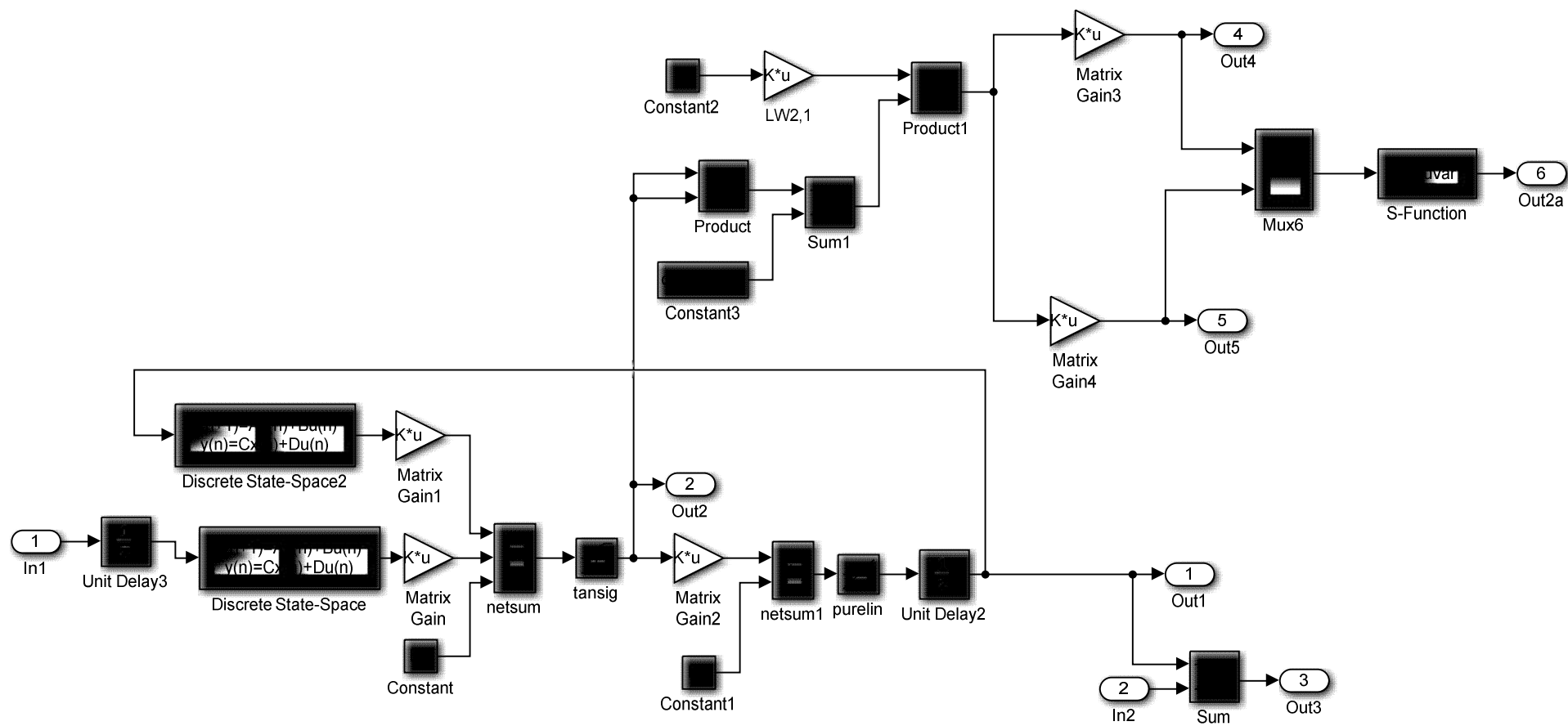


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

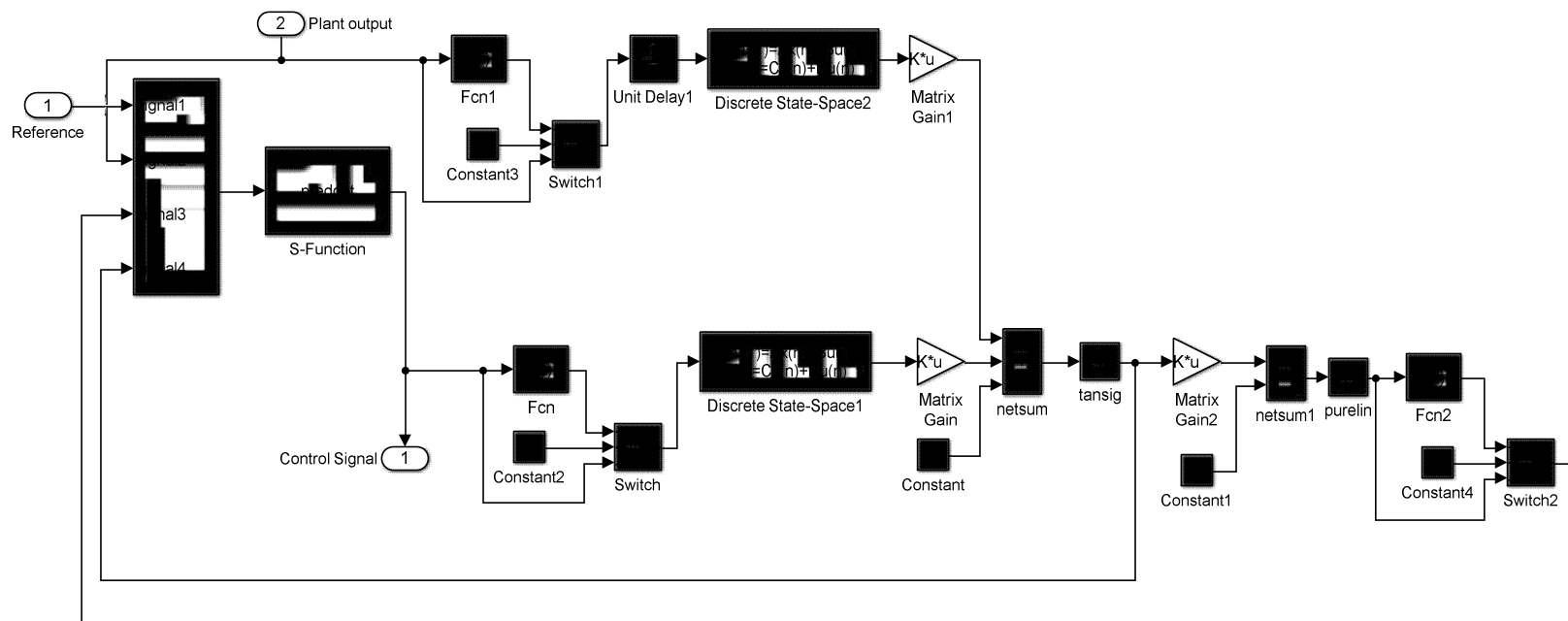


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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

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

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

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

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

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

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

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

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

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

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

Minimization Routine. Вибір процедури одновимірного пошуку.

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

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

впливає на швидкість та точність оптимізації контролера. Зазвичай, більша 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 трьохмасової системи з нейрорегулятором NN Predictive Controller використовуємо моделі Simulink, приведені на рис. 4.5 або 4.27.

На рис. 4.28 – 4.33 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.

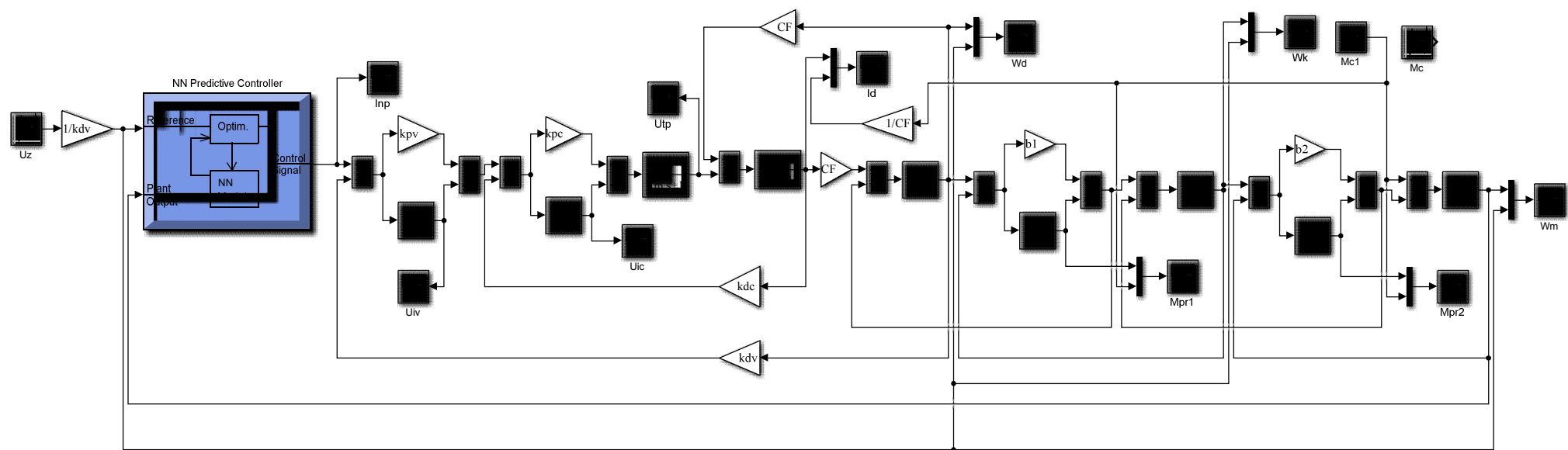


Рисунок 4.27 – Схема моделі трьохмасової системи з нейрорегулятором

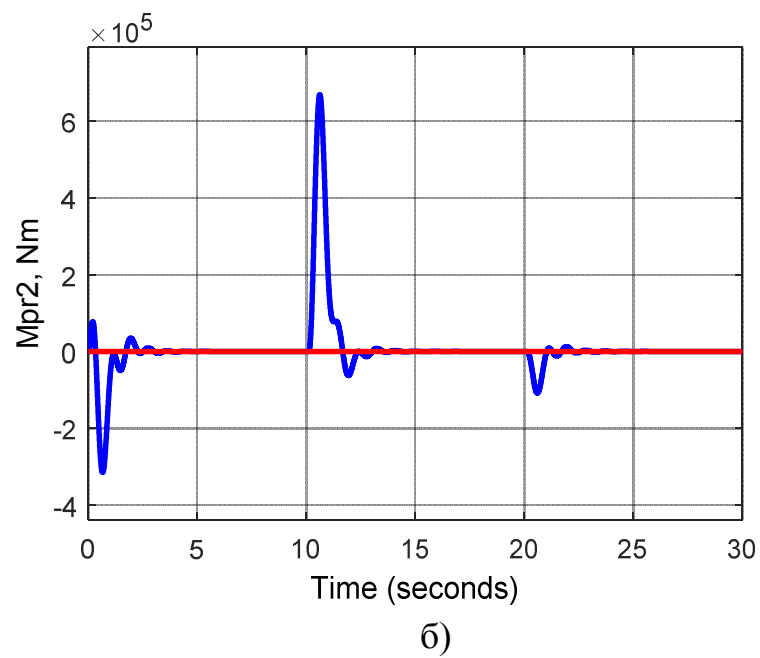
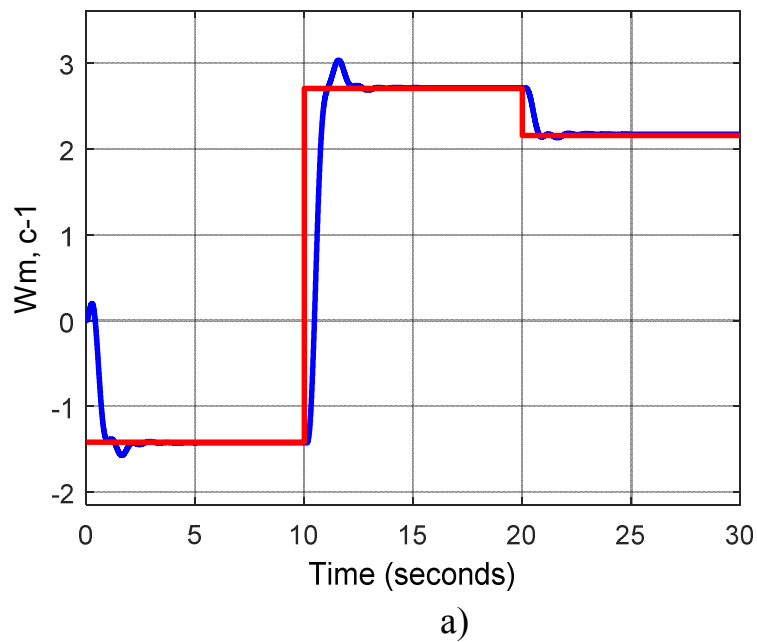
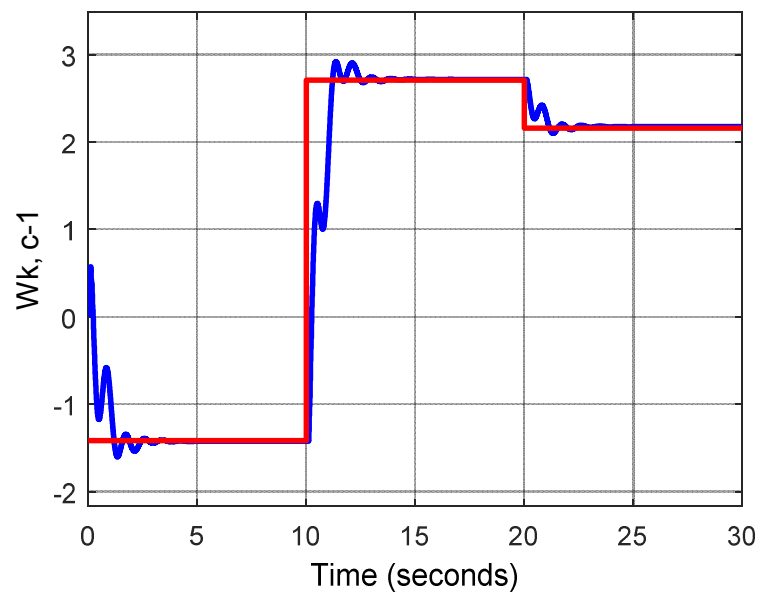
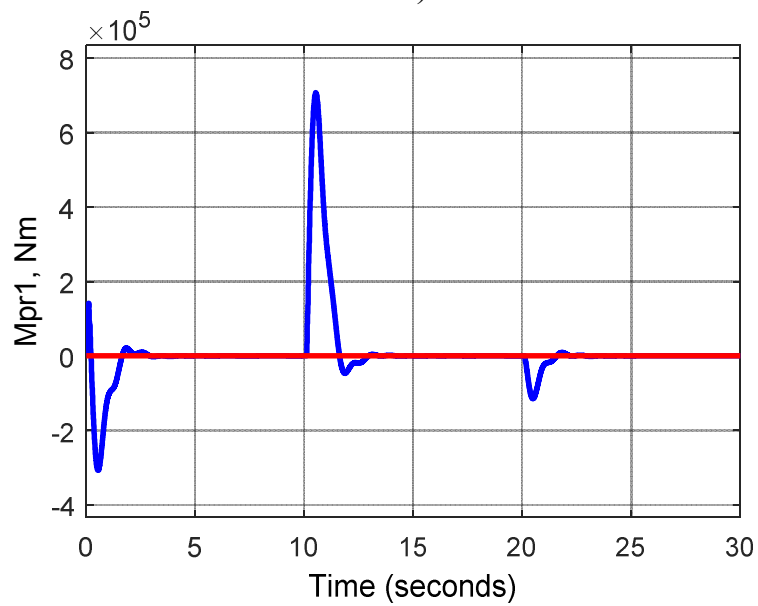


Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр2} = f(t)$ (б) трьохмасової системи з нейрорегулятором по задаючій дії

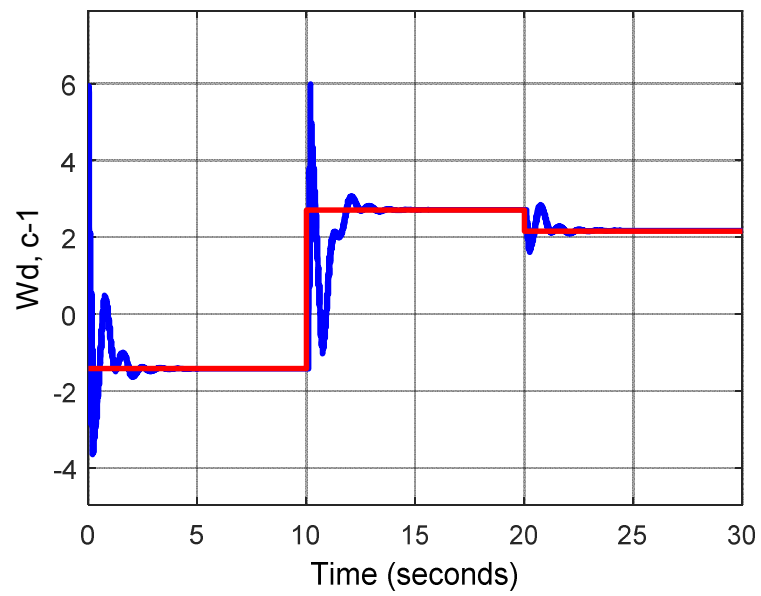


a)

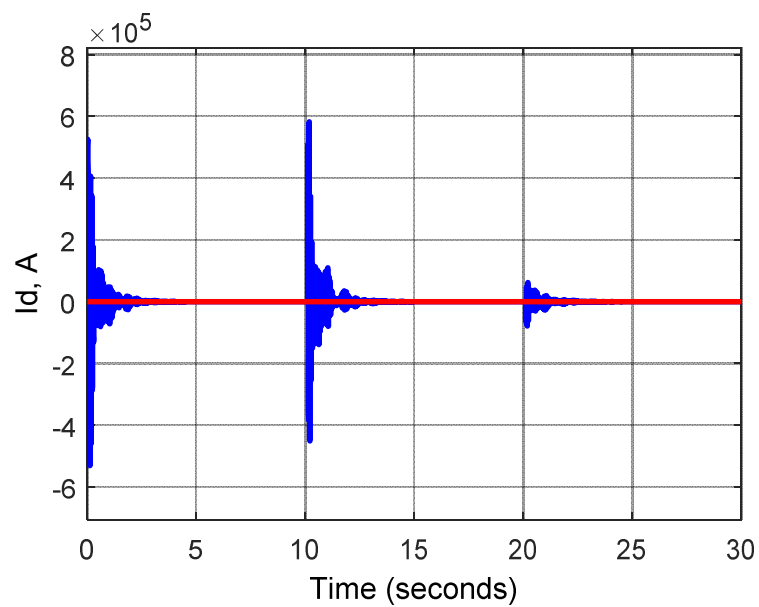


б)

Рисунок 4.29 – Графіки перехідних процесів швидкості каната $\omega_k = f(t)$ (а) і моменту пружності $M_{пр1} = f(t)$ (б) трьохмасової системи з нейрорегулятором по задаючій дії



a)



б)

Рисунок 4.30 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) трьохмасової системи з нейрорегулятором по задаючій дії

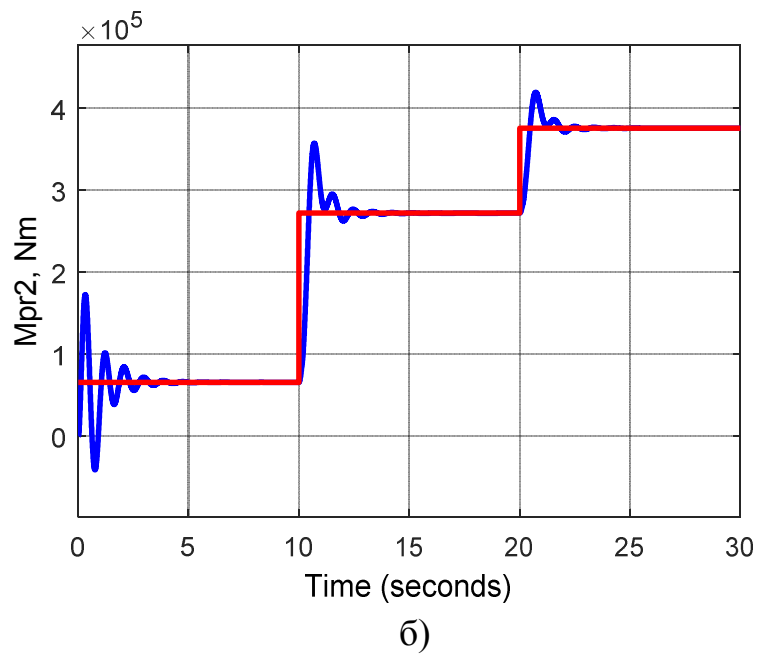
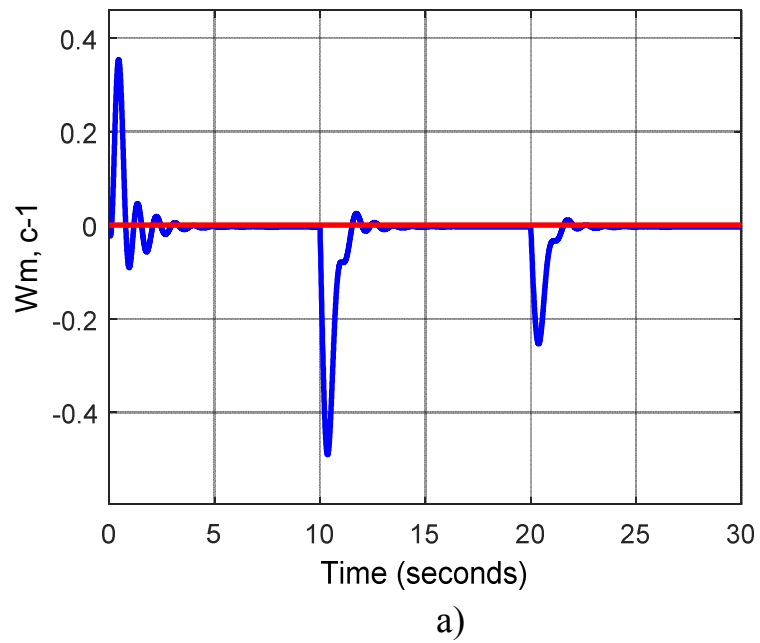
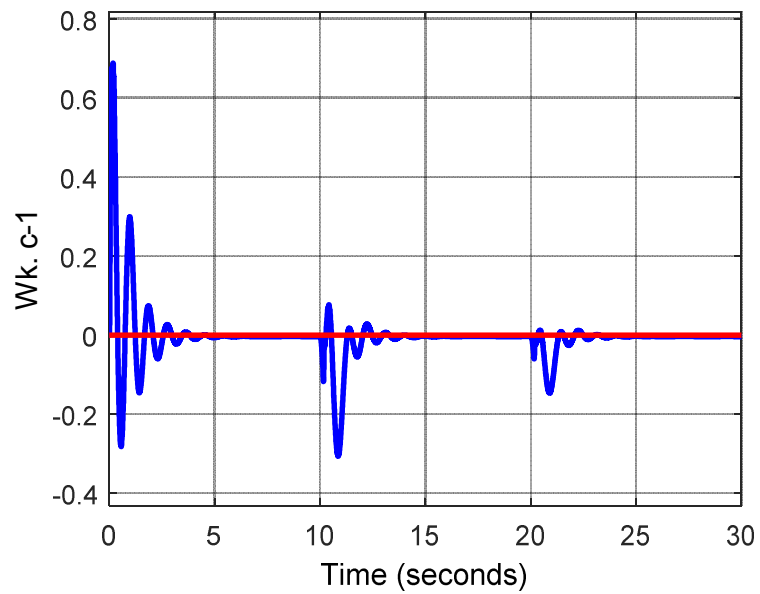
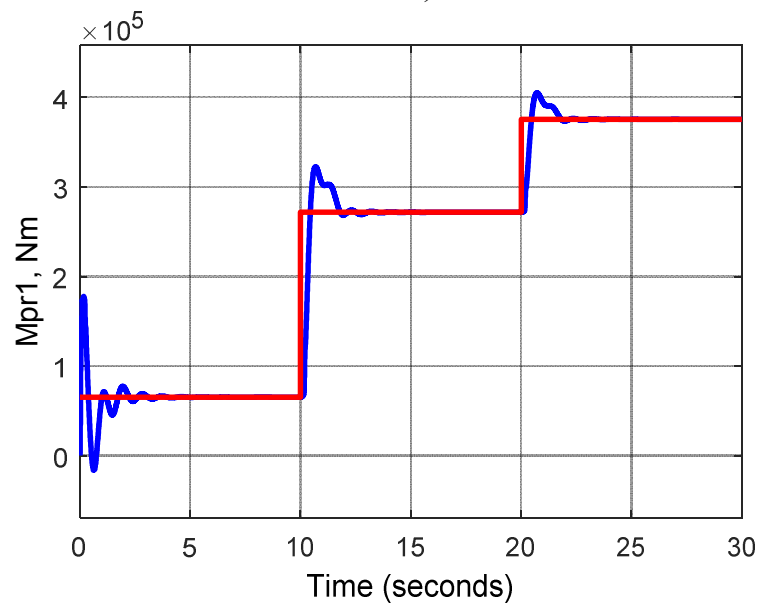


Рисунок 4.31 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр2} = f(t)$ (б) трьохмасової системи з нейрорегулятором по збурюючій дії



a)



б)

Рисунок 4.32 – Графіки перехідних процесів швидкості каната $\omega_k = f(t)$ (а) і моменту пружності $M_{пр1} = f(t)$ (б) трьохмасової системи з нейрорегулятором по збурюючій дії

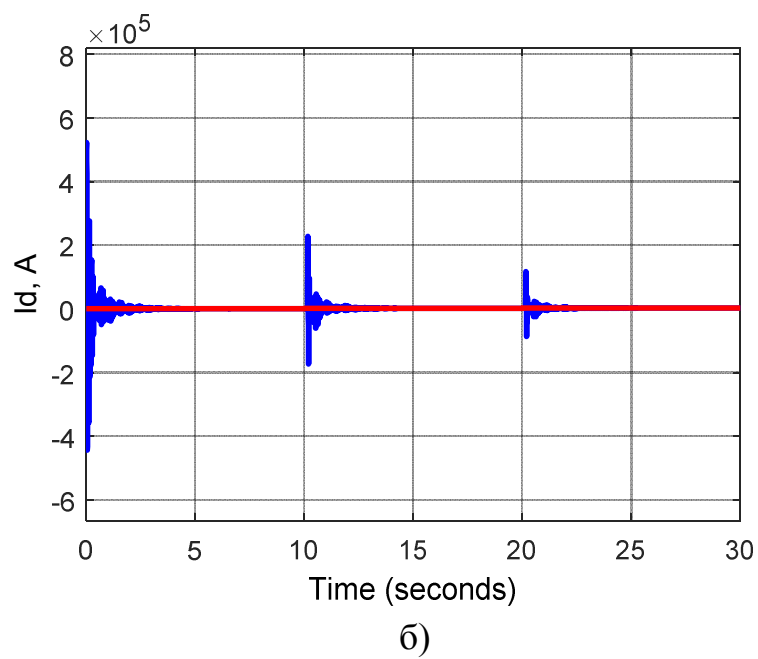
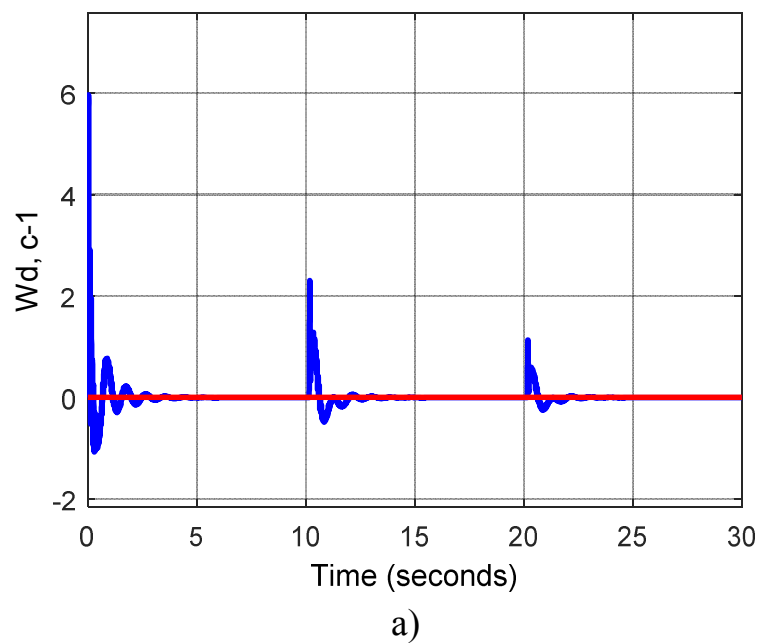


Рисунок 4.33 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) трьохмасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проаналізовано принципи побудови нейрорегулятора, реалізованих в пакеті прикладних програм (ППП) Neural Network Toolbox системи MATLAB: регулятор з прогнозом (NN Predictive Controller); регулятор на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) і регулятор на основі еталонної моделі (Model Reference Controller). Для системи стабілізації, що розглядається в даній роботі, обраний регулятор з прогнозом NN Predictive Controller. Розглянуто структуру та порядок синтезу даних 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 трьохмасової системи управління механізмом підйомі шахтної пвдіймальної 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 10с.

4. Створено модель трьохмасової 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 8%, 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 ТП–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 коректуючих пристроїв контурів, з оптимізацією контура струму по модульному критерію, а контура швидкості – по симетричному критерію. Проведено моделювання системи на ЕОМ з використанням пакету прокладних програм MATLAB.

4. Розроблено математичну модель трьохмасової системи, яка враховує пружні властивості підйомного канату. Моделювання системи виконано на

ЕОМ. Дослідження показали, що в трьохмасовій системі перехідні процеси основних координат супроводжуються значними коливаннями.

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 трьохмасовій 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. McCulloch W. S., Pitts W. A logical calculus of ideas imminent in nervous activity // Bulletin Mathematical Biophysics. – 1943. – № 5. – P.115-133.
14. Hebb D. The Organization of Behavior. – New York: Wiley Publications, 1949.
15. Rosenblatt F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain // Psychological Review. – 1958. – № 65. – P. 386-408.
16. Розенблатт Ф. Принципы нейродинамики. Персептрон и теория механизмов мозга: Пер. с англ. – М.: Мир, 1965
17. Hubel, D. H., & Wiesel, T N. (1959). Brain mechanisms of vision. Scientific American, 241, 150-162.
18. Widrow B., Hoff M. E. Adaptive Switching Circuits / IRE WESCON Convention Record – New York, IRE, 1960. – P. 96-104.
19. Widrow B., Winter R. Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition // IEEE Computer. – 1988. – № 21. – № 3. – P. 25-39.
20. Штейнбух К. Автомат и человек., М: Советское радио, 1967. – 496 с.
21. Минский М. Л., Пейперт С. Персептроны: Пер. с англ. – М.: Мир, 1971.
22. Hopfield J.J. Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons // Proc. of the National Academy of Science. – 1982. – 81. – P. 3088-3092.
23. Hopfield J. J., Tank D. W. Neural Computation of Decisions in Optimization Problems // Biol. Cybernetics. – 1985. – 52. – P. 141-152.
24. Клепиков В.Б., Махотило К.В., Сергеев С.А., Вороновский Г.К. Искусственные нейронные сети: новая парадигма в управлении // Труды межд. конф. «Проблемы автоматизированного электропривода. Теория и практика».– Харьков: Основа.– 1995. – С. 111-115.
25. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности.– Харьков: Основа.– 1997.– 112 с.

26. Sen P., Hearn G.E., Zhang Y. Adaptive Neural Controller // In Neural Network Systems Techniques and Applications. Hrsg. Leondes C.T. – 1998. – Vol.4. – P. 274-343.
27. Werbos P.J. Backpropagation and neurocontrol: A review and prospects // Proc. of International Joint Conf. On Neural Networks. – Washington DC. – 1989. – Vol. 1. – P. 209-216.
28. Psaltis A.; Sideris A., Yamamura A. A multi-layered neural network controller // IEEE Control Systems Mag. – 1988. – №8. – P. 17-21.
29. Saerens M.; Soquet A. A neural controller // Proceedings of the 1st IEE International Conference on Artificial Neural Networks, London. – 1989. – P. 211-215.
30. Narendra K.S. Neural networks for control: theory and practice // Proc. IEEE. – 1996. – №84. – P. 1385-1406.
31. Neural networks for control systems: A survey K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop // Automatica. – 1992. – Vol. 28. – № 6. P. 1083-1112.
32. Ishiguro A. et al. A neural network compensator for uncertainties of robotic manipulators // IEEE Trans, on Industrial Electronics. – 1992. – 339. – P. 565-569.
33. Garcia C.E.; Morari M. Internal Model Control: Unifying Review and Some New Results // Ind.Eng.Chem.Process Des.Dev. – 1982. – №21. – 308-323.
34. Park Y.-M., Choi M.S.; Lee Y. An optimal tracking neuro-controller for non-linear dynamic systems // IEEE Trans. Neural Networks. – 1996. – №7. – P. 1099-1110.
35. Mistry, S.I.; Nair S.S. Indirect control of a class of nonlinear dynamic systems. // IEEE Trans. Neural Networks. – 1996. – №7. – P. 1015-1023.
36. Miyamoto H., Kawato M., Setoyama, T., Suzuki R. Feedback error learning neural network for trajectory control of robotic manipulator // IEEE Trans, on Neural Networks. – 1988. – Vol. 1. – P. 251-265.
37. Jung S. Neural Network Controllers for Robot Manipulators. PhD thesis, University of California. – Davis. – 1996. – 132 p.
38. Jung S.; Hsia T.C. A new neural network control technique for robot manipulators // Robotica. – 1995. – Vol. 13. – P. 477-484.

39. Sorensen O. Neural Networks in Control Applications. PhD thesis. Aalborg University. Department of Control Engineering. – 1994. – 187 p.
40. Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // Automatica. – 1989. – Vol.25. – P.335-348.
41. Solowey D. Neural Generalized Predictive Control / Solowey D., Haley P.J //Proceedingof the 1996 IEEE Symposium on Intelligent Control. – 1996. – P. 277–282.
42. Clarke D.W., Mohtadi C., Tuffs C. Generalized Predictive Control - Part 1: The Basic Algorithm // Automática. – 1987. – Vol. 23. – P. 137-148.
43. Clarke D. W., Mohtadi C., Tuffs C. Generalized Predictive Control. Part 2: The Basic Algorithm // Automática. – 1987. – Vol. 23. – P. 149-163.
44. Clarke D. W. Advances in model-based predictive control.- In Advances in Model-Based Predictive Control / D. W. Clarke: Oxford University Press. – 1994. – 274
45. Montague G. A., Willis M. J., Tham M. T., Morris A. J. Artificial Neural Network Based Control // Int. Conference on Control. – 1991. – Vol.1. – P. 266-271.
46. Takahashi Y. Adaptive Predictive Control of Nonlinear Time-Varying System using Neural Network //IEEE International Conference on Neural Networks, 1993. – Vol.3. – P.1464-1468
47. Jeong Jun Song, Sunwon Park. Neural Model-Predictive Control for Nonlinear Chemical Processes // Journal of Chemical Engineering of Japan. – 1993. – Vol.26. – №4. –P.347-354.
48. Psychogios D. C., Ungar L. H. Nonlinear Internal Model Control and Model Predictive Control using Neural Networks // 5th IEEE Int. Symposium on Intelligent Control. – 1990. – P.1082-1087.
49. Press W. H., Flannery B. P., Teukolsky S. A., Vetterling V. T. Numerical Recipes.- The Art of Scientific Computing. Cambridge University Press, 1988. – 452.
50. Rumelhart D. E., McClelland J. L. and the PDP Research Group. Parallel Distributed Processing. The MIT Press. – 1986. – Vol.1. – Chapter 8. – 337 p.

51. Электрооборудование шахтных подъёмных машин /Ю.Т.Калашников, В.Е.Католиков, Г.И.Шпильберг и др., – М.: Недра, 1986. – 285 с.
52. Католиков В.Е., Динкель А.Д., Седуни А.М. Тиристорный электропривод с реверсом возбуждения двигателя рудничного подъёма. – М.: Недра, 1990.–382 с
53. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
54. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
55. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
56. 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.
57. 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.
58. 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.
59. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.
60. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.
61. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
62. Автоматизований електропривод ч. 2 [Електронний ресурс]: навчальний посібник для студентів освітньої програми «Електромеханічні системи автома-

тизації, електропривод та електромобільність» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / В.І. Теряєв. – Київ: КПІ ім. Ігоря Сікорського, 2022.– 204 с

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

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

65. 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.

66.Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоєнко. Харків, ХНАДУ, 2012. – 284 с.

67. Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтьєв, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.

68. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.

69. 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.

70. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 с.

ДОДАТОК А

МОДЕЛЬ НЕЙРОНА

А.1 Структурна схема нейрона

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

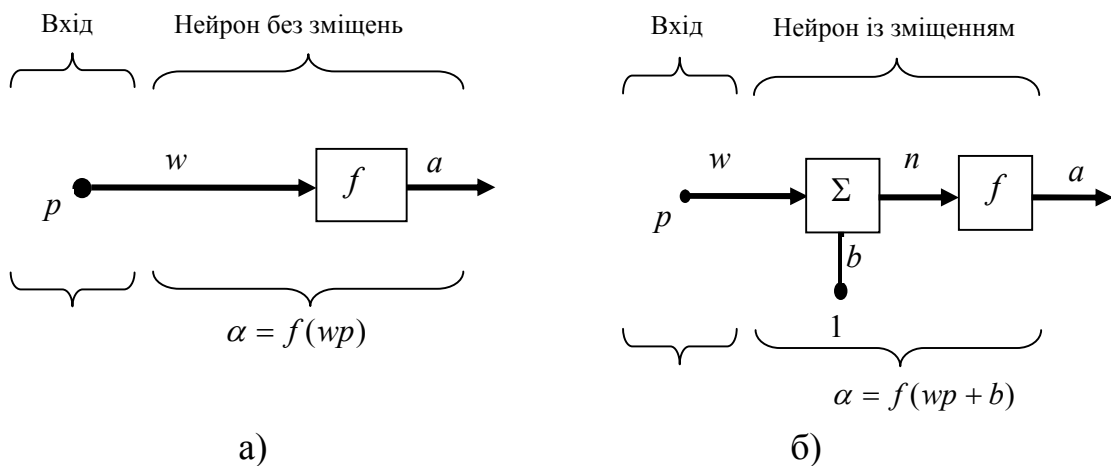


Рисунок А.1 – Структура простого нейрона

а) нейрон з одним скалярним входом;

б) нейрон з скалярним зсувом

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

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

трів нейрона так, щоб поведінка мережі відповідала деякій бажаній поведінці. Регулюючи ваги або параметри зсуву, можна навчити мережу виконувати конкретну роботу; можливо також, що мережа сама коректуватиме свої параметри, щоб досягти необхідного результату.

Рівняння нейрона із зсувом має вигляд

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

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

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

A.2 Функції активації нейронів

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

Розглянемо три найбільш поширені форми функції активації.

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

До складу пакету прикладних програм (ППП) `Neural Network Toolbox` системи `MATLAB` входить М-функція `hardlim`, що реалізовує функцію активації з жорсткими обмеженнями.

Лінійна функція активації `purelin`. Ця функція описується співвідношення $a = \text{purelin}(n) = n$ і показана на рис. A1.3.

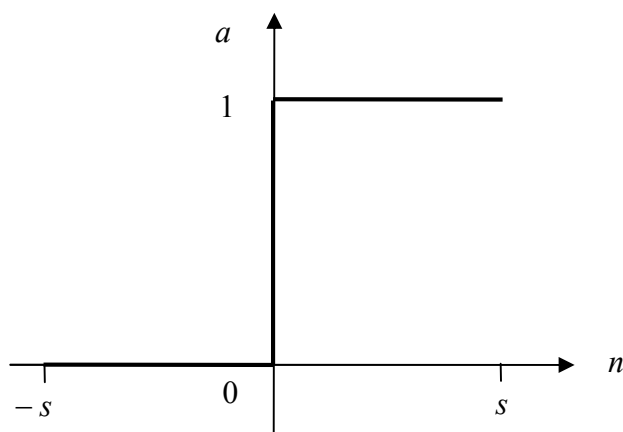


Рисунок А.2 – Одинична функція активації

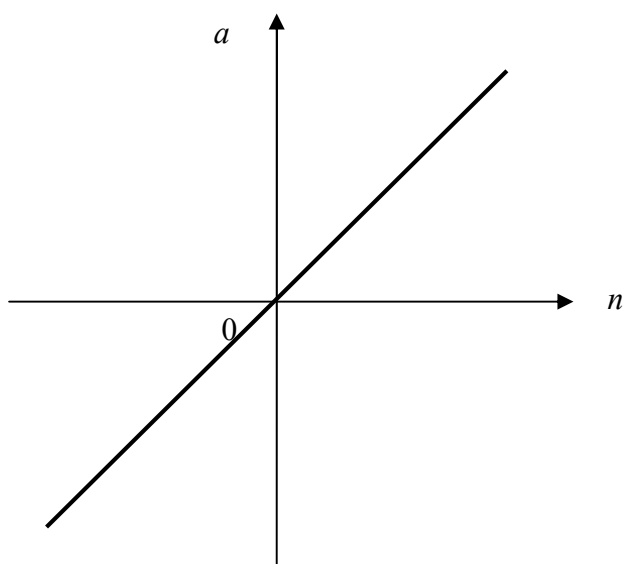


Рисунок А.3 – Лінійна функція активації

Логістична функція активації logsig . Ця функція описується співвідношення $a = \text{logsig}(n) = 1/(1 + \exp(-n))$ і показана на рис. А.4. Вона належить до класу сигмоїдальних функцій, і її аргумент може приймати будь-яке значення в діапазоні від $-\infty$ до $+\infty$ а вихід змінюється в діапазоні від 0 до 1. У ППП Neural Network Toolbox вона представлена М-функцією logsig . Завдяки властивості диференцируемости ця функція часто використовується в мережах з навчанням на основі методу зворотного розповсюдження помилки.

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

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

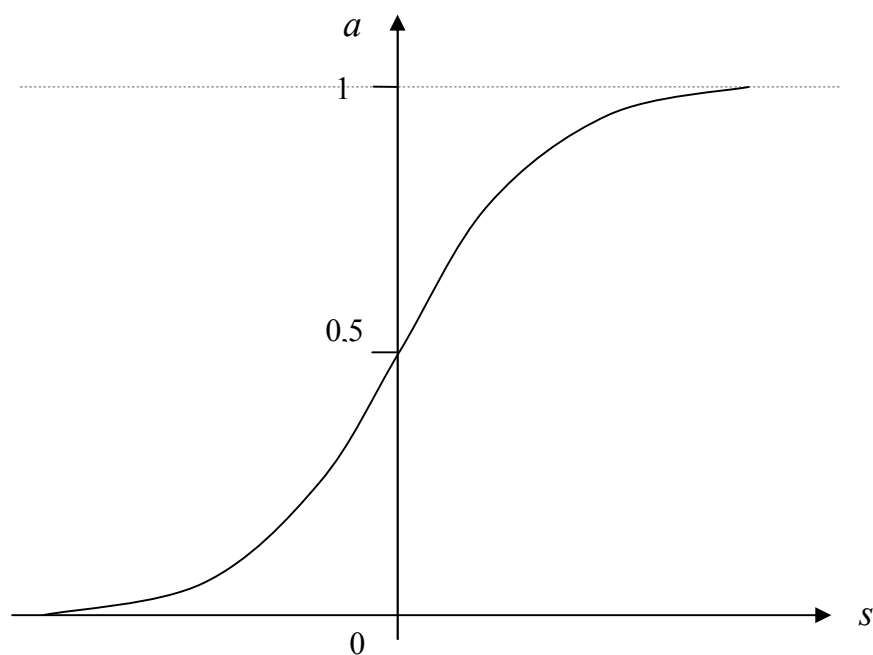


Рисунок А.4 – Логістична функція активації

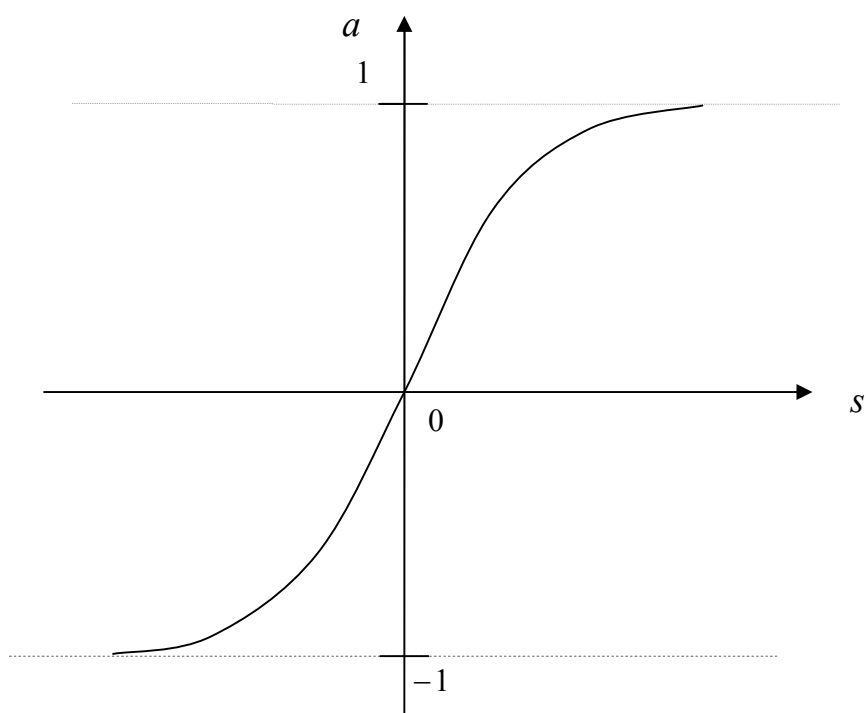


Рисунок А.5 – Функція активації типу гіперболічного тангенса

А.3 Нейрон з векторним входом

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

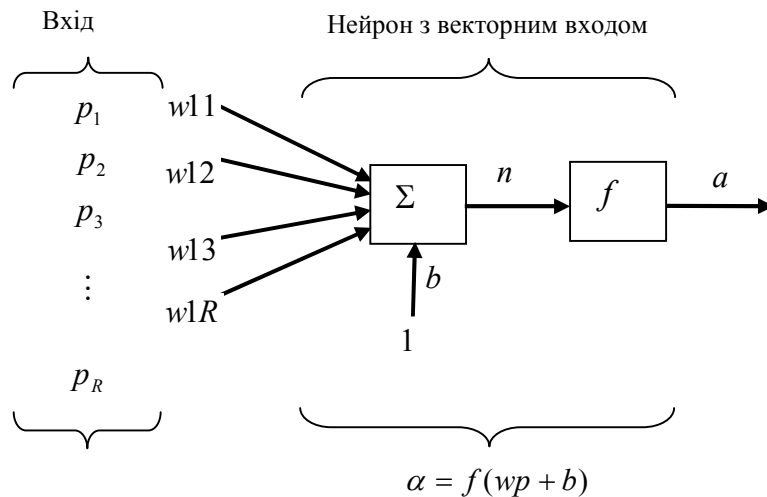


Рисунок А.6 – Схема нейрона з векторним входом

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

$$n = w_{11} \cdot p_1 + w_{12} \cdot p_2 + \dots + w_{1R} \cdot p_R + b \quad (1.3)$$

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

$$n = \mathbf{W} \cdot \mathbf{p} + b \quad (1.4)$$

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

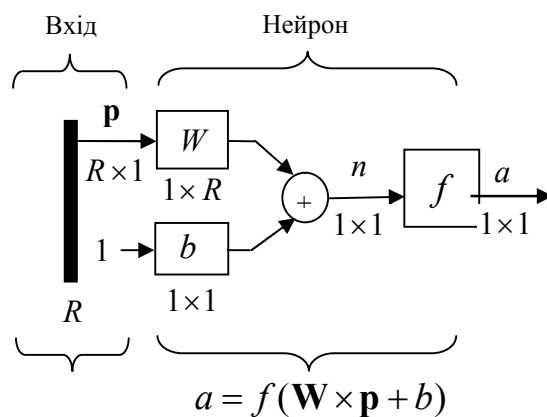


Рисунок А.7 – Укрупнена структурна схема нейрона

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

ДОДАТОК Б

АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ

Б.1 Одношарові мережі

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

Розгорнута схема мережі з одного шару з R вхідними елементами і S нейронами показана на рис. Б.1

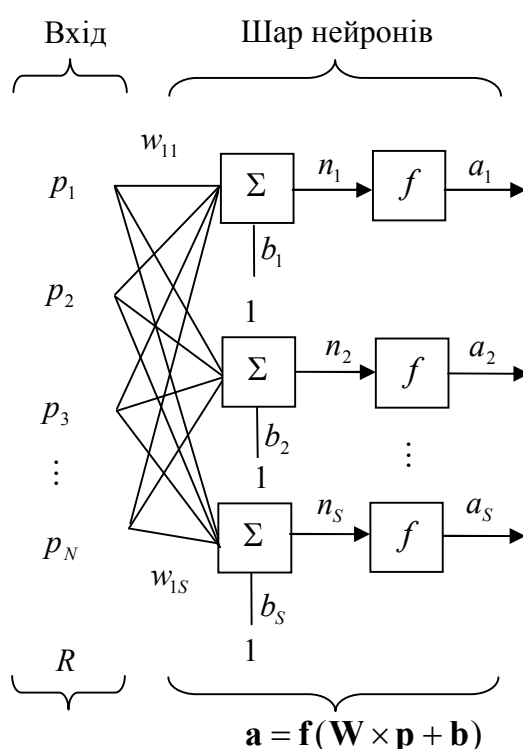


Рисунок Б.1 – Схема одношарової мережі

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

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

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

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

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

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

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

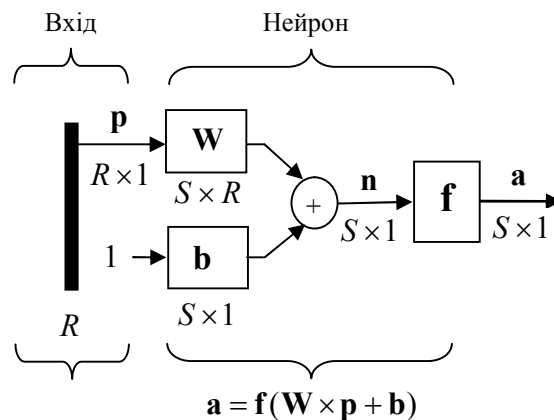


Рисунок Б.2 – Укрупнена структурна схема одношарової мережі

Б.2. Багатошарові мережі

Розглянемо мережі, що мають декілька шарів. Вагові матриці, сполучені з входами, називаються вагами вхідного шару, а вагові матриці для сигналів, витікаючі з шару, називаються вагами вихідного шару. Далі, будемо використовувати верхні індекси, щоб вказати джерело і адресат для різних вагів і інших елементів нейронної мережі. Щоб пояснити це, розглянемо спочатку тільки один, перший шар багатошарової мережі (рис. Б.3)

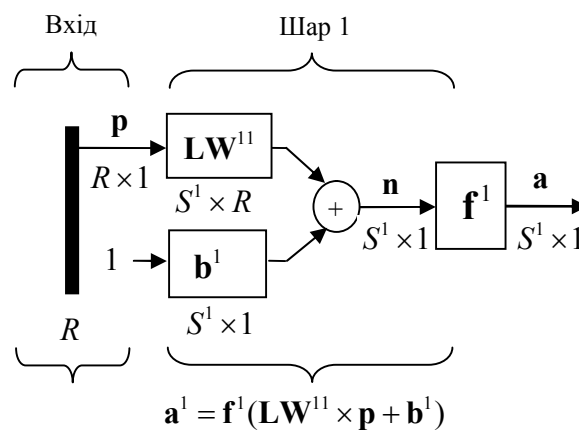


Рисунок Б.3 – Структурна схема першого шару багатошарової мережі

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

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

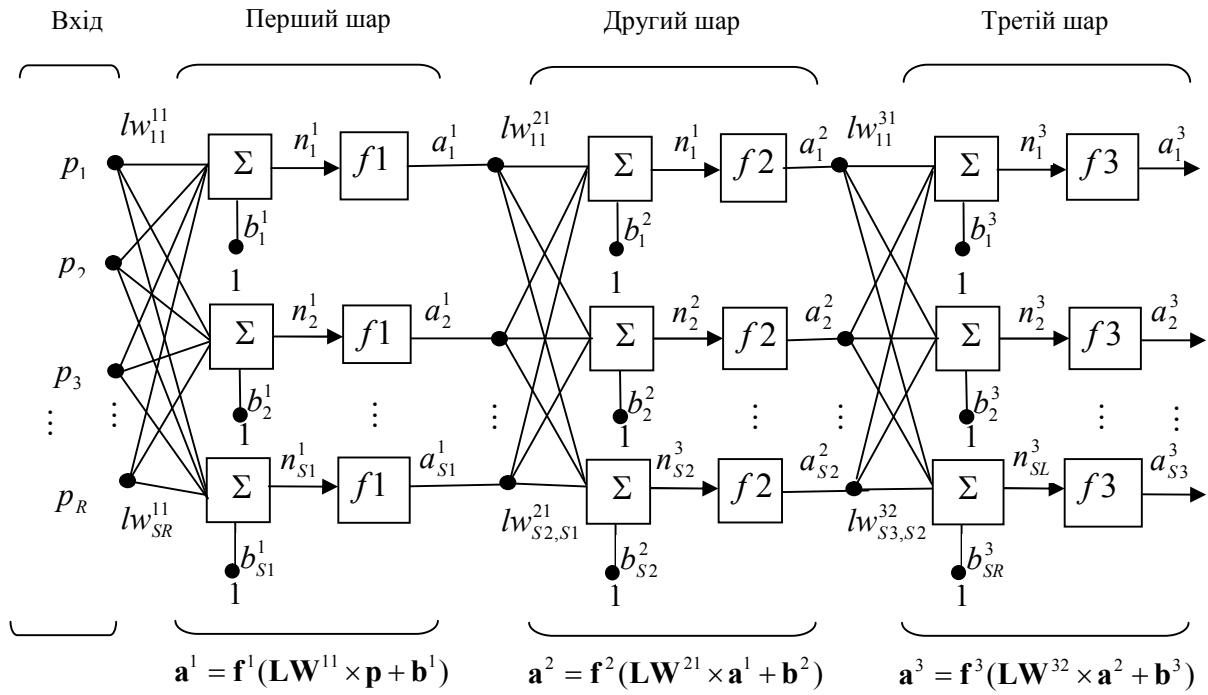


Рисунок Б.4 – Структурна схема багатошарової мережі

Мережа, показана вище, має R входів, S^1 нейронів в першому шарі, S^2 нейронів в другому шарі і т.д. Для спільності вважатимемо, що різні шари мають різне число нейронів. На зсуви для кожного нейрона поданий постійний вхідний сигнал 1. Відмітимо, що виходи кожного проміжного шару служать входами для наступного шару. Таким чином, шар 2 може бути розглянутий як один шар мережі з S^1 входами S^2 нейронами і $S^1 \times S^2$ матрицею вагів $\mathbf{W}_2$. Вхід до шару 2 є 1, а вихід - 2. Тепер, коли позначені всі вектори і матриці шару 2, можна трактувати його як самостійну одношарову мережу. Такий підхід може бути використаний до будь-якого шару мережі.

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

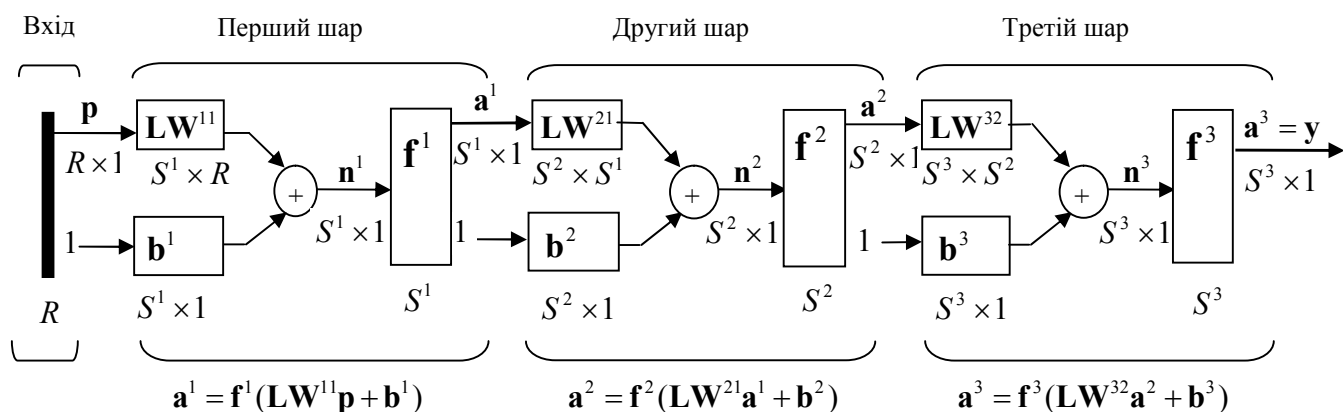


Рисунок Б.5 – Укрупнена структурна схема багатошарової мережі

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

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

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

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

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

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

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

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

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

Б.3 Мережі з прямою передачею сигналу

Одношарова мережа з S нейронами з функціями активації logsig , що має R входів, показана на рис. Б.6.

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

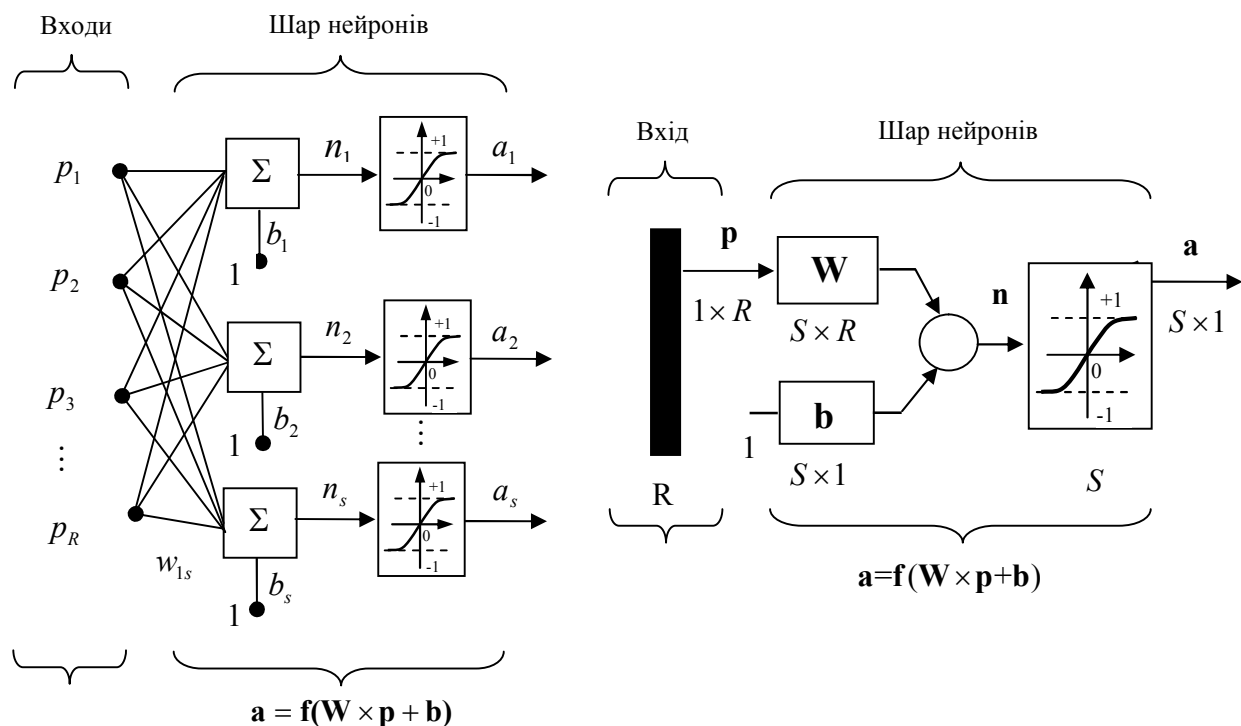


Рисунок Б.6 – Структурна схема одношарової мережі з R входами

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

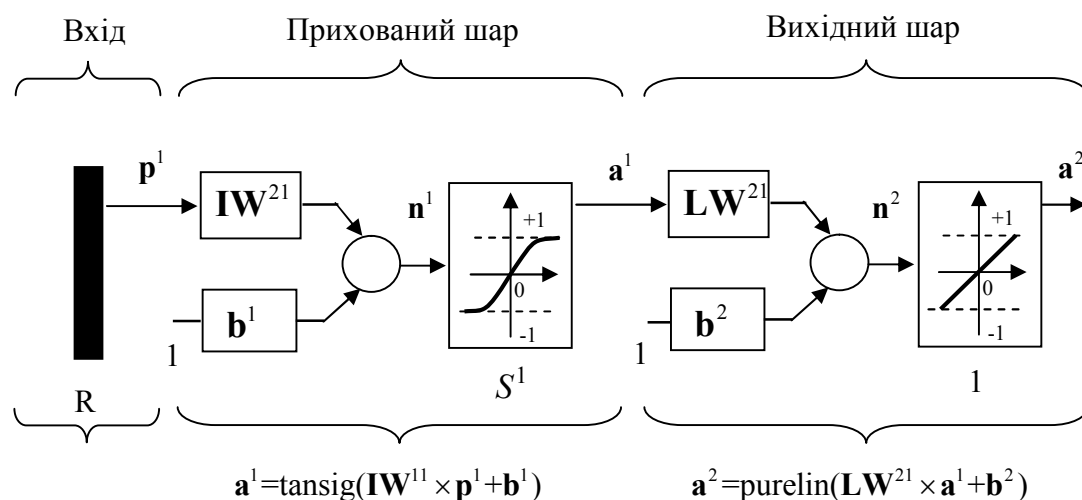


Рисунок Б.7 – Структурна схема багатошарової мережі з R входам

ДОДАТОК В

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

В.1 Завдання навчання

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

Вибір ознак, принаймні первинний, здійснюється евристично на основі наявного досвіду, який може підказати, які ознаки є найбільш важливими. Спочатку слід включити всі ознаки, які, на думку аналітиків або експертів, є істотними, на подальших етапах ця множина буде скорочена.

Нейронні мережі працюють з числовими даними, узятими, як правило, з деякого обмеженого діапазону. Це може створити проблеми, якщо значення спостережень виходять за межі цього діапазону або пропущені.

Питання про те, скільки потрібно мати спостережень для навчання мережі, часто виявляється непростим. Відомий ряд евристичних правил, які встановлюють зв'язок між кількістю необхідних спостережень і розмірами мережі. Просте з них свідчить, що кількість спостережень повинна бути в 10 разів більше числа зв'язків в мережі. Насправді це число залежить від складності того відображення, яке повинна відтворювати нейронна мережа. Із зростанням числа використовуваних ознак кількість спостережень зростає по нелінійному закону, так що вже при досить невеликому числі ознак, скажемо 50, може потрібно величезне число спостережень. Ця проблема носить назву "Прокляття розмірності".

Для більшості реальних завдань буває достатнім декількох сотень або тисяч спостережень. Для складних завдань може потрібно більша кількість, проте дуже рідко зустрічаються завдання, де потрібно менше 100 спостережень. Якщо даних мало, то мережа не має достатньої інформації для навчання, і краще, що

можна в цьому випадку зробити, - це спробувати підігнати до даним деяку лінійну модель.

В.2 Процедури адаптації і навчання

Після того, як визначена кількість шарів мережі і число нейронів в кожному з них, потрібно призначити значення вагів і зсувів, які мінімізують помилку рішення. Це досягається за допомогою процедур навчання. Шляхом аналізу тих, що є у розпорядженні аналітика вхідних і вихідних даних ваги і зсуву мережі автоматично настроюються так, щоб мінімізувати різницю між бажаним сигналом і отриманим на виході в результаті моделювання. Ця різниця носить назву помилки навчання. Таким чином, процес навчання - це процес підгонки параметрів тієї моделі процесу або явища, яка реалізується нейронною мережею. Помилка навчання для конкретної конфігурації нейронної мережі визначається шляхом прогону через мережу всіх наявних спостережень і порівняння вихідних значень з бажаними, цільовими значеннями. Ці різниці дозволяють сформувати так звану функцію помилок (критерій якості навчання). Як така функція найчастіше береться сума квадратів помилок. При моделюванні нейронних мереж з лінійними функціями активації нейронів можна побудувати алгоритм, що гарантує досягнення абсолютного мінімуму помилки навчання. Для нейронних мереж з нелінійними функціями активації в загальному випадку не можна гарантувати досягнення глобального мінімуму функції помилки.

При такому підході до процедури навчання може виявитися корисним геометричний аналіз поверхні функції помилок. Визначимо ваги і зсуву як вільні параметри моделі і їх загальне число позначимо через N ; кожному набору таких параметрів поставимо у відповідність одне вимірювання у вигляді помилки мережі. Тоді для всіляких поєднань вагів і зсувів відповідну помилку мережі можна зобразити крапкою в $N+1$ - мірному просторі, а всі такі крапки утворюють деяку поверхню, звану поверхнею функції помилок. При такому підході мета

навчання нейронної мережі полягає в тому, щоб знайти на цій багатовимірній поверхні глобальний мінімум.

У разі лінійної моделі мережі і функції помилок у вигляді суми квадратів така поверхня буде параболоїдом, який має єдиний мінімум і це дозволяє відшукати такий мінімум досить просто.

У разі нелінійної моделі поверхню помилок має набагато складніша будова і володіє поряд несприятливих властивостей, зокрема може мати локальні мінімуми, плоскі ділянки, сідлові крапки і довгі вузькі яри.

Визначити глобальний мінімум багатовимірної функції аналітично неможливо, і тому навчання нейронної мережі, по суті справи, є процедурою вивчення поверхні функції помилок. Відштовхуючись від випадково вибраної крапки на поверхні функції помилок, алгоритм навчання поступово відшукує глобальний мінімум. Як правило, для цього обчислюється градієнт (нахил) функції помилок в даній крапці, а потім ця інформація використовується для просування вниз по схилу. Врешті-решт алгоритм зупиняється в деякому мінімумі, який може опинитися лише локальним мінімумом, а якщо повезе, то і глобальним.

Таким чином, по суті алгоритми навчання нейронних мереж аналогічні алгоритмам пошуку глобального екстремуму функції багатьох змінних. Серед останніх слід виділити алгоритми зв'язаних градієнтів і Левенберга - Марквардта (Levenberg - Marquardt).

Проте з урахуванням специфіки нейронних мереж для них розроблені спеціальні алгоритми навчання, серед яких слід виділити алгоритм зворотного розповсюдження помилки.

При використанні алгоритму зворотного розповсюдження помилки мережа розраховує помилку, що виникає у вихідному шарі, і обчислює вектор градієнта як функцію вагів і зсувів. Цей вектор указує напрям найкоротшого спуску по поверхні для даної крапки, тому якщо просунутися в цьому напрямі, то помилка зменшиться. Послідовність таких кроків врешті-решт приведе до міні-

муму того або іншого типу. Певну трудність тут викликає вибір величини кроку.

При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути через рішення або піти в неправильному напрямі. Класичним прикладом такого явища при навчанні нейронної мережі є ситуація, коли алгоритм дуже поволі просувається по вузькому яру з крутими схилами, перестрибуючи з одного схилу на іншій. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, проте при цьому буде потрібно дуже багато ітерацій. На практиці величина кроку вибирається пропорційній крутизні схилу (градієнту функції помилок); такий коефіцієнт пропорційності називається параметром швидкості настройки. Правильний вибір параметра швидкості настройки залежить від конкретного завдання і зазвичай здійснюється досвідченим шляхом; цей параметр може також залежати від часу, зменшуючись у міру виконання алгоритму.

Алгоритм діє ітеративно, і його кроки прийнято називати епохами або циклами. На кожному циклі на вхід мережі послідовно подаються всі повчальні спостереження, вихідні значення порівнюються з цільовими значеннями і обчислюється функція помилки. Значення функції помилки, а також її градієнта використовуються для коректування вагів і зсувів, після чого всі дії повторюються. Початкові значення вагів і зсувів мережі вибираються випадковим чином, і процес навчання припиняється або коли реалізована певна кількість циклів, або коли помилка досягне деякого малого значення або перестане зменшуватися.

В.3 Явище перенавчання

Одна з найбільш серйозних труднощів при навчанні мережі полягає в тому, що у ряді випадків ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати; потрібно мінімізувати помилку, яка з'являється в мережі, коли на неї подаються абсолютно нові спостереження. Вельми важливо, щоб нейронна мережа володіла здатністю пристосовуватися до цих нових спостережень. Що ж відбувається насправді? Мережа навчається мінімізувати помилку на деякій

обмеженій повчальній множині. Це не відповідає вимогам теорії про наявність ідеальної і нескінченно великої повчальної множини. І це не відповідає тій реальній ситуації, коли треба мінімізувати конкретну функцію помилок для наперед невідомої моделі.

Це породжує проблему, яка відома як явище перенавчання. Звернемося до завдання апроксимації деякої функції многочленом. Графіки многочленів часто мають вельми хитромудрі форми, і чим вище ступінь многочлена, тим складніше їх форма. Якщо є деякий набір даних, то можна поставити мету підібрати для нього апроксимуючий многочлен і таким чином отримати відповідну математичну модель для цього набору даних. Оскільки початкові дані, як правило, задані з погрішностями, то не можна вважати, що краща модель задається кривою, яка проходить точно через задані точки. Многочлен низького порядку може виявитися достатньо грубим для апроксимації даних, тоді як многочлен високого порядку може точно слідувати даним, приймаючи при цьому вельми хитромудру форму, що не має ніякого відношення до форми дійсної залежності. Остання ситуація і демонструє те, що називається явищем перенавчання.

При роботі з нейронними мережами користувач стикається з тією ж проблемою. Мережі з великою кількістю вагів дозволяють відтворювати дуже складні функції, і в цьому сенсі вони схильні до перенавчання. Мережа ж з невеликою кількістю вагів може опинитися недостатньо гнучкою, щоб змоделювати наявну залежність. Наприклад, одношарова лінійна мережа здатна відтворювати тільки лінійні функції. Якщо використовувати багатшарові лінійні мережі, то помилка завжди буде менша, але може свідчити не про хорошу якість моделі, а про те, що виявляється явище перенавчання.

Для того, щоб виявити ефект перенавчання, використовується механізм контрольної перевірки. Частина повчальних спостережень резервується як контрольні спостереження і не використовується при навчанні мережі. Натомість у міру роботи алгоритму ці спостереження застосовуються для незалежного контролю результату. Спочатку помилка мережі на повчальне і контрольному множині буде однаковою; якщо вони істотно відрізняються, то, мабуть, це

означає, що розбиття спостережень на 2 множини не забезпечило їх однорідність. У міру навчання мережі помилка убиває, і, поки навчання зменшує функцію помилок помилка на контрольній множині також убиватиме. Якщо ж контрольна помилка перестала убивати або стала рости, це указує на те, що мережа почала дуже близько слідувати початковим даним і навчання слід зупинити. В цьому випадку слід зменшити кількість нейронів або шарів, бо мережа є дуже могутнього для вирішення даного завдання. Якщо ж, навпаки, мережа має недостатню потужність, щоб відтворити наявну залежність, те явище перенавчання швидше за все спостерігатися не буде і обидві помилки - навчання і перевірки - не досягнуть достатньо малого рівня.

Що виникають при роботі з нейронними мережами проблеми відшукування глобального мінімуму або вибору розміру мережі призводять до того, що при практичній роботі доводиться експериментувати з великим числом мереж різних конфігурацій, деколи навчаючи кожен з них кілька разів і порівнюючи отримані результати. Головним критерієм вибору в цих випадках є контрольна погрішність. При цьому застосовується правило, згідно якому з двох нейронних мереж з приблизно рівними контрольними погрішностями слід вибрати ту, яка простіше.

Необхідність багатократних експериментів веде до того, що контрольна множина починає грати ключову роль у виборі моделі нейронної мережі, тобто стає частиною процесу навчання. Тим самим його роль як незалежного критерію якості моделі ослабляється, оскільки при великому числі експериментів виникає ризик перенавчання нейронної мережі на контрольній множині. Для того, щоб гарантувати надійність вибраної моделі мережі, резервують ще одне - тестова безліч спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на повчальній і контрольній множинах реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки 1 раз: якщо його використовувати повторно для коректування процесу навчання, то воно фактично перетвориться на контрольну множину.

Отже, процедура побудови нейронної мережі складається з наступних кроків:

вибору початкової конфігурації мережі; наприклад, у вигляді одного шару з числом нейронів, рівним $1/2$ загальної кількості входів і виходів;

моделювання і навчання мережі з оцінкою контрольної помилки і використанням додаткових нейронів або проміжних шарів;

виявлення ефекту перенавчання і коректування конфігурації мережі.

В.4 Властивість узагальнення

При описі процедури навчання нейронних мереж неявно використовувалося припущення, що повчальна, контрольна і тестова множини є представницькими для вирішуваної задачі. Зазвичай як повчальні беруться дані, випробувань на ряду прикладів. Якщо обставини змінилися, то закономірності, що мали місце у минулому, можуть більше не діяти.

Крім того, нейронна мережа може навчатися тільки на тих даних, які вона має в своєму розпорядженні. Припустимо, що відоме повчальна множина для системи стабілізації літака при польоті в спокійній атмосфері, а потрібно спроектувати систему стабілізації на основі нейронної мережі для умов польоту при сильних обуреннях. Тоді навряд чи можна чекати від мережі правильного рішення в абсолютно новій для неї ситуації.

Класичним прикладом непредставницької моделі нейронної мережі є наступна ситуація. При проектуванні системи машинного зору, призначеної для автоматичного розпізнавання цілей, мережа навчалася на 100 картинках, що містять зображення танків, і на 100 інших картинках, де танків не було. Після навчання мережі був досягнутий стовідсотково "правильний" результат. Але коли на вхід мережі були подані нові дані, вона безнадійно провалилася. У чому ж була причина? З'ясувалося, що фотографії з танками були зроблені в похмурий, дощовий день, а фотографії без танків - в сонячний день. Мережа навчилася уловлювати різницю в загальній освітленості. Щоб мережа могла результатив-

но працювати, її слід було навчати на даних, де б були присутні всі погодні умови і типи освітлення, при яких мережу передбачається використовувати, і це не кажучи ще про рельєф місцевості, вугілля і дистанцію зйомки і т.д.

Якщо мережа мінімізує загальну погрішність, великого значення набувають пропорції, в яких представлені дані різних типів. Мережа, навчена на 900 "хороших" і 100 "поганих" спостереженнях, спотворюватиме результат на користь хороших спостережень, оскільки це дозволить алгоритму зменшити загальну погрішність. Якщо в реальній ситуації "хороші" і "погані" об'єкти представлені в іншій пропорції, то результати, що видаються мережею, можуть виявитися невірними. Прикладом цього може бути завдання виявлення захворювань. Хай, наприклад, при звичайних обстеженнях в середньому 90% людей виявляються здоровими і мережа, таким чином, навчається на даних, в яких пропорція здорові/хворі рівна 90/10. Потім ця ж мережа застосовується для діагностики пацієнтів з певними скаргами, серед яких співвідношення здорові/хворі вже 50/50. В цьому випадку мережа ставитиме діагноз занадто обережно і не розпізнаватиме захворювання у деяких хворих. Якщо ж, навпаки, мережу навчити на даних "з скаргами", а потім протестувати на "звичайних" даних, то вона видаватиме підвищене число неправильних діагнозів про наявність захворювання. У таких ситуаціях почальні дані потрібно скоректувати так, щоб були враховані відмінності в розподілі даних (наприклад, можна повторити рідкісні спостереження або видалити що часто зустрічаються). Як правило, краще всього постаратися зробити так, щоб спостереження різних типів були представлені рівномірно, і відповідно цьому інтерпретувати результати, які видає мережа.

Здатність мережі, навченої на деякій безлічі даних, видавати правильні результати для достатньо широкого класу нових даних, у тому числі і не представлених при навчанні, називається властивістю узагальнення нейронної мережі.

Інший підхід до процедури навчання мережі можна сформулювати, якщо розглядати її як процедуру, зворотну моделюванню. В цьому випадку потрібно підібрати такі значення вагів і зсувів, які забезпечували б потрібну відповід-

ність між входами і бажаними значеннями на виході. Така процедура навчання носить назву адаптації і достатньо широко застосовується для настройки параметрів нейронних мереж.

В.5 Методи навчання нейронних мереж

Як тільки початкові ваги і зсуву нейронів встановлені користувачем або за допомогою датчика випадкових чисел, мережа готова для того, щоб почати процедуру її навчання. Мережа може бути навчена рішенню різних прикладних задач - апроксимації функцій, ідентифікації і управління об'єктами, розпізнавання образів, класифікації об'єктів і т.п. Процес навчання вимагає набору прикладів її бажаної поведінки - входів p і бажаних (цільових) виходів t ; під час цього процесу ваги і зсуву настраюються так, щоб мінімізувати деякий функціонал помилки. За умовчанням як такий функціонал для мереж з прямою передачею сигналів приймається середньоквадратична помилка між векторами виходу $\mathbf{a}$ і $\mathbf{t}$. Розглянемо декілька методів навчання для мереж з прямою передачею сигналів.

При навчанні мережі розраховується деякий функціонал, що характеризує якість навчання:

$$J = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^{S^M} (t_i^q - a_i^{qS^M})^2, \quad (\text{В.1})$$

де J - функціонал;

Q - об'єм вибірки;

M - число шарів мережі;

q - номер вибірки;

S^M - число нейронів вихідного шаруючи;

$\mathbf{a}^q = [a_i^{qM}]$ - вектор сигналу на виході мережі;

$\mathbf{t}^q = [t_i^q]$ - вектор бажаних (цільових) значень сигналу на виході мережі

для вибірки з номером q .

Потім за допомогою того або іншого методу навчання визначаються значення параметрів (вагів і зсувів), що настраюються, мережі, які забезпечують мінімальне значення функціонала помилки. Більшість методів навчання заснована на обчисленні градієнта функціонала помилки по параметрах, що настраюються.

В.5.1 Навчання одношарової мережі.

Найпростіше градієнт функціонала обчислюється для одношарових нейронних мереж. В цьому випадку $M=1$ і вираз для функціонала приймає вигляд:

$$J = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^{S^1} (t_i^q - a_i^{qS^1})^2 = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^{S^M} (t_i^q - f(n_i^q))^2 \quad i = \overline{1, S}, \quad (\text{B.2})$$

де $f(n_i^q)$ - функція активації;

$n_i^q = \sum_{j=0}^R w_{ij} p_j^q$ - сигнал на вході функції активації для i -го нейрона;

$\mathbf{p}^q = [p_i^q]$ - вектор вхідного сигналу;

R - число елементів вектора входу;

S - число нейронів в шарі;

w_{ij} - вагові коефіцієнти мережі.

Включимо вектор зсуву до складу матриці вагів $\mathbf{W} = [w_{ij}]$, $i = \overline{1, S}$, $j = \overline{1, R}$, а вектор входу доповнимо елементом, рівним 1.

Застосовуючи правило диференціювання складної функції, обчислимо градієнт функціонала помилки, припускаючи при цьому, що функція активації дифференцируема.

$$\frac{\partial J}{\partial w_{ij}} = - \sum_{q=1}^Q (t_i^q - f(n_i^q)) \frac{\partial(f(n_i^q))}{\partial w_{ij}} = - \sum_{q=1}^Q (t_i^q - f(n_i^q)) f'(n_i^q) p_j^q, \quad (\text{B.3})$$

Введемо позначення

$$\Delta_i^q = (t_i^q - f(n_i^q))f'(n_i^q)p_j^q = (t_i^q - a_i^q)f'(n_i^q)p_j^q, \quad i = \overline{1, S} \quad (\text{B.4})$$

і перетворимо вираз (B.3) таким чином

$$\frac{\partial J}{\partial w_{ij}} = - \sum_{q=1}^Q \Delta_i^q f'(n_i^q) p_j^q, \quad i = \overline{1, S} \quad (\text{B.5})$$

Отримані вирази спрощуються, якщо мережа лінійна. Оскільки $f'(n_i^q) = 1$. В цьому випадку вираз (B.3) приймає вигляд:

$$\frac{\partial J}{\partial w_{ij}} = (t_i^q - a_i^q) p_j^q, \quad i = \overline{1, S}, j = \overline{0, R}. \quad (\text{B.6})$$

Вираз (B.6) покладений в основу алгоритму WH, вживаного для навчання лінійних нейронних мереж.

Лінійні мережі можуть бути навчені і без використання ітераційних методів, а шляхом рішення наступної системи лінійних рівнянь:

$$\sum_{j=0}^R w_{ij} p_j^q = t_i^q, \quad i = \overline{1, S}, \quad q = \overline{1, Q} \quad (\text{B.7})$$

або у векторній формі запису:

$$\mathbf{Wp} = \mathbf{t}, \quad \mathbf{W} = [w_{ij}], \quad \mathbf{p} = [p_j^q], \quad \mathbf{t} = [t_i^q], \quad i = \overline{1, S}, \quad j = \overline{0, R}, \quad q = \overline{1, Q} \quad (\text{B.8})$$

Якщо число невідомих системи (B.7) рівно числу рівнянь, то така система може бути вирішена, наприклад, методом виключення Гауса з вибором головного елемента. Якщо ж число рівнянь перевищує число невідомих, то рішення шукається з використанням методу найменших квадратів.

В.5.2. Навчання багатошарової мережі.

Архітектура багатошарової мережі істотно залежить від вирішуваної задачі. Для лінійних нейронних мереж може бути встановлений зв'язок між сумарною кількістю вагів і зсувів з довжиною повчальної послідовності. Для інших типів мереж число шарів і нейронів в шарі часто визначається досвідом, інтуїцією проектувальника і евристичними правилами.

Навчання мережі включає декілька кроків:

- вибір початкової конфігурації мережі з використанням, наприклад, наступного евристичного правила: кількість нейронів проміжного шару визначається половиною сумарної кількості входів і виходів;
- проведення ряду експериментів з різними конфігураціями мережі і вибір тій, яка дає мінімальне значення функціонала помилки;
- якщо якість навчання недостатньо, слід збільшити число нейронів шаруючи або кількість шарів;
- якщо спостерігається явище перенавчання, слід зменшити число нейронів в шарі або видалити один або декілька шарів.

Нейронні мережі, призначені для вирішення практичних завдань, можуть містити до декількох тисяч параметрів, що настраюються, тому обчислення градієнта може зажадати вельми великих витрат обчислювальних ресурсів. З урахуванням специфіки багатошарових нейронних мереж для них розроблені спеціальні методи розрахунку градієнта, серед яких слід виділити метод зворотного розповсюдження помилки.

В.5.3 Метод зворотного розповсюдження помилки.

Термін "зворотне розповсюдження" відноситься до процесу, за допомогою якого можуть бути обчислені похідні функціонала помилки по параметрах мережі. Цей процес може використовуватися у поєднанні з різними стратегіями оптимізації. Існує багато варіантів і самого алгоритму зворотного розповсюдження. Звернемося до одного з них.

Розглянемо вираз для градієнта критерію якості по вагових коефіцієнтах для вихідного шару M :

$$\frac{dJ}{dw_{ij}^M} = \frac{\partial}{\partial w_{ij}^M} \left(\frac{1}{2} \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM})^2 \right) = \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) \frac{\partial a_k^{qM}}{\partial w_{ij}^M} ; \quad i = \overline{1, S^M} ; \quad j = \overline{0, S^{M-1}}, \quad (\text{B.9})$$

де S^M - число нейронів в шарі;

a_k^{qM} - k -й елемент вектора виходу шару M для елемента вибірки з номером q .

Правило функціонування шару M :

$$a_k^{qM} = f_M \left(\sum_{i=0}^{S^{M-1}} w_{ij}^M a_i^{q(M-1)} \right), \quad m = \overline{1, S^M} . \quad (\text{B.10})$$

З рівняння (B.10) виходить

$$\frac{\partial a_k^{qM}}{\partial w_{ij}^M} = \begin{cases} 0, & k \neq i \\ f'(n_i^{qM}) a_j^{q(M-1)}, & k = i \end{cases} \quad i = \overline{1, S^M} ; \quad j = \overline{0, S^{M-1}} \quad (\text{B.11})$$

Після підстановки (B.11) в (B.9) маємо:

$$\frac{dJ}{dw_{ij}^M} = - \sum_{q=1}^Q (t_i^q - a_i^{qM}) f'(n_i^{qM}) a_i^{q(M-1)},$$

Якщо позначити

$$\Delta_i^{qM} = (t_i^q - a_i^{qM}) f'(n_i^{qM}), \quad i = 1, \dots, S^M \quad (\text{B.12})$$

то отримаємо

$$\frac{dJ}{dw_{ij}^M} = - \sum_{q=1}^Q \Delta_i^{qM} a_i^{q(M-1)} ; \quad i = \overline{1, S^M} ; \quad j = \overline{0, S^{M-1}} \quad (\text{B.13})$$

Перейдемо до виведення співвідношень для настройки вагів w_{ij}^{M-1} шару $M-1$

$$\begin{aligned}
\frac{dJ}{dw_{ij}^{M-1}} &= - \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) f'_M(n_k^{qM}) \frac{\partial n_k^{qM}}{\partial a_i^{q(M-1)}} \frac{\partial a_k^{q(M-1)}}{\partial w_{ij}^{M-1}} a_i^{q(M-1)} = \\
&= - \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) f'_M(n_k^{qM}) w_k^M \frac{\partial n_k^{qM}}{\partial a_i^{q(M-1)}} f'_{M-1}(n_i^{q(M-1)}) a_j^{q(M-2)} = \\
&= - \sum_{q=1}^Q \Delta_i^{q(M-1)} a_j^{q(M-2)}, \tag{B.14}
\end{aligned}$$

де

$$\Delta_i^{qM} = \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) f'_M(n_k^{qM}) w_k^M f'_{M-1}(n_i^{q(M-1)}) = \left(\sum_{k=1}^{S^M} \Delta_k^{qM} \right) f'_{M-1}(n_i^{q(M-1)}), \quad i = \overline{1, S^{M-1}},$$

Для шарів $M-2, M-3, \dots, 1$ обчислення часних похідних критерію J по елементах матриць вагових коефіцієнтів виконується аналогічно. У результаті отримуємо наступну загальну формулу:

$$\frac{dJ}{dw_{ij}^{M-1}} = \sum_{q=1}^Q \Delta_i^{q(r-1)} a_j^{q(r-1)}, \quad r = \overline{1, M}, \quad i = \overline{1, S^r}, \quad i = \overline{0, S^{r-1}}, \tag{B.15}$$

де r - номер шару.

$$\Delta_i^{qr} = \left(\sum_{k=1}^{S^{r-1}} \Delta_k^{q(r-1)} w_{ki}^{r+1} \right) f'_r(n_i^{qr}), \quad r = \overline{1, M-1}$$

$$\Delta_i^{qM} = (t_i^q - a_i^{qM}) f'_M(n_i^{qM}), \quad i = \overline{1, S^{M-1}}$$

На рис. В.1 представлена схема обчислень, відповідна виразу (B.15). На схемі символом $*$ позначена операція поелементного множення векторів, а символом $**$ - множення вектора Δ на $\mathbf{a}^T$; символ, що позначає номер елементу вибірки, скорочено опущений.

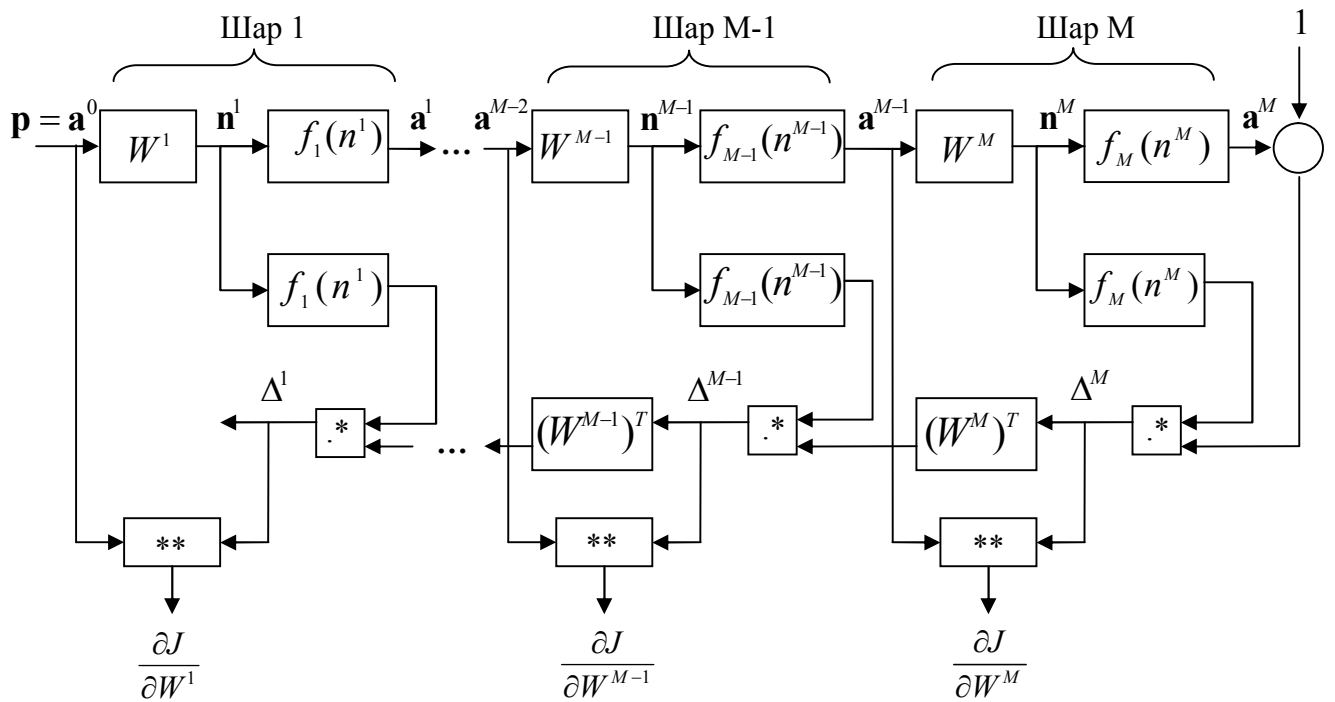


Рисунок В.1 – Схема обчислень часних похідних критерію J по елементах матриць вагових коефіцієнтів

В.5.4 Характеристика методів навчання.

Методи, використовувані при навчанні нейронних мереж, багато в чому аналогічні методам визначення екстремуму функції декілька змінних. У свою чергу, останні діляться на 3 категорії - методи нульового, першого і другого порядку.

У методах нульового порядку для знаходження екстремуму використовується тільки інформація про значення функції в заданих точках.

У методах першого порядку використовується градієнт функціонала помилки по параметрах, що настраюються

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a_k \mathbf{g}_k \quad (\text{B.16})$$

де $\mathbf{x}_k$ - вектор параметрів;

a_k - параметр швидкості навчання;

$\mathbf{g}_k$ - градієнт функціонала, відповідні ітерації з номером k .

Вектор в напрямі, протилежному градієнту, указує напрям найкоротшого спуску по поверхні функціонала помилки. Якщо реалізується рух в цьому напрямі, то помилка зменшуватиметься. Послідовність таких кроків врешті-решт приведе до значень параметрів, що настроюються, що забезпечують мінімум функціонала. Певну трудність тут викликає вибір параметра швидкості навчання a_k . При великому значенні параметра a_k збіжність буде швидкою, але існує небезпека пропустити рішення або піти в неправильному напрямі. Класичним прикладом є ситуація, коли алгоритм дуже поволі просувається по вузькому яру з крутими схилами, перестрибуючи з одного на іншій. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, проте при цьому буде потрібно дуже багато ітерацій. Залежно від прийнятого алгоритму параметр швидкості навчання може бути постійним або змінним. Правильний вибір цього параметра залежить від конкретного завдання і зазвичай здійснюється досвідченим шляхом; у разі змінного параметра його значення зменшується у міру наближення до мінімуму функціонала.

У алгоритмах зв'язаного градієнта пошук мінімуму виконується уздовж зв'язаних напрямів, що забезпечує зазвичай швидшу збіжність, ніж при найскорішому спуску. Всі алгоритми зв'язаних градієнтів на першій ітерації починають рух у напрямі антиградієнта

$$\mathbf{p}_0 = -\mathbf{g}_0 \quad (\text{B.17})$$

Тоді напрям наступного руху визначається так, щоб воно було зв'язане з попереднім. Відповідний вираз для нового напрямку руху є комбінацією нового напрямку найскорішого спуску і попереднього напрямку:

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}, \quad (\text{B.18})$$

де $\mathbf{p}_k$ - напрям руху

$\mathbf{g}_k$ - градієнт функціонала помилки

β_k - коефіцієнт відповідають ітерації з номером k . Коли напрям спуску визначений, то нове значення вектора параметрів $\mathbf{x}_{k+1}$, що настроюються, обчислюється за формулою

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a_k \mathbf{p}_k \quad (\text{B.19})$$

Методи другого порядку вимагають знання других похідних функціонала помилки. До методів другого порядку відноситься метод Ньютона. Основний крок методу Ньютона визначається по формулі

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{g}_k \quad (\text{B.20})$$

де $\mathbf{x}_k$ - вектор значень параметрів на k -й ітерації;

$\mathbf{H}$ - матриця других приватних похідних цільової функції, або матриця Гессе;

$\mathbf{g}_k$ - вектор градієнта на k -й ітерації.

У багатьох випадках метод Ньютона сходиться швидше, ніж методи зв'язаного градієнта, але вимагає великих витрат із-за обчислення гессиана. Для того, щоб уникнути обчислення матриці Гессе, пропонуються різні способи її заміни наближеними виразами, що породжує так звані квазиньютонові алгоритми (алгоритм методу січних площин OSS, алгоритм LM Левенберга-Марквардта).

ДОДАТОК Г

УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ З ПРОГНОЗОМ

Г.1 Алгоритм нейроуправління з прогнозом

У роботі використовується ефективна реалізація узагальненого управління з прогнозом (Generalized Predictive Control, GPC) з використанням багатошарової прямонаправленої нейронної мережі, як нелінійній моделі об'єкту управління [41]. Завдяки використанню оптимізаційного алгоритму Ньютона-Рафсона, число ітерацій, необхідних для збіжності, значно менше, ніж при використанні інших методів. Головні витрати алгоритму Ньютона-Рафсона в обчисленні Гесиана, але навіть з цими додатковими розрахунками низьке число ітерацій робить алгоритм Ньютона-Рафсона швидшим, ніж інші методи, при цьому він може використовуватися для управління в режимі реального часу (real-time control).

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

Узагальнене управління з прогнозом, введене Кларком і його співробітниками в 1987, належить до класу методів цифрового управління, званих модель-орієнтоване управління з прогнозом (Model-Based Predictive Control, MBPC) [42-44]. MBPC техніки були успішно проаналізовані і впроваджені в процес управління виробництвом в кінці 1970-х і продовжують використовуватися, оскільки вони системно враховують обмеження реальних об'єктів в режимі реального часу. GPC застосовуються в управлінні немінімально фазовими об'єктами, нестійкими об'єктами і об'єктами із змінним або невідомим часом затримки. Узагальнене управління з прогнозом також є робастним по відношенню до помилок моделювання, перевизначення і недовизначення значень параметрів і шумів датчиків [42]. GPC було спочатку розроблене для моделей лінійних об'єктів, що приводить до рівнянь, які можуть бути вирішені аналіти-

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

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

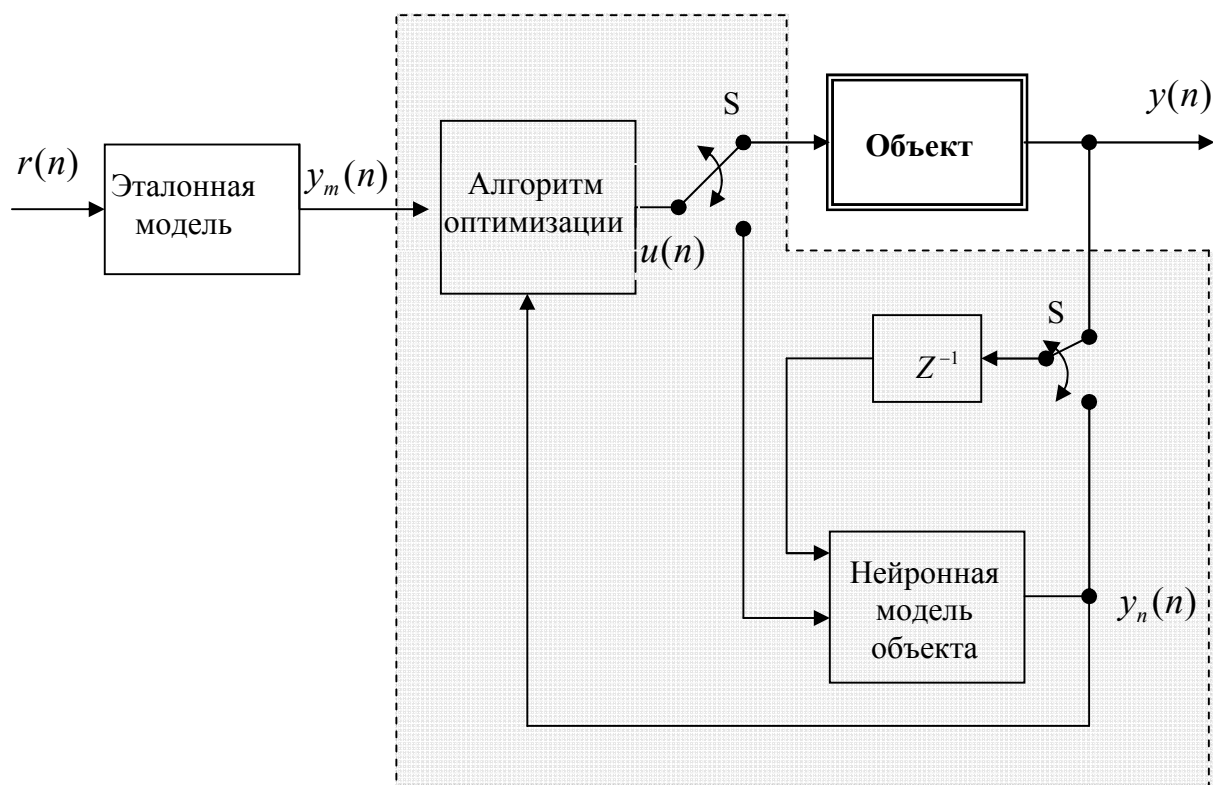


Рисунок Г.1 – Схема системи узагальненого нейроуправління з прогнозом

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

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

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

Узагальнений алгоритм нейроуправління з прогнозом складається з наступних основних кроків.

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

Якість обчислення при реалізації узагальненого управління з прогнозом в основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізацій, які реалізуються в CFM, такі як: Non-

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

Якість обчислення при реалізації узагальненого управління з прогнозом в основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізації, які реалізуються в CFM, такі як: Non-gradient [45], Simplex [46] і Successive Quadratic Programming [47, 48]. Вибір методу мінімізації може бути заснований на наступних критеріях: число ітерацій, обчислювальні витрати і точність рішення. У загальному випадку, ці алгоритми вимагають великого числа ітерацій, що робить управління в режимі реального часу скрутним. Дуже невелике число публікацій присвячені реалізаціям управління в режимі реального часу для об'єктів, що мають великі постійні часу [47, 48]. Для можливості використання в системах з об'єктами, що мають малі постійні часу, потрібний швидший оптимізаційний алгоритм. Алгоритм Ньютона-Рафсона володіє квадратичною збіжністю, тоді як інші алгоритми мають нижчу швидкість збіжності. Вища швидкість збіжності алгоритму Ньютона-Рафсона вимагає високих обчислювальних витрат, але виправдовується показником швидкості збіжності.

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

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

де N_1 – нижня межа прогнозу;

N_2 – верхня межа прогнозу;

N_u – діапазон управління;

y_m – бажана траєкторія;

y_n – передбачений вихід нейронної мережі;

λ – ваговий множник;

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

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

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

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

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

кожної ітерації розраховується також значення управляючого вектора

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

де $k=1, \dots$ – номер ітерації.

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

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

де якобіан позначений, як

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

а гесіан, як

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

Рішення рівняння (8.2) вимагає явного звернення матриці Гессе. Цей процес може зажадати великих обчислювальних витрат. Одним з підходів, використовуваних, щоб уникнути звернення матриці, служить LU декомпозиція [49] для отримання вхідного вектора $\mathbf{U}(k+1)$. Це досягається шляхом запису рівняння (8.2) у вигляді системи лінійних рівнянь $\mathbf{Ax} = \mathbf{b}$, що приводить до виразу

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

де

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

У даній формі рівняння (8.3) може бути вирішене за допомогою двох процедур, описаних в [49]: процедури LU декомпозиції, ludcmp і процедури рішення системи лінійних рівнянь lubksb. Після того, як обчислений вектор $\mathbf{x}$. Процедура повторюється до тих пір, поки зміна кожній компоненти $\mathbf{U}(\mathbf{k} + 1)$ не стає меншим деякого ε . При рішенні системи відносно $\mathbf{x}$ для кожної ітерації алгоритму Ньютона-Рафсона необхідне обчислення кожного елементу якобіана і гессиана. Елемент якобіана з індексом h має наступний вигляд:

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

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

дельта функції, яка визначається як

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

отримаємо

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

Елемент з індексами m, h гессиана визначається таким чином:

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

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

Можна скористатися також визначенням дельта-функції для виразу

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

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

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

Г.2 Структура нейронної мережі для моделювання об'єкту

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

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

ким чином, що безліч входних значень продукує бажані вихідні значення. Помилка формується між відповідями нейронної мережі $y_n(n)$ і об'єкту $y(n)$. Надалі ця помилка використовується для оновлення вагів за допомогою одного з розглянутих вище методів, наприклад градієнтного спуску [50]. Цей процес повторюється до тих пір, поки помилка не досягне допустимого рівня.

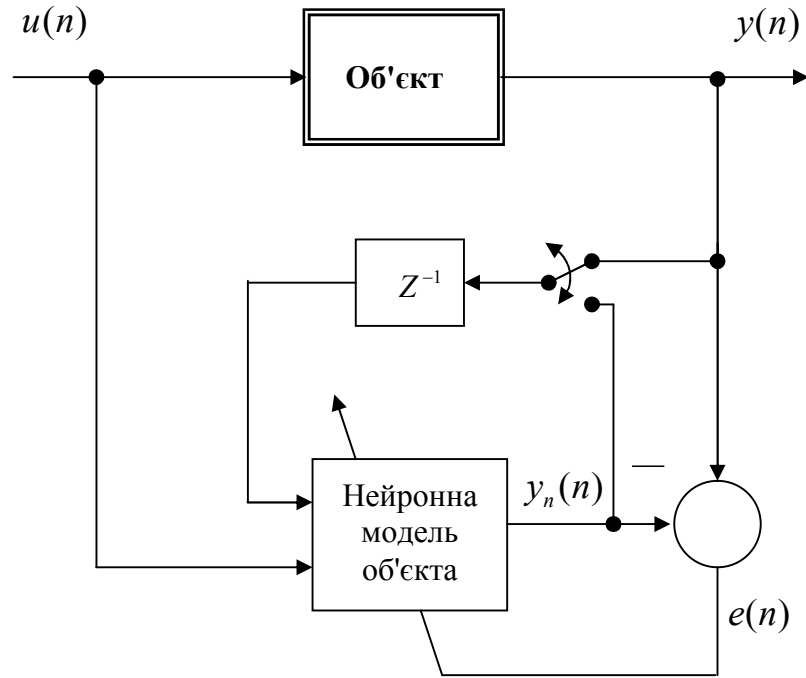


Рисунок Г.2 – Схема тренування мережі в режимі offline

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

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

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

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

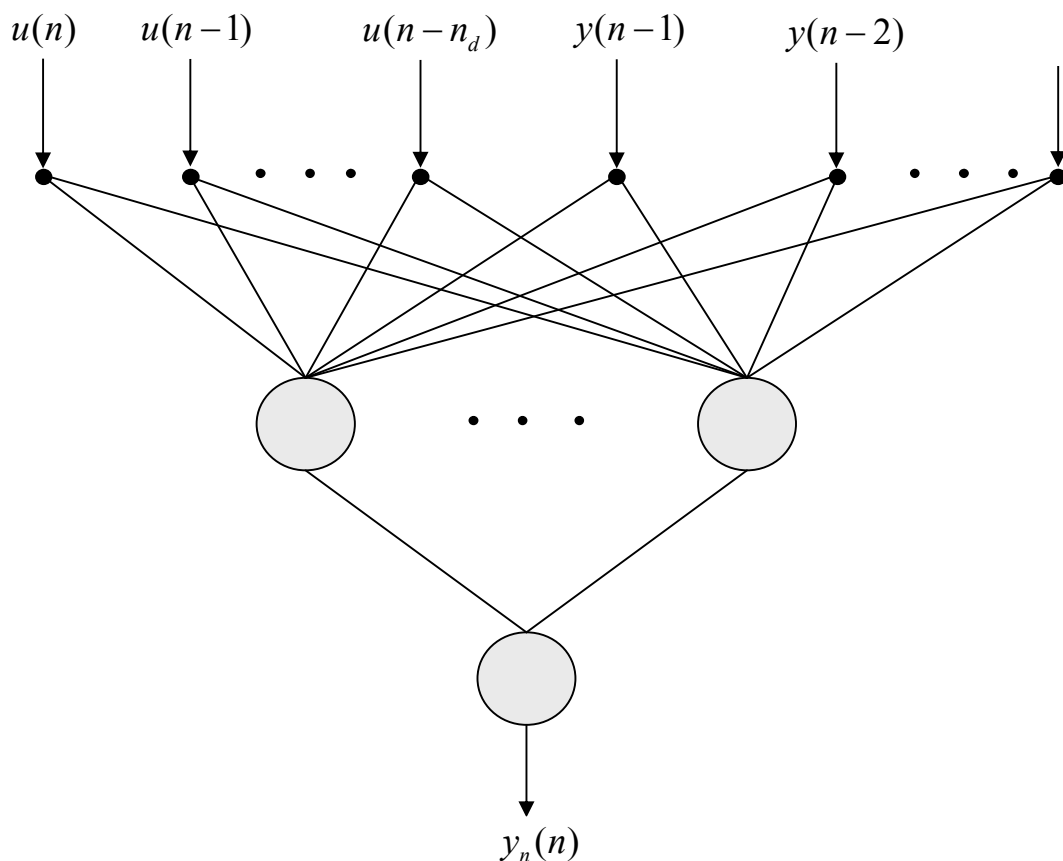


Рисунок Г.3 – Багатошарова нейронна мережа прямого розповсюдження з елементами затримок за часом

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

Рівняння мережі з даною архітектурою виглядають таким чином:

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

і

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

де $y_n(n)$ – вихід нейронної мережі;

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

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

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

L – число шарів;

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

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

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

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

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

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

b_j – зсув j -го прихованого нейрона;

b – зсув нейрона виходу.

Г.3 Прогноз з використанням нейронної мережі

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

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

і

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

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

Г.4 Похідні рівнянь нейронної мережі

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

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

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

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

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

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

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

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

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

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

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

i

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

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

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

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

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом
% механізму підйому шахтної підіймальної установки
% Система ТП-Д, трьохфазова
% 2024 рік

% Вихідні дані
ktp=66;
Tm=0.005;
Tms=2*Tm;
Re=0.0228;
Te=0.069;
CF=165;
kdc=0.0038;
kdv=6.7;
In=3180;;
Mn=In*CF;
Jd=532864;
Jk=51097;
Jm=145990;
Js=Jd+Jm+Jk;
Js=729823;
w0=2;
c=(w0^2)*((Jd+Jk/2)*(Jm+Jk/2))/((Jd+Jk/2)+(Jm+Jk/2));
c1=2*c;
c2=2*c1;
b1=0.01*c1;
b2=0.01*c2;

kic=Re/(2*Tm*kdc*ktp);
kpc=kic*Te;
kiv=(kdc*Js)/(32*Tm^2*CF*kdv);
kpv=8*Tm*kiv;

% Матриці
A=[-b2/Jm 1/Jm b2/Jm zeros(1,6);
-c2 0 c2 zeros(1,6);
b2/Jk -1/Jk (-b1-b2)/Jk 1/Jk b1/Jk zeros(1,4);
0 0 -c1 0 c1 zeros(1,4);
0 0 b1/Jd -1/Jd -b1/Jd CF/Jd zeros(1,3);
0 0 0 0 -CF/(Re*Te) -1/Te 1/(Re*Te) 0 0;
0 0 0 0 (-kpv*kpc*ktp*kdv)/Tm -(kdc*kpc*ktp)/Tm -1/Tm ktp/Tm
ktp*kpc/Tm;
0 0 0 0 -kdv*kpv*kic -kic*kdc 0 0 kic;
0 0 0 0 -kdv*kiv zeros(1,4)]
B=[0 -1/Jm;
zeros(5,2);
```

```
(kpv*kpc*ktp)/Tm 0;  
kpv*kic 0;  
kiv 0]  
%C=eye(9,9)  
C=zeros(6,9);  
C(1,1)=1;C(2,2)=1;C(3,3)=1; C(4,4)=1;  
C(5,5)=1;C(6,6)=1;  
D=zeros(6,2)
```