

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

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

НЕЙРОМЕРЕЖЕВІ СИСТЕМИ УПРАВЛІННЯ ОСНОВНИМИ
МЕХАНІЗМАМИ МОСТОВОГО КРАНА. ТЕМА КОМПЛЕКСНА.
СПЕЦІАЛЬНА ЧАСТИНА. СИНТЕЗ І ДОСЛІДЖЕННЯ СИСТЕМИ
УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ МОСТА
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)

_____/ Андрій ШАПОВАЛ
(підпис) (ім'я та прізвище)

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

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

«До захисту допущено»

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

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

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

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

Навчально-науковий інститут «Українська 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)

« » 2024p.

ЗАВДАННЯ

на кваліфікаційну роботу магістерського рівня вищої освіти

студенту (ці) _____ Андрій ШАПОВАЛ
(ім'я, прізвище)

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

6. Консультант:

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

7. Дата видачі завдання “ 9 вересня 2024 р.

Керівник

(підпис)

Геннадій КАНЮК
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Андрій ШАПОВАЛ
(ім'я, прізвище)

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфікаційної роботи**

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про виконання завдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму переміщення моста мостового крана.	09.09.2024- 18.09.2024	
2	Розрахунок системи підлеглого регулювання координат електроприводу з оптимізацією контурів за модульним та симетричним критеріями. Моделювання системи на ЕОМ.	19.09.2024- 15.10.2024	
3	Дослідження системи з урахуванням пружних елементів. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ЕОМ.	16.10.2024- 05.11.2024	
4	Синтез нейромережової системи управління на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2024- 25.11.2024	
5	Оформлення роботи	26.11.2024- 05.12.2024	

Студент (ка)

(підпис)

Андрій ШАПОВАЛ
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 182 сторінки, 62 рисунки, 4 таблиці, 73 використаних джерел, 5 додатків.

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

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

В результаті виконання магістерської кваліфікаційної роботи науково обґрунтовано перспективність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму переміщення моста мостового крана. Розроблена система управління координат електроприводу механізму. Розраховані параметри системи автоматичного управління. Проведено моделювання системи на ЕОМ.

Розроблена математична модель системи з урахуванням пружних елементів механічної частини електроприводу. Розраховані динамічні характеристики двомасової системи. Розглянута структура нейромережевої системи управління і виконаний синтез нейрорегулятора NN Predictive Controller в системі MATLAB. Проведено моделювання нейромережевої системи управління з нейрорегулятором на ЕОМ. Встановлено, що синтезована нейромережева система має високі показники якості функціонування.

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

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

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

ABSTRACT

An explanatory message consists of: 182 pages, 62 figures, 4 tables, 73 used sources, и 5 additions.

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

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

As a result of the master's qualification work scientifically grounded perspective the use of neural network technology to control multimass electromechanical system of the moving bridge mechanism of the bridge crane. The control system of coordinates of mechanism electric drive is worked out. The parameters of automatic control system are calculated. The simulation of the system is executed on a computer.

Worked out the mathematical model of the system which takes into account the elastic elements of the mechanical part of the electric drive is developed. Dynamic descriptions of the two-mass system are calculated.. The structure of neural network control system are considered and made a synthesis of neuroregulator NN Predictive Controller in MATLAB. The simulation of neural network control system with neuroregulation is conducted on a computer. It is set that synthesized the neural network system has high indexes of quality of functioning.

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

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

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

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

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

ЗМІСТ

ВСТУП	8
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	17
1.3 Вимоги до електроустаткування мостових кранів	29
1.4 Обґрунтування вибору системи електроприводу механізму підйому мостового крана	30
1.5 Постановка задачі дослідження	33
ВИСНОВКИ ДО РОЗДІЛУ 1	36
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ МОСТА МОСТОВОГО КРАНА	38
2.1 Технічні характеристики мостового крана. Вимоги до системи управління	38
2.2 Розрахунок потужності електроприводу механізму переміщення моста. Вибір приводних двигунів	39
2.3 Перевірка двигунів по тепловому режиму і запасу зчеплення при пуску і гальмуванні	41
2.4 Вибір основного електроустаткування і розрахунок параметрів силового кола електроприводу	43
2.5 Вибір структури системи автоматичного управління	49
2.6 Синтез послідовного корегуючого пристрою контура струму	50
2.7 Синтез послідовного корегуючого пристрою контура ЕРС	57
2.8 Дослідження динамічних характеристик системи	65
ВИСНОВКИ ДО РОЗДІЛУ 2	69
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ	71

3.1 Математична модель двомасової системи	71
3.2 Розрахунок перехідних процесів двомасової систем на ЕОМ	79
ВИСНОВКИ ДО РОЗДІЛУ 3	85
РОЗДІЛ 4. СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	86
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	86
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	88
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	91
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	134
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	136
ВИСНОВКИ ДО РОЗДІЛУ 4	142
ВИСНОВКИ	144
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	146
ДОДАТОК А. СТРУКТУРА ШТУЧНОГО НЕЙРОНА	153
ДОДАТОК Б. АРХІТЕКТУРА ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ.....	159
Б.1 Архітектури найбільш поширених ШНМ.....	159
Б.2 Штучні нейронні мережі прямого розповсюдження	161
ДОДАТОК В. НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ	165
В.1 Загальні питання навчання нейронних мереж.....	165
В.2 Явище перенавчання	167
В.3 Властивість узагальнення.....	169
В.4 Метод зворотного розповсюдження помилки.....	171
ДОДАТОК Г. ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ	176
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	181

ВСТУП

Актуальність теми. Удосконалення систем автоматизованого електроприводу з використанням сучасних наукових і технічних досягнень є необхідною умовою для вирішення завдань підвищення ефективності суспільного виробництва, збільшення продуктивності праці та покращення якості продукції.

На сьогодні в нашій країні активно розробляються нові конструкції підйомно-транспортних комплексів, автоматизованих машин і механізмів, машин циклічної дії та конвейєрних ліній, що сприяє значному збільшенню виробництва обладнання для механізації вантажно-розвантажувальних і складських робіт. Крани є важливою складовою частиною комплексної механізації у всіх секторах народного господарства.

Більшість вантажопідйомних машин, що виготовляються в країні, оснащені електричними приводами для основних робочих механізмів. Відповідно, ефективність роботи цих машин значною мірою залежить від якості електрообладнання кранів. Електроприводи таких машин працюють у режимі короткочасної експлуатації з частими включеннями, мають широкий спектр регулювання швидкості та стикаються з високими навантаженнями під час розгону та гальмування.

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

Протягом останніх десятиліть виникло багато управлінських методів, включаючи ті, які базуються на адаптивному управлінні. Це стало можливим завдяки прогресу в сучасній теорії управління та теорії стійкості. Можливість

автоматично адаптуватися до змін у динаміці керованого об'єкта та зовнішнього середовища зробила адаптивні системи автоматичного управління більш привабливими для різноманітних практичних застосувань. Однак ефективність таких систем управління реальними об'єктами значною мірою залежить від якості математичних моделей, які використовуються. Ці моделі повинні максимально точно відображати властивості вивченого об'єкта і одночасно бути зручними для використання алгоритмів управління. Недостатня кількість інформації про умови функціонування, властивості об'єктів і перешкоди вимагають використання адаптивного підходу при управлінні ними. Цей підхід дозволяє використовувати спрощені, зокрема, лінійні моделі. Втім, хоча такий підхід може значно зменшити невизначеність і реалізувати ефективне управління у деяких випадках, обмеження лінійними моделями не завжди забезпечує досягнення бажаного результату. Тому розробка систем управління, яка базується на адаптивному підході, у поєднанні з методами теорії штучних нейронних мереж, є більш ефективною.

Нейронні мережі є галуззю штучного інтелекту, що використовує подібні до живих організмів процеси для обробки сигналів. Однією з ключових особливостей такої мережі, яка вказує на її великий потенціал, є можливість паралельної обробки інформації різними шляхами. З великою кількістю зв'язків між нейронами, це сприяє швидкості обробки інформації, і часто навіть дозволяє перетворення сигналів у реальному часі. Іншим важливим аспектом є стійкість мережі до помилок: у разі пошкодження деяких зв'язків, функції цих пошкоджених зв'язків можуть бути взяті на себе сусідніми ланками, що забезпечує безперебійну роботу мережі.

Інша не менш важлива властивість – здібність до навчання і узагальнення накопичених знань. Нейронна мережа володіє рисами штучного інтелекту. На-тренованана обмеженій множині даних мережа здатна узагальнювати отриману інформацію і показувати добрі результати даних, що не використалися в процесі навчання.

Ця мережа відрізняється можливістю впровадження через технологію надвеликого ступеня інтеграції. Оскільки її складові елементи мають малі відмінності між собою, але використовуються у великих кількостях, є потенціал для створення універсальних процесорів з однорідною структурою, які зможуть обробляти різноманітні дані. Останнім часом, через поширення пристроїв з високим рівнем інтеграції та поширення використання обчислювальної техніки, зацікавленість у нейронних мережах значно зросла, що призвело до значних досягнень у їх вивченні. Це створило базу для нових технологічних рішень у таких областях, як обробка відеоінформації, штучне розпізнавання, управління складними системами, аудіозаписів і так далі. У практичних сценаріях, штучні нейронні мережі зазвичай використовуються як частина управління або як система, що генерує виконавчі сигнали для інших підсистем з відмінною методологічною базою.

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 (НСУ). Найбільше розповсюдження для цілей управління знайшли багат шарові нелінійні нейронні мережі.

Розробка нових і вдосконалення існуючих систем управління основних механізмів мостового крана є важливим народно-господарським завданням. Тому тема роботи є актуальною.

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

1. Аналіз існуючих систем, а також можливостей і перспектив застосування 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 пересування моста мостового крана з урахуванням пружних елементів.

Предметом дослідження є моделі і методи синтезу нейромережової системи управління двомасовою електромеханічною системою механізму переміщення моста мостового крана.

Методи дослідження. Всі теоретичні положення дисертації базуються на фундаментальних положеннях теорії автоматичного управління. Методи математичного моделювання 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 переміщення моста мостового крана.

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

3. Вперше запропонована математична модель системи управління електроприводом переміщення моста мостового крана з урахуванням пружних елементів у вигляді багат шарової прямонаправленої мережі – багат шарового персеプトрона.

4. Здійснений подальший розвиток математичних моделей об'єктів управління систем із складними кінематичними зв'язками у вигляді двомасових електромеханічних систем.

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

Практичне значення одержаних результатів полягає в розробці нейромережевої системи управління двомасовою електромеханічною системою механізму переміщення моста мостового крана, що забезпечує високоякісне регулювання.

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

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

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

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 182 сторінки, серед них: 62 рисунки, з яких 39 рисунків за текстом, 23 рисунки на 22 окремих сторінках, 4 таблиці, 5 додатків на 30 сторінках, список літературних джерел з 73 найменування на 7 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 перцептронів 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 мадаліна 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) «Перцептрони» [21], в якій доводиться принципова обмеженість мо-

жливостей 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 «Нітап 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 будувати внутрішні уявлення про зовнішній світ, прогнозувати поведінку об'єкту управління, але так само володіє здатністю генерувати оптимальне управління відповідно до вимог еталонної моделі. У численних публікаціях 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 багатошарові прямонаправлені мережі, що характеризуються наявністю зв'язків між нейронами тільки в прямому напрямку без зворотних зв'язків всередині мережі – багатошаровий персептрон. В додатку А розглянуто структуру штучного нейрона., 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.

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 з 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 кри-

терій 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].

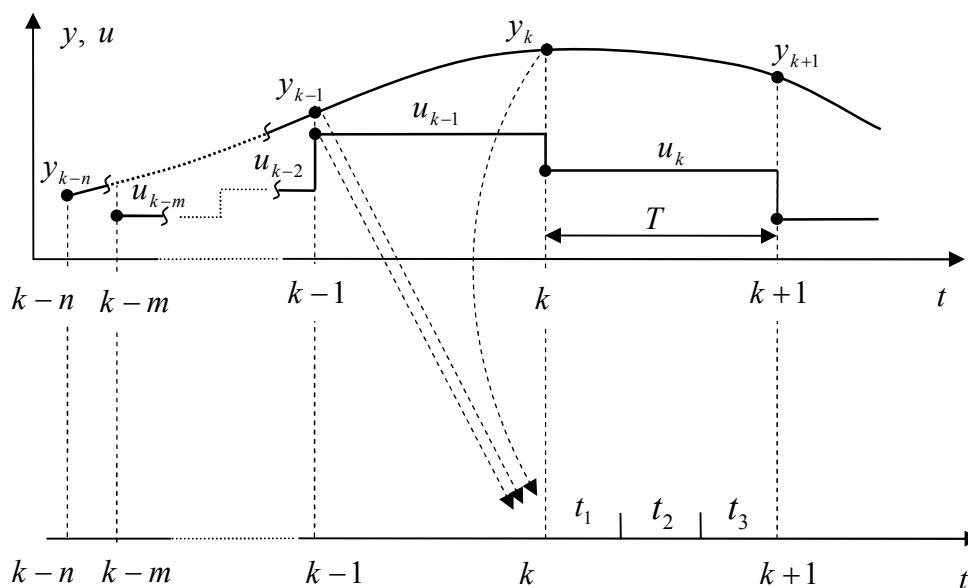


Рисунок. 1.1 – Регулювання і тренування мережі в реальному часі

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 випробу-

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

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 Псалтіса і ін. [28], ці методи названі «*General learning*» і «*specialized learning*». Відмінності структур прямого інверсного регулювання з offline і online тренуванням розглядаються нижче.

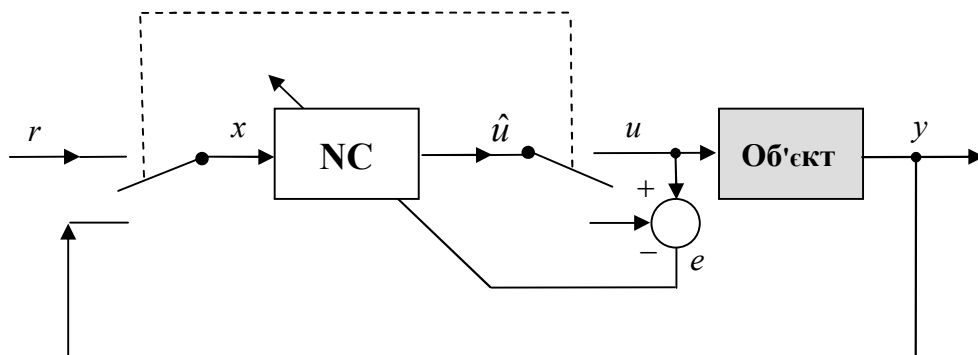


Рисунок 1.2 – Пряме інверсне регулювання з offline тренуванням

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.

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.

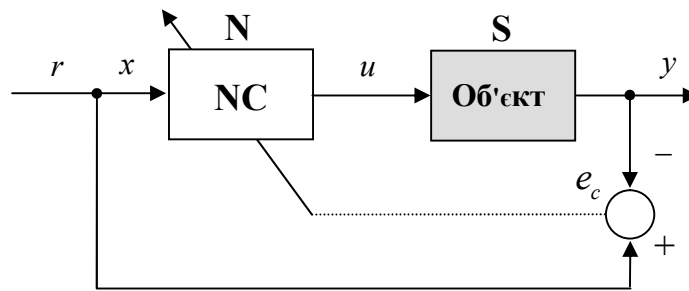


Рисунок 1.3 – Пряме інверсне регулювання з online тренуванням

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.

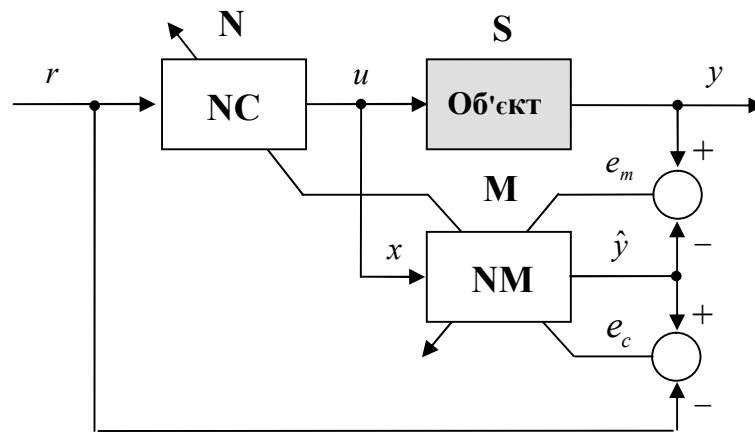


Рисунок 1.4– Непряме інверсне нейрорегулювання

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:

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

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 **НС** 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 та-

кож 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 нейромODEЛЕЙ може служити яскравим прикладом того, як нейротехнології можуть бути інтегровані в класичні адаптивні системи регулювання. Структурна схема системи представлена на рис. 1.5.

Аналогічно непрямому нейрорегулюванню, спочатку здійснюється тренування нейромODEЛІ **NM** у режимі *offline*. З допомогою **NM** формується матриця Якобі для корекції сигналу помилки e_n щоб забезпечити тренування інверсного нейрорегулятора **NC** 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.

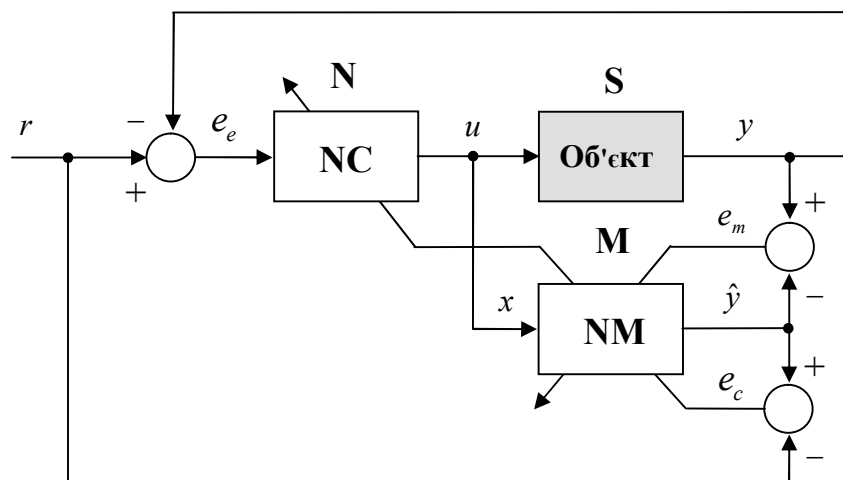


Рисунок 1.5 – Система регулювання з внутрішньою моделлю

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 $\mathbf{NC}_{\text{FF}}$ і $\mathbf{NC}_{\text{FB}}$ 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. $\mathbf{NC}_{\text{FF}}$ 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].

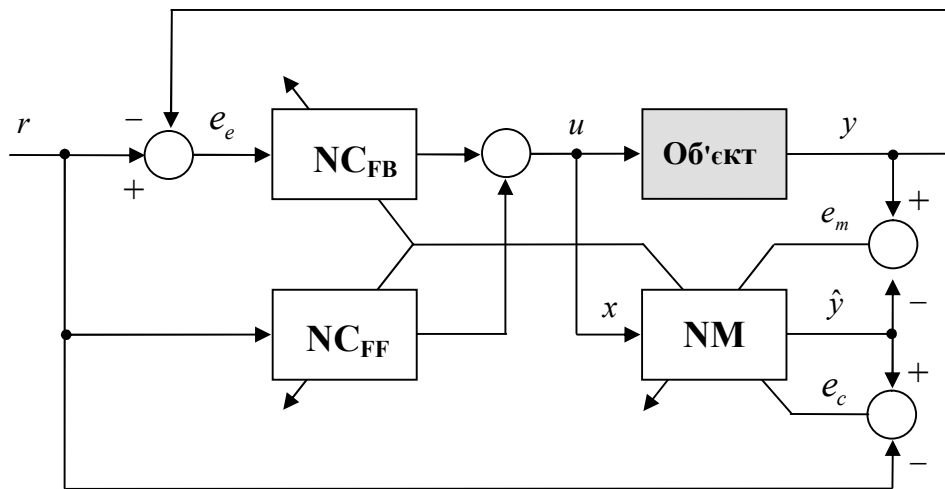


Рисунок 1.6 – Feedforward-Feedback-Error-регулятор

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 допомогою 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.

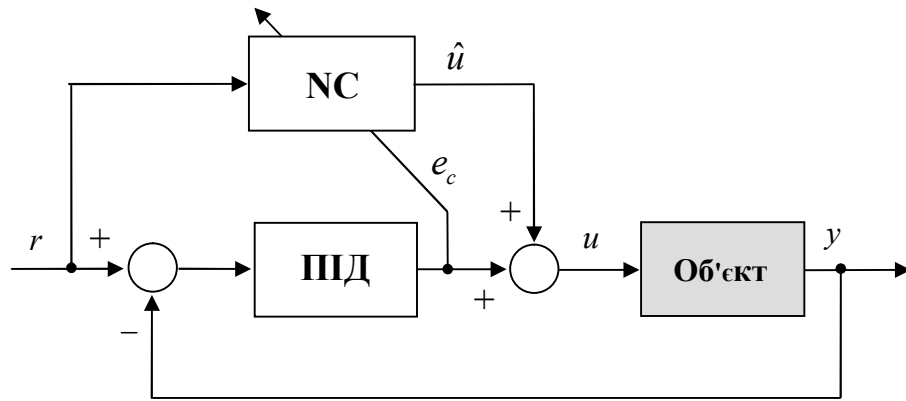


Рисунок 1.7 – Регулювання з Feedback-Error-Learning

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 справедливим, 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].

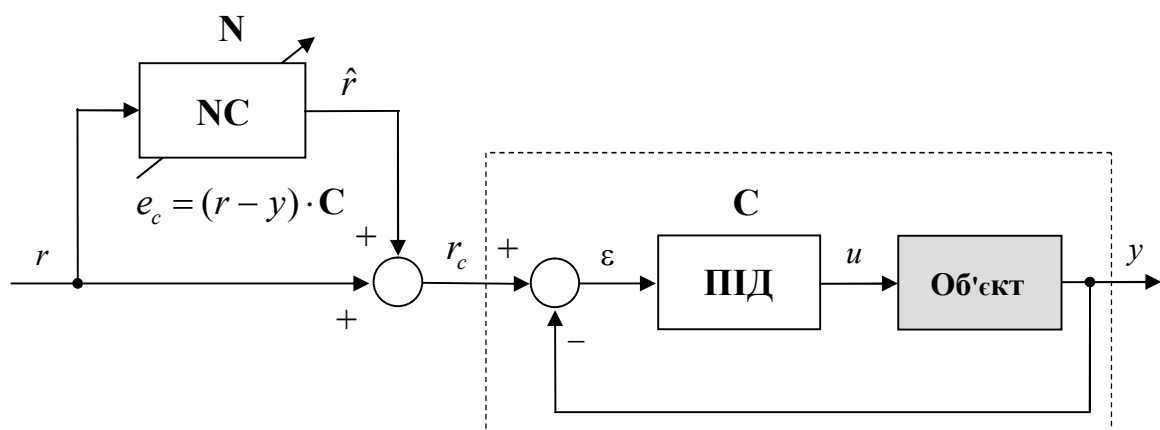


Рисунок 1.8 – Регулювання методом компенсації завдання

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

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

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.

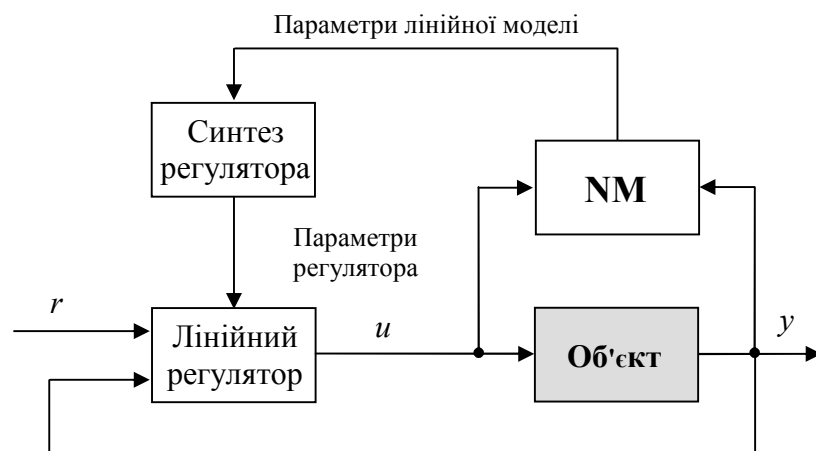


Рисунок 1.9 – Instantaneous linearization і синтез лінійного регулятора

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.

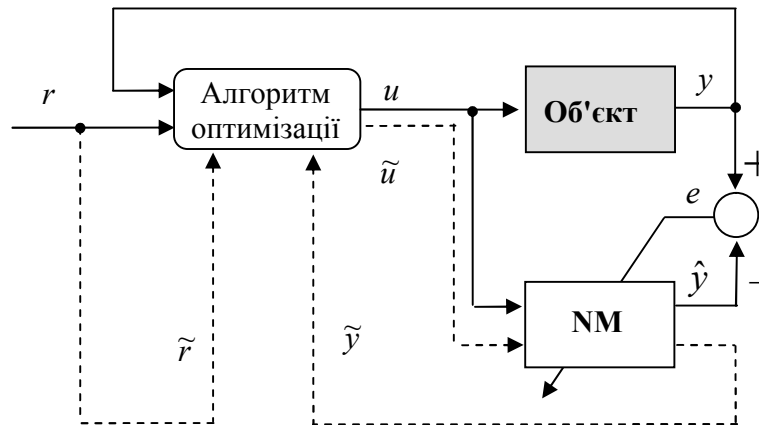


Рисунок 1.10 – Предиктивне регулювання

1.3 Вимоги до електроустаткування мостових кранів

Впровадження високого рівня механізації та автоматизації виробничих процесів промислових підприємств пов'язане не лише з основними технологічними операціями, але й з виконанням різноманітних допоміжних операцій, таких як транспортування сировини, готової продукції та палива. Ці завдання у багатьох випадках виконуються за допомогою електричних кранів.

Різні конструкції електричних кранів зустрічаються майже у всіх галузях промисловості. Наприклад, мостові крани використовуються у цехах металургійних та машинобудівних заводів, порталні і козлові перевантажувальні крани – на рудних дворах заводів і вугільних складах електростанцій, а також баштові, кабельні та інші види кранів використовуються у будівельній сфері.

Окрім стаціонарних кранів, що працюють в межах певної території, велика питома вага, особливо в міському і транспортному господарстві, мають пе-

ресувні крани: автомобільні, гусеничні, залізничні плавучі. Кожна з вказаних груп кранів включає значну кількість конструктивних модифікацій. Наприклад, мостові крани бувають крюки, грейфери, посадочні, кліщові і т.д. Специфіка цеху або підприємства і їх технологія визначають в основному конструктивний вид встановленого крана.

На промислових підприємствах найбільш широко використовується таунтовий кран, який є універсальним підйомно-транспортним пристроєм. Мостовий кран складається зі сталевих конструкцій моста, що опирається на ходові візки та здатний переміщуватись по підкранових рейках. Ці рейки закріплені над обслуговуваною площею на стаціонарних опорах. Вздовж моста крана розташовані рейки, по яких рухається візок, оснащений механізмом пересування та підйомною лебідкою для підйому та опускання вантажів. Основними механізмами мостового крана є: механізм переміщення моста, механізм переміщення візка та підйомна лебідка, всі ці механізми оснащені власним електроприводом.

Електричне устаткування кранів повинне забезпечити надійну роботу повторно – короткочасному режимі і великій частоті включень, в умовах запиленої приміщень, високій вологості повітря, різких змінах температури. В той же час до електроустаткування пред'являються жорсткі вимоги в сенсі безперебійності в роботі, високій продуктивності, безпеці обслуговування і простоти експлуатації.

1.4 Обґрунтування вибору системи електроприводу механізму підйому мостового крана

Основними даними для вибору електроприводів кранів є вантажопідйомність і швидкість руху, діапазон регулювання швидкості і необхідна жорсткість характеристики, конструктивні дані і маса механічного устаткування, число включень в годину, тривалість включення, умови навколишнього середовища, експлуатаційні витрати.

Економічна оцінка систем управління має ґрунтуватися на принципі мінімізації витрат, що включає первісні витрати за таблицею 2.2, витрати на експлуатаційний ремонт, а також витрати на електроенергію, споживану з мережі під час пуску та гальмування кранів протягом періоду експлуатації до капітального ремонту (10 років). Обирається система, яка демонструє найкращі економічні характеристики.

Таблиця 2.2 – Порівняльні економічні дані системи електроприводів
(вартість електроустаткування, віднесена до 1 кВт потужності двигуна)

Найменування	Інтервал потужностей електроприводу, кВт						
	0,7 -2	2 -5	5-12	12 -30	30 -70	70 -150	Понад 150
Електродвигун:							
постійного струму	-	120	72	38	28	26	25
з фазним ротором	-	50	28	17	15	13	11
двух-, трьохшвидкісний короткозамкнений	70	60	40	28	25	-	-
Комплект магнітного пускача	10	6	5	-	-	-	-
Апаратура							
силового кулачкового контроллера	20	10	5,5	3	3	-	-
магнітного контроллера	100	75	40	16	9	7,5	6,5
живлення і управління системи Г – Д	-	120	70	42	32	22	20
живлення і управління системи ТП – ДП	-	250	200	80	50	32	30
живлення і управління системи НПЧ - АДД	-	-	320	150	65	-	-
управління системи							
ТПН – АДФ	-	-	200	80	45	32	-
МКИ – АДФ	-	80	55	28	-	-	-

У разі, якщо економічні показники порівнюваних систем є схожими, проводиться додаткова оцінка за массогабаритними характеристиками та умовами розташування електрообладнання. Орієнтовні дані про масу різних систем управління наведені в таблиці 2.3. Оскільки розташування перетворювальних агрегатів або напівпровідникових перетворювачів на кранах пов'язане з певними труднощами, може виникнути потреба у використанні системи, що не має оптимальних економічних характеристик.

Таблиця 2.3 - Порівняльні показники маси електроприводів

Система електроприводу	Маса електроприводу, кг/кВт, в інтервалі потужностей, кВт						
	0,7 -2	2 -5	5 -12	12 -30	30-70	70-150	Понад 150
МП – АДК	18	12	10	-	-	-	-
До – АДФ	-	30	20	16	-	-	-
МКД –АДФ	-	58	35	21	24	18	16
МК – АДД	60	52	42	30	-	-	-
Г – Д	-	-	100	60	52	52	52
ТП – ДП	-	-	80	52	36	42	37
НПЧ – АДД	-	-	65	42	35	-	-
ТПН – АДФ	-	-	45	26	28	20	-
МК – ДП	-	70	48	38	37	36	35
МКИ – АДФ	80	65	40	27	-	-	-

Тиристорні електроприводи постійного струму (система ТП-Д) застосовуються для кранівських механізмів, коли необхідне забезпечення високоякісного регулювання при потужності приводу понад 60 кВт. За показниками регулювання системи ТП-Д подібні до системи Г-Д. При цьому тиристорний електропривод має кілька переваг перед системою Г-Д і фактично витіснив електроприводи за цією системою на кранах. На сьогодні тиристорні електроприводи постійного струму встановлюються на потужних мостових ливарних кранах, перегружателях, бетоноукладачах, високопродуктивних баштових кранах для

висотного будівництва та інших унікальних кранівських комплексах. Потужність електроприводів таких комплексів може досягати 400-600 кВт.

Тиристорні електроприводи, незважаючи на деякі відмінності в конструкції окремих вузлів автоматики і захисту, розробляються на основі стандартних типових схем, які залежать від використання відповідного тиристорного перетворювача (ТП). До них відносяться схеми з реверсивними ТП серій АТРК і ТПЕ, а також з нереверсивними ТП серії АТК і контакторним реверсом. Використання нереверсивних ТП доцільне для приводів потужністю до 100 кВт і з невеликою кількістю включень при режимі роботи не більше 4М. Для більш напружених режимів роботи і потужностей понад 100 кВт слід використовувати схеми з реверсивними ТП.

Залежно від призначення і конструкції кранівських механізмів, тиристорні електроприводи можуть бути одно- і багатодвигунними. Зазвичай багатодвигунні електроприводи використовуються для механізмів пересування кранів. Для механізмів підйому багатодвигунні електроприводи (зазвичай двохдвигунні) застосовуються при високій потужності системи. При цьому електричні схеми одно- і багатодвигунних систем не мають суттєвих відмінностей.

На основі техніко-економічного аналізу приводів кранівських механізмів обирається електропривід за схемою тиристорного перетворювача – двигун.

1.5 Постановка задачі дослідження

Синтез і дослідження динамічних характеристик системи електроприводу механізму переміщення моста зазвичай виконується в припущенні нескінченної жорсткості всіх зв'язків між елементами механічної частини системи. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер руху елементів тільки в середньому, оскільки не враховує вплив пружності реальних механічних зв'язків. Внаслідок кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, прикладення до якої управляючої дії (моменту двигуна), або збурюючої дії (моменту

статичного навантаження) викликає коливання зв'язаних мас, які збільшують максимальні навантаження зв'язків і ускладнюють точність відробітку необхідних рухів і переміщень. Загальновживані регулятори не в змозі забезпечити задовільні показники якості функціонування системи.

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

У науковій літературі, як у вітчизняній, так і за кордоном, ретельно досліджуються теоретичні аспекти і методи створення систем управління складними багатовимірними динамічними об'єктами на основі багатошарових нейронних мереж. У працях, присвячених цим мережам, велика увага приділяється основним характеристикам нелінійних багатошарових нейронних мереж та алгоритмам їх навчання в реальному часі, а також приводяться приклади комп'ютерного моделювання систем управління, заснованих на таких нейромережах, для складних нелінійних об'єктів з багатими взаємозв'язками.

В даний час отримано багато прикладів успішно працюючих нейромережових систем управління. У літературі описані численні приклади практичного застосування нейронних мереж для вирішення задач керування конкретними об'єктами: літаком [42-44], вертольотом [45], гірничозбагачувальним процесом [46], автомобілем-роботом [47], гібридним двигуном автомобіля [48], пневмоциліндром [49], об'єктом спеціального призначення [50], моделлю перевернутого маятника [51], електропідчю [52], турбогенератором [53], зварювальним

апаратом [54] і інших. Ці роботи наочно ілюструють ефективність застосування нейромережових методів управління нелінійними динамічними об'єктами в складних умовах функціонування.

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

Метою роботи є розробка нейромережової системи управління механізмом переміщення моста мостового крана з пружності механічних зв'язків.

Для досягнення поставленої мети необхідно вирішити наступні завдання. задач.

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

2. Розробити математичну моделі системи управління з урахуванням кінцевої жорсткості зв'язку між двигуном і мостом, провести моделювання багатомасової системи на ЕОМ і виконати аналіз отриманих результатів.

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

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

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

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

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Дослідження щодо використання методів нейронних мереж для управління електромеханічними системами показали, що штучні нейронні мережі є перспективною альтернативою традиційним методам побудови систем керування нелінійними об'єктами. Основною перевагою штучних нейронних мереж є їх здатність до навчання, що дає змогу знаходити більш ефективні рішення для складних завдань управління. Крім того, завдяки наявності в структурі нейронних мереж нейронів з нелінійними функціями активації, їх можна використовувати для розв'язання задач управління нелінійними системами, тоді як традиційні методи не завжди справляються з такими завданнями.

2. Показана ефективність використання мережі типу багатошаровий персептрон для побудови нейромережевої моделі об'єкта управління. Багатошаровий персептрон є універсальним апроксиматором-класифікатором і належить в сьогоденній практиці до найбільш використовуваних нейронних мереж. Розглянуті питання навчання нейронних мереж, вибрано метод навчання нейронної мережі регулятора.

3. Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання динамічних систем є отримання їх адекватних математичних моделей. Показано, що завдання ідентифікації нелінійних динамічних об'єктів успішно вирішуються за допомогою штучних нейронних мереж. Найбільше застосування для ідентифікації отримали багатошаровий персептрон і радіально-базисні мережі.

4. Для оптимізації функціонування об'єкта необхідно коригувати параметри регуляторів у відповідності з змінами робочих умов. Тому розробка систем управління на основі адаптивного підходу, комбінуючи методи теорії ШНМ, є доцільною, оскільки це дозволяє створювати прості нейромережеві моделі та адаптивно налаштовувати їх параметри разом із параметрами регулятора відповідно до змінюваних умов.

5. Після аналізу сучасних структур систем нейрорегулювання обрано принцип нелінійного предиктивного регулювання для створення нейромережевого регулятора системи.

6. Аналіз основних характеристик мостових кранів із урахуванням пружності механічних зв'язків підтверджує доцільність застосування нейромережевих технологій для розробки нейромережевої системи управління механізмом переміщення моста мостового крана.

7. Сформульована задача дослідження полягає у розробці нейромережевої системи управління механізмом переміщення моста мостового крана з урахуванням податливості механічних зв'язків.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ МОСТА МОСТОВОГО КРАНА

2.1 Технічні характеристики мостового крана. Вимоги до системи управління

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

Таблиця 2.1 – Технічні дані мостового крана

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

Вимоги до показників якості функціонування системи управління.

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

2.2 Розрахунок потужності електроприводу механізму переміщення моста. Вибір приводних двигунів

Енергетика електроприводів кранів як система повторно-короткочасного режиму роботи характеризується двома основними особливостями:

- 1) необхідністю урахування втрат пуско-гальмівних режимів, які складають значну частину в загальному балансі втрат;
- 2) підвищеними в порівнянні з електроприводами тривалого режиму роботи постійними втратами, визначуваними необхідністю забезпечення високих динамічних моментів.

У [55] розроблена методика вибору двигунів кранів, що враховує дані особливості роботи електроприводів кранів.

У основу методики покладено поняття розрахункової потужності, на яку повинен бути вибраний двигун крана за умовами забезпечення пускового режиму.

Для механізму пересування моста розрахункова потужність визначається по формулі:

$$P_p = \frac{0,66(G + Q_{\Pi})\alpha_M V_M}{10^3 \eta_M} + \frac{P_{ст}}{1,75}, \quad (2.1)$$

де G - маса переміщуваного механізму, тобто сумарна маса моста і візка

$$G = G_M + G_B, \quad (2.2)$$

$$G = (28,5 + 8,5) \cdot 10^3 = 37 \cdot 10^3 \text{ кг}$$

Q_{Π} - маса вантажу з підвіскою;

$$Q_{\Pi} = Q + q, \quad (2.3)$$

$$Q_{\Pi} = 41 \cdot 10^3 + 1 \cdot 10^3 = 42 \cdot 10^3 \text{ кг}$$

α_m - прискорення механізму, приймаємо $\alpha_m = 0,3$ згідно рекомендаціями [55];

η_m - ККД механізму, $\eta_m = 0,9$ [55];

$P_{ст}$ - потужність статичного навантаження сталого руху, визначається по виразу:

$$P_{ст} = \frac{(G + Q)V_m}{10^3 m_m \eta_m} \left(\frac{\varphi_{п} d_{ст} + 2\mu}{D_m} k_{рб} + \beta \right), \quad (2.4)$$

де $\varphi_{п}$ - коефіцієнт тертя в підшипниках маточин коліс, для підшипників кочення $\varphi_{п} = 0,015$;

D_m - діаметр ходового колеса;

$d_{ст}$ - діаметр маточин ходового колеса, зазвичай для розрахунків застосовується $d_{ст} / D_m = 0,25$;

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

$k_{рб}$ - коефіцієнт форми ходового колеса, що враховує тертя реборд ходового колеса $k_{рб} = 1,3 \div 1,4$; приймаємо $k_{рб} = 1,35$;

m_m - число механізмів пересування, для механізму пересування моста $m_m = 2$;

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

V_m - швидкість пересування моста

$$P_{ст} = \frac{9,8(37 \cdot 10^3 + 42 \cdot 10^3) \cdot 0,85}{10^3 \cdot 2 \cdot 0,86} \cdot \left(\frac{0,015 \cdot 0,14 + 2 \cdot 0,5 \cdot 10^{-3}}{0,56} \cdot 1,35 + 0,003 \right) = 4,7 \text{ кВт}$$

Розрахункова потужність

$$P_p = \frac{0,66(37 \cdot 10^3 + 42 \cdot 10^3) \cdot 0,25 \cdot 0,85}{10^3 \cdot 2 \cdot 0,86} + \frac{3,98}{1,75} = 8,63 \text{ кВт}$$

По таблиці 6,5 [55] визначається тепловий коефіцієнт $k_T = 1,3$.

Визначаємо розрахункову потужність двигуна по тепловому режиму

$$P_{\text{ном.т}} = P_{\text{ст}} / k_{\text{т}} \quad (2.5)$$

$$P_{\text{ном.т}} = 4,7 / 0,7 = 6,7 \text{ кВт}$$

Оскільки $P_{\text{р}} > P_{\text{ном.т}}$, то попередній вибір двигуна здійснюємо по $P_{\text{р}}$. Вибираємо два двигуни Д32. Номінальна потужність кожного двигуна при ПВ=40% становить $P_{\text{н}} = 9,5 \text{ кВт}$, номінальна напруга $U_{\text{н}} = 220 \text{ В}$, номінальний струм $I_{\text{н}} = 57 \text{ А}$, номінальна частота обертання $n_{\text{н}} = 800 \text{ об/хв}$, момент інерції $J_{\text{дв}} = 0,425 \text{ кг} \cdot \text{м}^2$, $N = 558$, $2a = 2$, $2p = 4$, $R_{\text{я}} + R_{\text{дп}} = 0,28 \text{ Ом}$, $\Phi_{\text{н}} = 0,0135 \text{ Вб}$, $\eta = 0,77$.

2.3 Перевірка двигунів по тепловому режиму і запасу зчеплення при пуску і гальмуванні

Визначаємо сумарний момент інерції, приведений до валу двигуна

$$J_{\Sigma} = 2 \cdot 1,2 J_{\text{дв}} + \frac{91 \cdot (G + Q_{\text{п}}) \cdot V_{\text{м}}^2}{m_{\text{м}} \cdot n_{\text{н}}^2 \cdot \eta_{\text{м}}} \quad (2.6)$$

$$J_{\Sigma} = 2 \cdot 1,2 \cdot 0,425 + \frac{91 \cdot (37 \cdot 10^3 + 42 \cdot 10^3) \cdot 0,85^2}{2 \cdot 800^2} = 1,02 + 4,06 = 5,08 \text{ кг} \cdot \text{м}^2$$

Для перевірки двигуна по тепловому режиму визначаємо еквівалентний ККД електроприводу

$$\eta_{\text{екв}} = \frac{\eta_{\text{екв.б}}}{1 + \frac{(\eta_{\text{екв.б}} - \eta_{\text{екв.Н}})}{\eta_{\text{екв.Н}}} \cdot \frac{J_{\Sigma}}{1,2 \cdot J_{\text{дв}}} \cdot \left(\frac{n_{\text{max}}}{1000} \right)^2}, \quad (2.7)$$

де $\eta_{\text{екв.б}}$ - ККД при числі включень $N_{\text{в}} = 0$ $\eta_{\text{екв.б}} = 0,94$ [55];

$\eta_{\text{екв.Н}}$ - ККД при числі включень $N_{\text{в}} = 120$ $\eta_{\text{екв.Н}} = 0,92$ [55];

$n_{\max}$ - максимальна частота обертання; приймаємо $n_{\max} = n_{\text{н}}$

$$\eta_{\text{екв}} = \frac{0,94}{1 + \frac{0,94 - 0,92}{0,92} \cdot \frac{5,08}{1,2 \cdot 2} \left(\frac{800}{1000} \right)^2} = 0,776.$$

Визначаємо розрахунковою потужність приводу по тепловому режиму

$$P_{\text{розр}} = \frac{k_3 k_{\text{екв}} k_{\text{н}} \sqrt{\varepsilon_p / \varepsilon_{\text{ном}}}}{k_0 k_p} \cdot \frac{\eta_{\text{екв.б}}}{\eta_{\text{экв.б}} - k_{\text{дн}} (\eta_{\text{екв.б}} - \eta_{\text{екв}})} \cdot P_{\text{ст}}, \quad (2.8)$$

де k_3 - коефіцієнт, який враховує запас потужності, що закладається при проектуванні приводу, для групи електроприводу 4М $k_3 = 1$ з табл. 6.4 [55];

$k_{\text{екв}}$ - коефіцієнт, що визначається з табл.6.4 [55], $k_{\text{екв}} = 0,71$;

ε_p - фактична тривалість включення, для групи 4М $\varepsilon_p = 0,4$;

$\varepsilon_{\text{ном}} = ПВ_{\text{ст}}$ - стандартна тривалість включення механізму;

$k_{\text{дн}}$ - динамічний коефіцієнт, для електроприводу тиристора постійного струму $k_{\text{дн}} = 1,25$ з табл. 6.5 [55]

Коефіцієнти визначаються по рис. 6.12 [55]: $k_0 = 1$, $k_p = 1$, $k_0 = 1$, $k_{\text{н}} = 1$.

$$P_{\text{ном.т}} = \frac{1 \cdot 0,71 \sqrt{\frac{0,4}{0,4}} \cdot 1}{1 \cdot 1} \cdot \frac{0,94}{0,94 - 1,25(0,94 - 0,776)} \cdot 3,98 = 7,26 \text{ кВт}.$$

Оскільки виконується умова $P_{\text{н}} > P_{\text{ном.т}}$, то двигун задовольняє тепловому режиму.

Електропривод механізму пересування необхідно перевіряти по запасу зчеплення при пуску і гальмуванні для найбільш сприятливих умов роботи.

Визначаємо забезпечуване приводом прискорення

$$a_{\text{дв}} = \frac{10^3 [k_{\text{п}} \gamma_{\text{п}} \eta_{\text{м}} P_{\text{п}} - P_{\text{ст}}]}{(G + Q_{\text{п}}) V_{\text{м}} \alpha'}, \quad (2.9)$$

де k_{π} , γ_{π} - відповідно коефіцієнти використання двигуна по пусковому моменту і кратність відношення максимального пускового моменту до номінального моменту двигуна ;

$$k_{\pi} = 0,9 \quad \gamma_{\pi} = 2 \text{ з табл. 6.2 [55]}$$

α' - коефіцієнт, рівний відношенню сумарного моменту інерції J_{Σ} до моменту інерції поступально рухомих мас $J_{\text{пост}}$

$$\alpha' = \frac{J_{\Sigma}}{J_{\text{пост}}} = \frac{J_{\Sigma}}{J_{\Sigma} - 1,2J_{\text{дв}}} , \quad (2.10)$$

$$\alpha' = \frac{5,08}{5,08 - 2 \cdot 1,2 \cdot 0,425} = 1,13 ,$$

$$\alpha_{\text{дв}} = \frac{10^3 [0,9 \cdot 2 \cdot 0,86 \cdot 8,84 - 3,98 \cdot 0,86]}{(37 \cdot 10^3 + 42 \cdot 10^3) \cdot 0,85 \cdot 1,13} = 0,17 .$$

Максимальне значення прискорення визначаємо згідно (6.51) [55].

$$\alpha_{\text{max}} = 1,4 \cdot a - 0,09 , \quad (2.11)$$

де a - прискорення, визначене по таблиці 1.4 [55] $a = 0,25$

$$\alpha_{\text{max}} = 1,4 \cdot 0,25 - 0,09 = 0,26 .$$

Прискорення $\alpha_{\text{дв}}$ менше допустимого α_{max} , тому двигун задовольняє умовам зчеплення і може бути використаний для електроприводу переміщення моста.

2.4 Вибір основного електроустаткування і розрахунок параметрів силового кола електроприводу

Відповідно до рекомендацій [55] для живлення якірного ланцюга послідовно включених двигунів і їх обмотки збудження вибираємо реверсивний ти-

ристорний перетворювач ТПЕ-200/200-460 з наступними технічними характеристиками: номінальна випрямлена напруга $U_{\text{дн}} = 460 \text{ В}$, номінальний випрямлений струм $I_{\text{дн}} = 100 \text{ А}$, номінальна випрямлена напруга збудження $U_{\text{дзн}} = 230 \text{ В}$, номінальний випрямлений струм збудження $I_{\text{дзн}} = 15 \text{ А}$.

Перетворювач отримує живлення від мережі 380 В, 50 Гц через токообмежуючий реактор. По каталогу вибираємо реактор типу РТСТ-8-0505У3. Технічні характеристики реактора наступні: номінальна лінійна напруга живлячої мережі 410 В, номінальна сила фазного струму 82 А, номінальна індуктивність фази $L_p = 0,505 \cdot 10^{-3} \text{ Гн}$, активний опір обмоток $R_p = 0,0317 \text{ Ом}$.

Вибір згладжуючого реактора виконуємо виходячи з умови обмеження допустимого для двигуна рівня пульсацій струму. Розрахунок виконуємо по основній гармоніці пульсацій з урахуванням того, що вплив вищих гармонійних значно менший. Допустиме значення пульсацій струму для двигунів кранів Д складає $i_{\pi} = 5 \div 10\%$ номінального струму двигунів. Необхідне значення індуктивності якірного ланцюга для згладжування пульсацій до заданого рівня визначається по наступній формулі

$$L_d = \frac{I_{\pi} E_{\text{до}}}{i_{\pi} \omega_1 I_{\pi}} , \quad (2.12)$$

де I_{π} - струм навантаження перетворювача (приймаємо рівним номінальному струму двигуна $I_{\pi} = I_{\text{н}} = 57 \text{ А}$;

ω_1 - кутова частота першої гармоніки випрямленої напруги

$$\omega_1 = 2\pi m f ,$$

m - число фаз перетворювача (для трифазної мостової схеми $m=6$)

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

E_{d0} - максимальна величина випрямленої ЕРС, що має місце при куті управління $\alpha = 0$

$$E_{d0} = \frac{m}{\pi} \sqrt{2} U_{\text{л}} \sin \frac{\pi}{m}, \quad (2.13)$$

m - число тактів випрямлення; для трифазної мостової схеми $m = 6$;

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

$$E_{d0} = \frac{6}{3,14} \sqrt{2} \cdot 380 \cdot \sin \frac{180^\circ}{6} = 514 \text{ В}. \quad (2.14)$$

$l_{\text{п}}$ - відносна величина пульсацій першої гармонічної випрямленої напруги. Величина $l_{\text{п}}$ може бути визначена з графіка $l_{\text{п}} = f(t)$ рис.5.3 [55], побудованого для мостової трифазної схеми випрямлення.

Значення $l_{\text{п}}$ вибирається для граничного кута регулювання α_{max} , який визначається з виразу для швидкісної характеристики електроприводу

$$\omega = \frac{E_{d0} \cdot \cos \alpha - \Delta U_{\text{в}} - R_{\Sigma} I_{\text{д}}}{c}. \quad (2.15)$$

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

$$\omega_{\text{н}} = \frac{\pi n_{\text{н}}}{30}, \quad (2.16)$$

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

$$\omega_{\text{min}} = 0,1 \cdot 83,7 = 8,37 \text{ с}^{-1}.$$

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

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

R_{Σ} - сумарний активний опір ланцюгу електроприводу (визначимо R_{Σ} без урахування активного опору згладжуючого реактора)

$$R_{\Sigma} = 2 \cdot R_{\text{як}} + R_{\text{епр}}, \quad (2.17)$$

$R_{\text{як}}$ - активний опір якірного ланцюга двигуна:

$$R_{\text{як}} = k_t \cdot (R_{\text{я}} + R_{\text{дп}} + R_{\text{ко}}), \quad (2.18)$$

де k_t - температурний коефіцієнт; для двигунів серії Д, що мають клас ізоляції H, значення $k_t = 1,36$;

$R_{\text{я}}$, $R_{\text{дп}}$, $R_{\text{ко}}$ - активні опори обмотки якоря, додаткових полюсів і компенсаційної обмотки відповідно. Для двигуна Д32 $R_{\text{я}} + R_{\text{дп}} = 0,28 \text{ Ом}$, $R_{\text{ко}} = 0$.

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

Опір двох двигунів:

$$R_{\text{як}\Sigma} = 2R_{\text{як}} = 2 \cdot 0,1477 = 0,2954 \text{ Ом}.$$

Еквівалентний опір тиристорного перетворювача:

$$R_{\text{епр}} = R_{\text{п}} + R_{\text{е}}, \quad (2.19)$$

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

$$R_{\text{е}} = m \cdot f \cdot L_{\text{п}},$$

$$R_{\text{е}} = 6 \cdot 50 \cdot 0,505 \cdot 10^{-3} = 0,0325 \text{ Ом},$$

$$R_{\text{епр}} = 31,7 \cdot 10^{-3} + 0,0325 = 0,064 \text{ Ом},$$

Сумарний активний опір якірного ланцюга електроприводу

$$R_{\Sigma} = 2 \cdot 0,38 + 0,064 = 0,824 \text{ Ом}.$$

c - постійна проти ЕРС двигуна

$$c = \frac{pN}{2\pi a} \Phi_{\text{н}}, \quad (2.20)$$

p - число пар головних полюсів ($p=2$);

a - число пар паралельних гілок обмотки якоря ($a=1$);

N - число активних провідників обмотки якоря ($N=558$);

$\Phi_{\text{н}}$ - магнітний потік на полюс ($\Phi_{\text{н}} = 0,0135 \text{ Вб}$).

$$c = \frac{2 \cdot 558}{2 \cdot 3,14 \cdot 1} \cdot 0,0135 \cdot 10^{-2} = 2,37.$$

З виразу (2.15) визначаємо α_{max}

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

$$\alpha_{\text{max}} = \arccos \left(\frac{8,37 \cdot 2,37 + 1,5 + 0,824 \cdot 57}{514} \right) = 77,6^{\circ}.$$

Величина $l_{\text{п}}$, відповідна $\alpha_{\text{max}} = 77,6^{\circ} = 77,6^{\circ}$ дорівнює 0,23.

Необхідна величина індуктивності рівна:

$$L_{\text{д}} = \frac{0,23 \cdot 514}{0,08 \cdot 1884 \cdot 57} = 0,01376 \text{ Гн}.$$

Індуктивність якірного кола електродвигуна:

$$L_{\text{як}} = k \cdot \frac{U_{\text{н}}}{I_{\text{н}}} \cdot \frac{I}{\omega_{\text{н}} p}, \quad (2.21)$$

де k - коефіцієнт, рівний 0,6 для некомпенсованих двигунів;

p - число пар полюсів; двигуни серії Д є чотирьохполюсними, тобто $p = 2$;

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

Сумарна індуктивність якірного ланцюга електропроводу:

$$L_{\Sigma} = 2 \cdot L_{\text{як}} + 2L_{\text{р}}, \quad (2.22)$$

$$L_{\Sigma} = 2 \cdot 0,0138 + 2 \cdot 0,505 \cdot 10^{-3} = 0,0287 \text{ Гн}.$$

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

Електромагнітна постійна часу якірного ланцюга двигуна:

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

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

Електромагнітна постійна часу силового ланцюга електроприводу:

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

$$T_e = \frac{0,0287}{0,824} = 0,035 \text{ с}.$$

Електромеханічна постійна часу електроприводу:

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

$$T_m = 5,08 \cdot \frac{0,824}{(2 \cdot 2,37)^2} = 0,186 \text{ с},$$

Коефіцієнт посилення двигуна:

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

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

Коефіцієнт посилення тиристорного перетворювача визначуваний для синусоїдальної напруги порівняння.

$$k_{тп} = \frac{E_{d0}}{U_m}, \quad (2.27)$$

U_m - амплітудне значення напруги порівняння, приймаємо $U_m = 24 \text{ В}$

$$k_{тп} = \frac{514}{24} = 21,4.$$

2.5 Вибір структури системи автоматичного управління

Характер перехідного процесу залежить від співвідношення постійних часу елементів системи. Тому для формування якісних перехідних процесів необхідно забезпечити відповідні значення цих співвідношень, використовуючи корегуючі регулятори. Загальний підхід до побудови автоматичних систем регулювання, незалежно від типу електроприводу, включає створення структурних схем на основі виразів для окремих ланок і введення відповідних регуляторів у систему. У практиці систем електроприводів найбільше використовуються системи підлеглого регулювання параметрів. У таких системах реалізується принцип компенсації основних інерцій об'єкта управління шляхом

послідовного підключення регулятора. Кількість регуляторів у системі відповідає кількості регульованих параметрів, а кожен контур системи складається з двох ланок: об'єкта регулювання та регулятора.

Відповідно до рекомендацій [55], вибирається двоконтурна однократно інтегруюча САР з внутрішнім контуром струму і зовнішнім контуром, замкнутим за ЕРС двигуна. Для спрощення системи датчик ЕРС не використовується, а підсумовування сигналів, пропорційних напрузі і падінню напруги в якірному ланцюзі, відбувається на вході регулятора ЕРС. Системи регулювання швидкості із зворотним зв'язком по ЕРС застосовуються для електроприводів, коли вимоги до точності регулювання швидкості не є високими. У загальнопромислових електроприводах електроприводи цього класу є переважаючими за кількістю.

2.6 Синтез послідовного коректуючого пристрою контура струму

Згідно рекомендаціям [55], для контура регулювання напруги ТП вживаний П – регулятор, а для контура струму – ПІ – регулятор; відповідно контур струму оптимізуємо по модульному критерію.

Спрощена принципова схема контура струму з внутрішнім контуром напруги показана на рис. 2.1. На рисунку прийняті наступні позначення: М – двигун постійного струму; ДН, ДТ – датчики напруги і струму відповідно; ТП – перетворювач тиристора; R_d – дільник напруги.

Відповідна алгоритмічна схема контура струму приведена на рисунку 2.2, де $W_{pn}(p)$ і $W_{pt}(p)$ - передавальні функції регулятора напруги і струму відповідно.

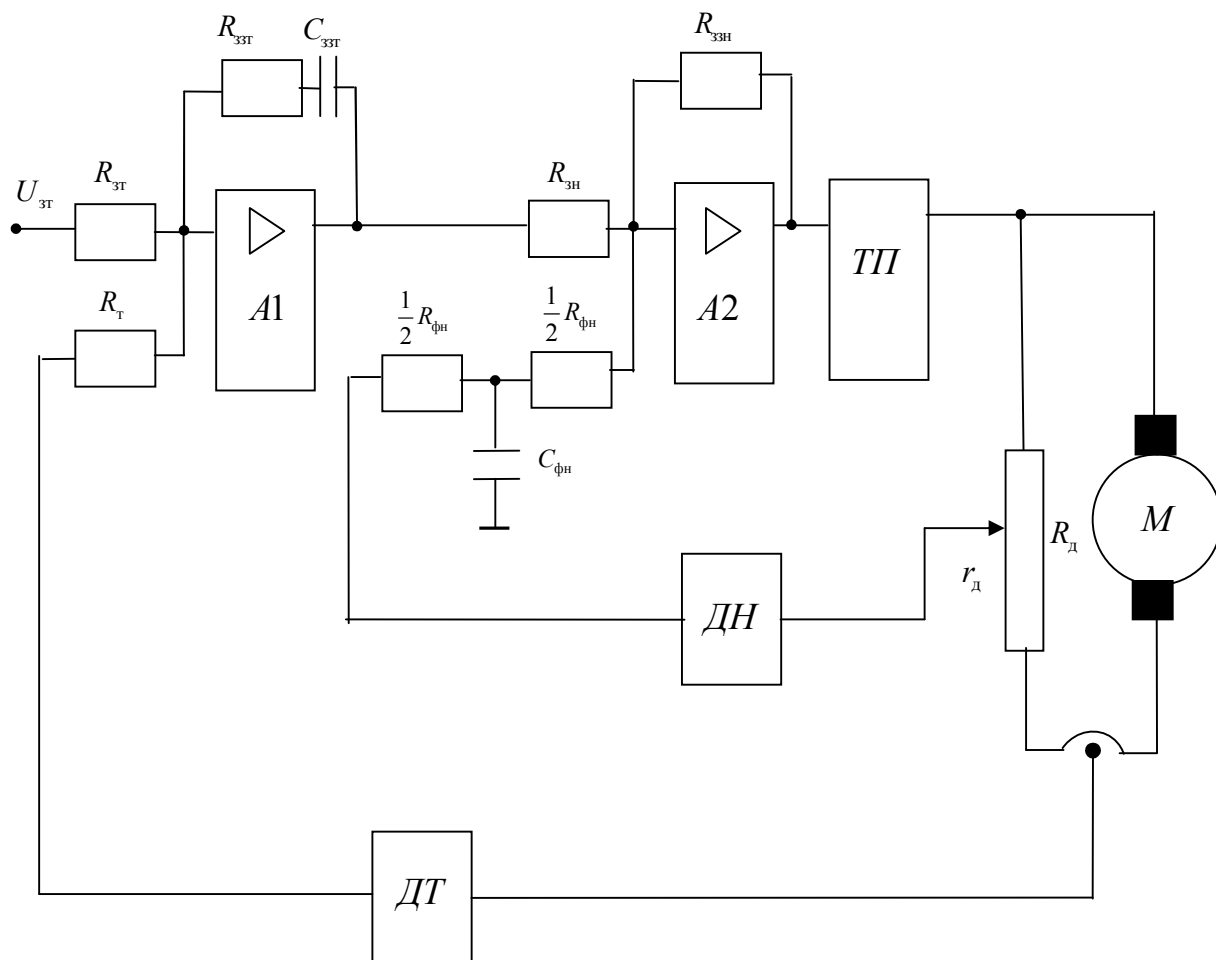


Рисунок 2.1 – Спрощена принципова схема контура струму

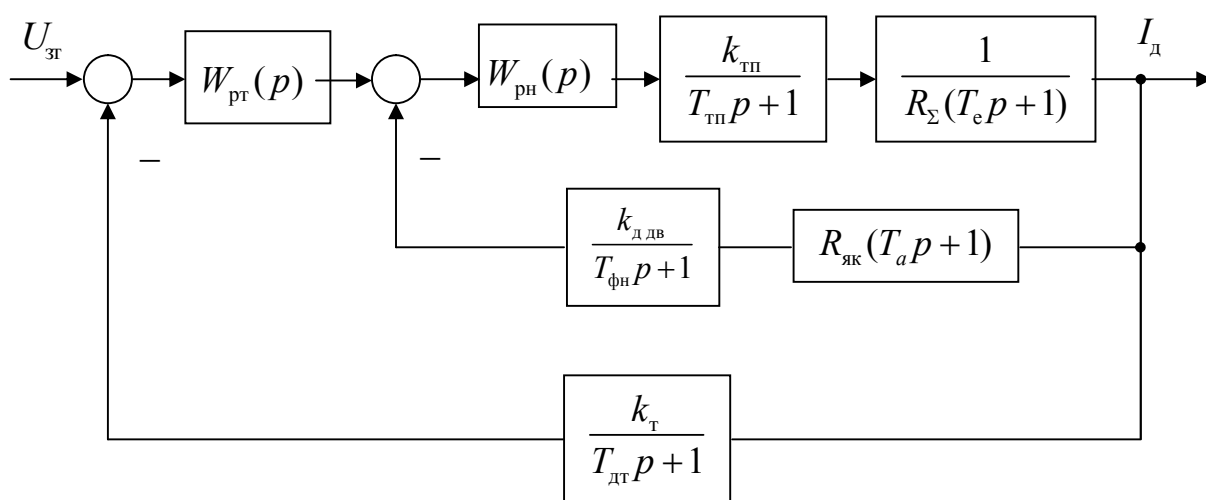


Рисунок 2.2 – Алгоритмічна схема контура струму

Визначаємо величину коефіцієнта дільника напруги двигуна виходячи з умови отримання 24 В на вході датчика напруги при максимальній напрузі двигуна.

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

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

Критичний коефіцієнт посилення контура напруги визначається відповідно до методики [55]:

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

де $T_{\text{т}} = 2T_{\text{мт}}$;

$T_{\text{мт}}$ - мала постійна, що не компенсується, контура струму, яка складається з постійної часу датчика струму з фільтром на вході і постійної часу тиристорного перетворювача. Оскільки таке визначення $T_{\text{мт}}$ є досить складним завданням, то величиною $T_{\text{мт}}$ у практичних розрахунках зазвичай задаються, виходячи зі встановлених практикою меж $4 \div 10$ мс. Приймаємо $T_{\text{мт}} = 0,005$ с; тоді $T_{\text{т}} = 2 \cdot 0,005 = 0,01$ с

T_1 - період першою гармонікою випрямленої напруги

$$T_1 = \frac{2\pi}{\omega_1} = \frac{2\pi}{2\pi \cdot m \cdot f} = \frac{1}{m \cdot f}, \quad (2.30)$$

$$T_1 = \frac{1}{6 \cdot 50} = \frac{1}{300} \text{ с}.$$

Граничний коефіцієнт посилення контура напруги:

$$k_{\text{н кр}} = \frac{3,14^2 \cdot 0,035 \cdot 0,01}{2 \cdot \left(\frac{1}{300}\right)^2 \cdot \left(1 - \frac{0,01}{0,035}\right)} = 96.$$

Приймаємо згідно рекомендаціям [55] $k_{\text{н}} = 40$.

Сумарний коефіцієнт посилення контура напруги рівний:

$$k_{\text{н}} = k_{\text{рп}} k_{\text{тп}} k_{\text{д дв}} \frac{R_{\text{як}}}{R_{\Sigma}}. \quad (2.31)$$

З даного співвідношення визначаємо величину коефіцієнта посилення регулятора контура напруги (П - регулятора).

$$W_{\text{рн}}(p) = k_{\text{рн}} = \frac{R_{\text{ззн}}}{R_{\text{зн}}} = \frac{k_{\text{н}} R_{\text{я}\Sigma}}{k_{\text{тп}} k_{\text{д дв}} (2 \cdot R_{\text{як}})}, \quad (2.32)$$

$$W_{\text{рн}}(p) = k_{\text{рн}} = \frac{40 \cdot 0,894}{21,4 \cdot 0,1091 \cdot (2 \cdot 0,824)} = 9,3$$

Приймаємо: $R_{\text{зн}} = 40 \text{ кОм}$ тоді

$$R_{\text{ззн}} = R_{\text{зн}} \cdot k_{\text{рн}}, \quad (2.33)$$

$$R_{\text{ззн}} = 40 \cdot 9,3 = 371,8 \text{ кОм}$$

Постійну часу фільтру $T_{\text{фн}}$ в цепи зворотному зв'язку по напрузі приймаємо рівною T_a . Параметри фільтру визначаються із співвідношення:

$$\frac{R_{\text{фн}}}{4} \cdot C_{\text{фн}} = T_{\text{фн}} = T_a = 0,036 \text{ с} \quad (2.34)$$

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

$$C_{\text{фн}} = \frac{4T_a}{R_{\text{зн}}}, \quad (2.35)$$

$$C_{\text{фн}} = \frac{4 \cdot 0,036}{40 \cdot 10^3} = 3,6 \cdot 10^{-6} \Phi$$

Передавальна функція замкнутого контура напруги:

$$\Phi_{\text{н}}(p) = \frac{k_{\text{рн}} k_{\text{тп}}}{R_{\Sigma}(1 + k_{\text{н}})} \cdot \frac{1}{\left(\frac{T_{\text{е}}}{1 + k_{\text{н}}} p + 1 \right) \cdot (T_{\text{тп}} p + 1)} . \quad (2.36)$$

Алгоритмічна схема контура струму з урахуванням (2.36) має вид, показаний на рис. 2.3

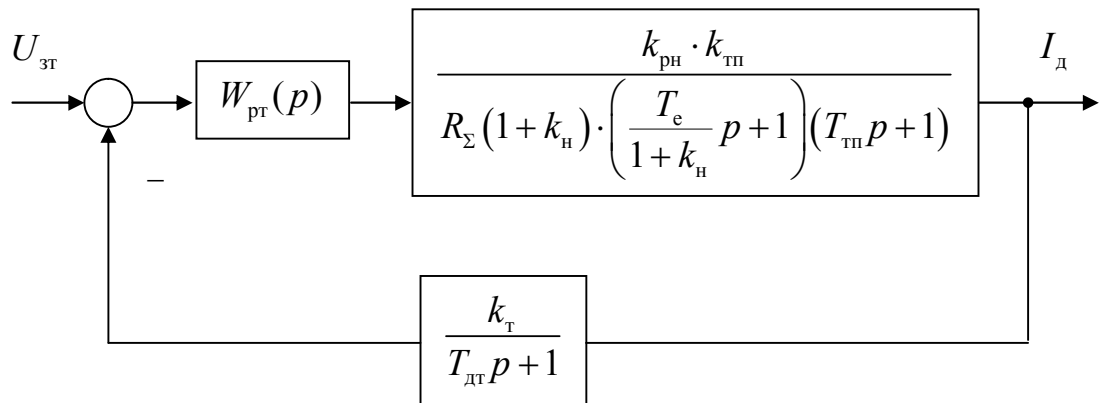


Рисунок 2.3 – Алгоритмічна схема контура струму з перетвореним контуром напруги

Для того, щоб можна було скористатися загальноприйнятою методикою синтезу послідовних коректуючи пристроїв, необхідно цю схему перетворити до алгоритмічної схеми з одиничним зворотним зв'язком (рис. 2.4). Форсуюча ланка $(T_{\text{дт}} p + 1)$ у блоці, що стоїть на вході замкнутого контура струму з одиничним зворотним зв'язком, згідно загальноприйнятій методиці, не враховується.

Передавальна функція розімкненого контура струму, оптимізованого по модульному критерію, повинна мати наступний вид:

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

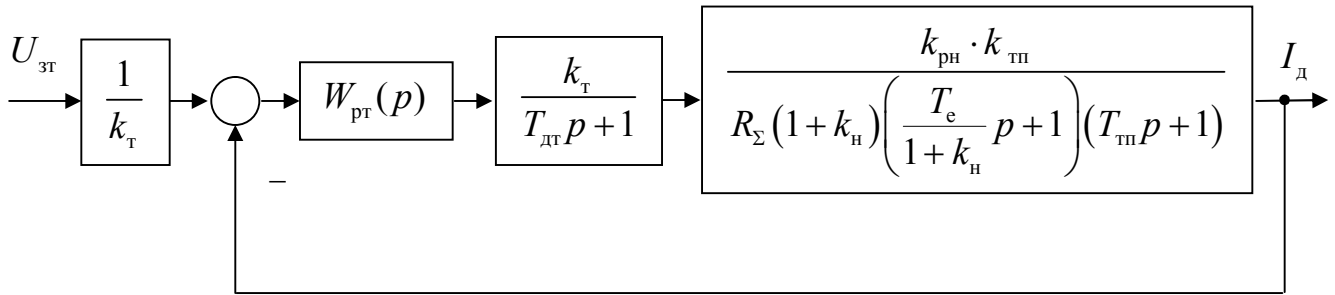


Рисунок 2.4 – Алгоритмічна схема контуру струму
з одиничним зворотним зв'язком

Для отримання такої передавальної функції необхідно включити в контур струму ПІ – регулятор з наступною передавальною функцією.

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

де

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

$$R_{\text{зт}} = \frac{2T_{\text{мт}} k_{\text{рн}} k_{\text{тп}} k_{\text{т}}}{R_{\Sigma} (1 + k_{\text{н}}) C_{\text{ззт}}} \quad (2.40)$$

У виразі (2.40) $k_{\text{т}}$ - коефіцієнт посилення ланцюга зворотного зв'язку по струму

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

де $k_{\text{ш}}$ - коефіцієнт посилення шунта.

$$k_{\text{ш}} = \frac{U_{\text{н ш}}}{I_{\text{н ш}}} \quad (2.42)$$

Приймаємо номінальний струм шунта $I_{\text{н ш}}$ рівним максимальному значенню струму робочого перевантаження двигуна, $I_{\text{н ш}} = I_{\text{max}} = 2,5 I_{\text{н}}$, падіння напруги на шунті при номінальному струмі $U_{\text{нш}} = 75 \cdot 10^{-3} \text{ В}$

$$k_{\text{ш}} = \frac{75 \cdot 10^{-3}}{2,5 \cdot 57} = 0,53 \cdot 10^{-3} \text{ В / А}$$

$k_{\text{дт}}$ - коефіцієнт посилення датчика струму.

Як датчик струму використовуємо операційний підсилювач з коефіцієнтом посилення 320. В цьому випадку при рівності опорів $R_{\text{зт}}$ і $R_{\text{т}}$ при струмі максимального робочого перевантаження, рівному $2,5I_{\text{н}}$ максимальна вихідна напруга датчика струму $U_{\text{дт max}}$ не перевищує 24 В. Таким чином:

$$k_{\text{т}} = 0,53 \cdot 10^{-3} \cdot 320 = 0,168 \text{ В / А}$$

Із співвідношення (2.39) визначаємо значення опору $R_{\text{зт}}$

$$R_{\text{зт}} = \frac{T_{\text{е}}}{(1 + k_{\text{н}}) \cdot C_{\text{зт}}} . \quad (2.43)$$

Приймаємо $C_{\text{зт}} = 5 \cdot 10^{-6} \text{ Ф}$, тоді

$$R_{\text{зт}} = \frac{0,035}{(1 + 40) \cdot 5 \cdot 10^{-6}} = 170 \cdot 10^3 \text{ Ом} .$$

Із співвідношення (2.40) визначаємо значення опору $R_{\text{зт}}$

$$R_{\text{зт}} = \frac{2T_{\text{ш}} k_{\text{рн}} k_{\text{тп}} k_{\text{т}}}{R_{\text{я}\Sigma} (1 + k_{\text{н}}) \cdot C_{\text{зт}}} . \quad (2.44)$$

$$R_{\text{зт}} = \frac{2 \cdot 0,005 \cdot 9,2 \cdot 21,4 \cdot 0,168}{0,824 \cdot (1 + 40) \cdot 5 \cdot 10^{-6}} = 1,9 \cdot 10^3 \text{ Ом}$$

$$R_{\text{т}} = R_{\text{зт}} = 1,9 \cdot 10^3 \text{ Ом} \quad (2.45)$$

Передавальна функція замкнутого контура струму, оптимізованого згідно модульного критерію:

$$W_{\text{зт}}(p) = \frac{1/k_{\text{т}}}{2T_{\text{мт}}^2 p^2 + 2T_{\text{мт}} p + 1} \quad (2.46)$$

2.7 Синтез послідовного коректуючого пристрою контура ЕРС

На рис.2.5 представлена принципова схема замкнутої системи автоматичного управління.

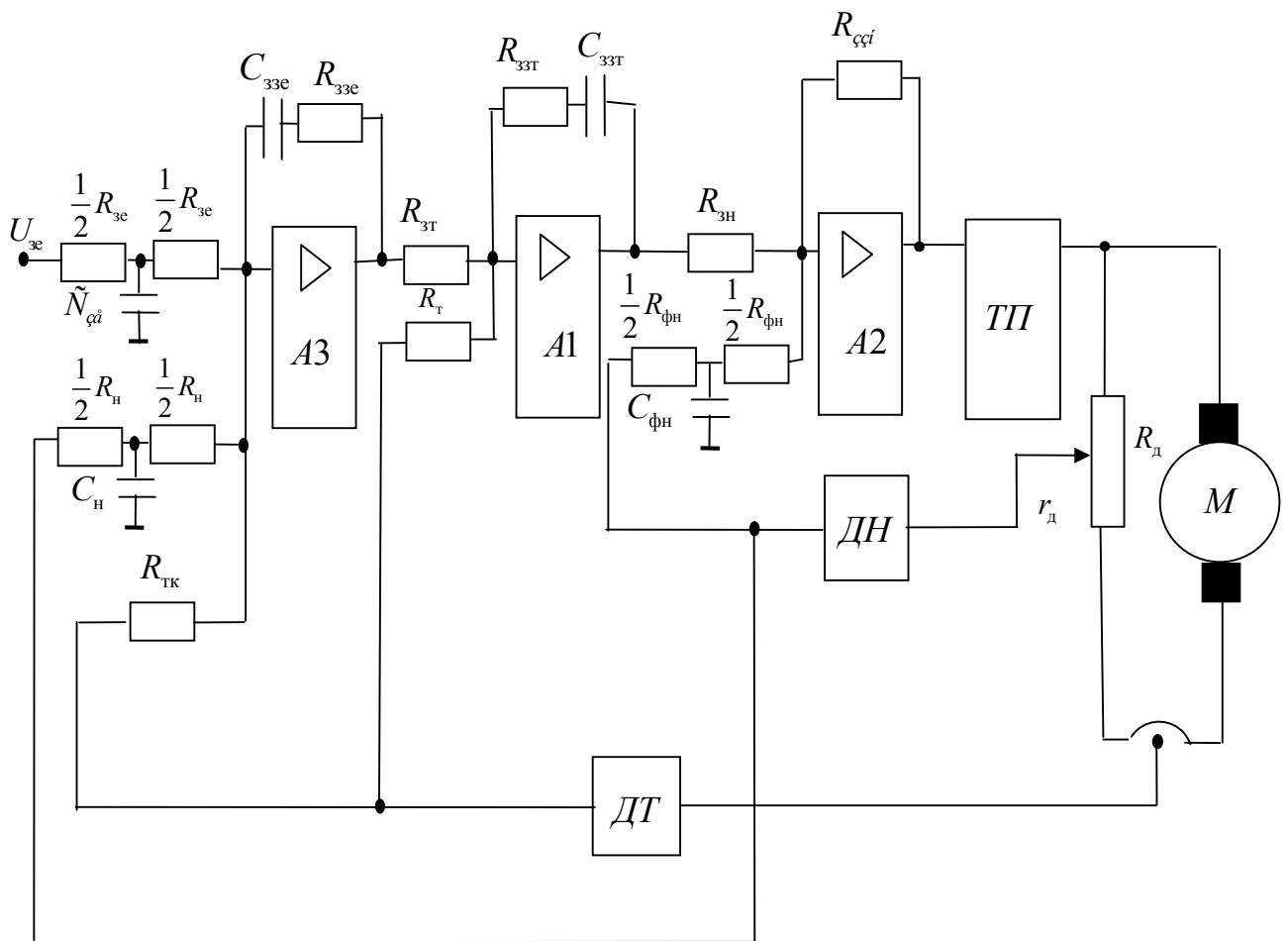


Рисунок 2.5 – Принципова схема системи управління із зворотним зв'язком по ЕРС

Оскільки електропривод мостового крана відноситься до загальнопромислових механізмів, в яких вимоги до точності регулювання швидкості відносно невисокі, то застосовуємо систему регулювання швидкості обертання із зворотним зв'язком по ЕРС двигуна. Вимірювання ЕРС проводиться шляхом підсумо-

ування сигналу, пропорційного напрузі двигуна з сигналом струмової компенсації, пропорційним падінню напруги в активному опорі ланцюга якоря двигуна. Застосовуємо систему, де підсумовування сигналів здійснюється безпосередньо на вході регулятора ЕРС. Гальванічне розділення ланцюга сигналу зворотного зв'язку по напрузі і силовий тут проводиться за допомогою датчика напруги, а для струмової компенсації використовується напруга датчика зворотного зв'язку контура струму.

На рис.2.6 представлена алгоритмічна схема системи.

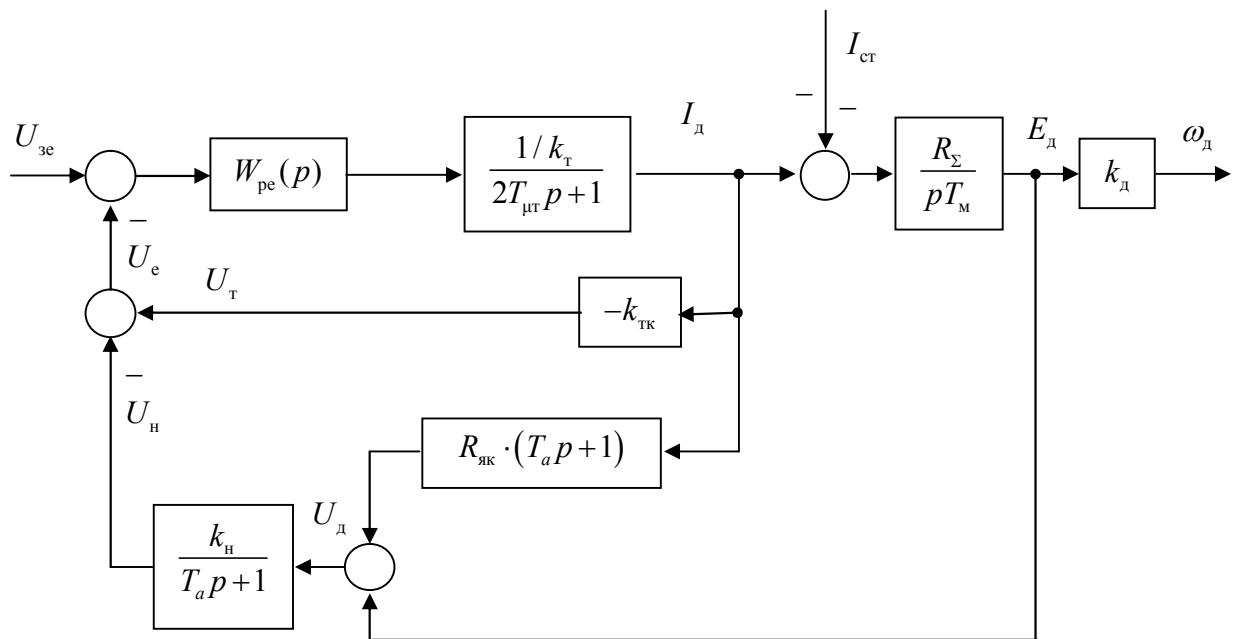


Рисунок 2.6 – Алгоритмічна схема системи регулювання швидкості із зворотним зв'язком по ЕРС

Оптимізований по модульному критерію контур струму представлений аперіодичною ланкою з передавальною функцією

$$W_{\tau}(p) = \frac{1/k_t}{2T_{\mu\tau}p + 1} \quad (2.47)$$

яка виходить при зневазі доданками $2T_{\mu\tau}^2 p^2$ у знаменнику передавальної функції контура струму (2.46).

Коефіцієнт посилення ланцюга струмової компенсації вибирається таким

чином:

$$k_{\text{тк}} = k_{\text{н}} R_{\text{як}} \quad (2.48)$$

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

Сигнал на вході датчика ЕРС рівний:

$$U_{\text{е}} = U_{\text{н}} - U_{\text{т}} = \frac{k_{\text{н}}}{T_{\text{а}}p + 1} E_{\text{д}} \quad (2.49)$$

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

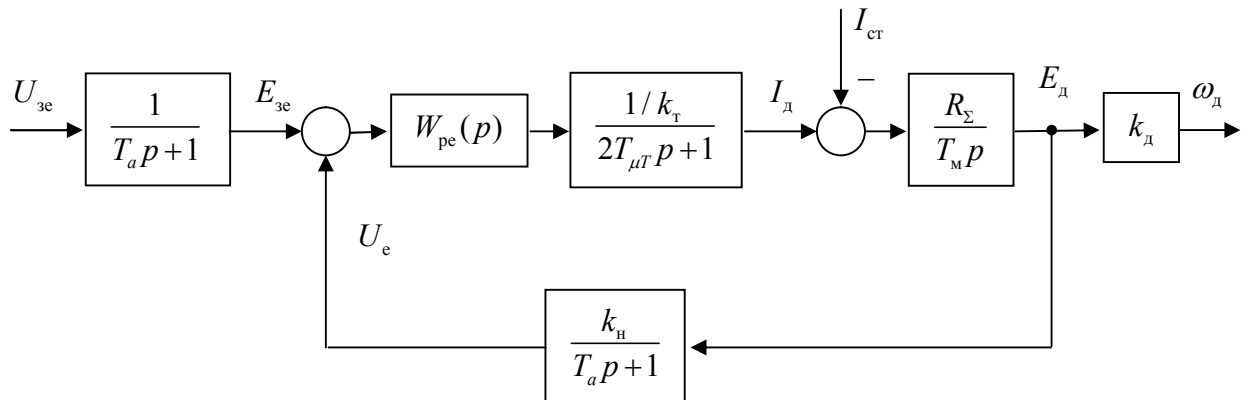


Рисунок 2.7 – Перетворена алгоритмічна схема системи

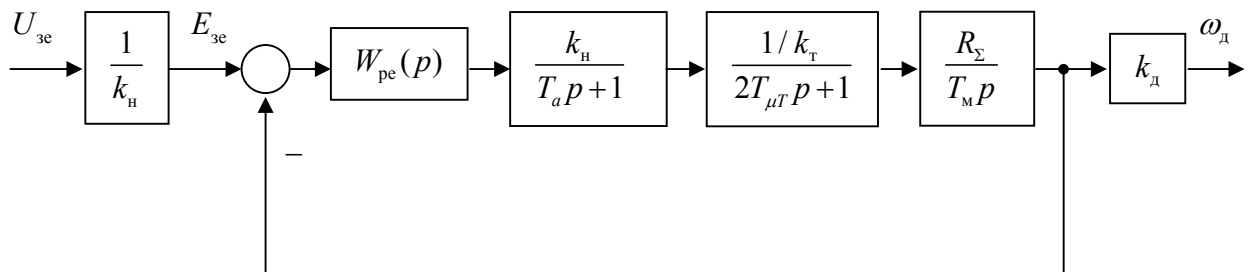


Рисунок 2.8 – Розрахункова алгоритмічна схема системи
з одиничним зворотним зв'язком

Для оптимізації контура ЕРС по симетричному критерію необхідно використовувати ПІ – регулятор з передавальною функцією.

$$W_{\text{pe}}(p) = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\text{мe}} k_{\text{н}} R_{\Sigma}} \cdot \frac{1 + 4T_{\text{мe}} p}{4T_{\text{мe}} p} \quad (2.50)$$

де $T_{\text{мe}}$ - мала постійна, що не компенсується, контура ЕРС

$$T_{\text{мe}} = 2T_{\text{мт}} + T_{\text{а}}, \quad (2.51)$$

$$T_{\text{мe}} = 2 \cdot 0,005 + 0,035 = 0,045 \text{ с}.$$

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

Коефіцієнт посилення $k_{\text{н}}$ рівний

$$k_{\text{н}} = \frac{U_{\text{зе max}}}{E_{\text{д max}}} \quad (2.52)$$

де $E_{\text{д max}}$ - ЕРС двигуна;

$U_{\text{зе max}}$ - напруга завдання, відповідна $E_{\text{д max}}$; приймаємо $U_{\text{зе max}} = 24 \text{ В}$.

У якості $E_{\text{д max}}$ приймемо ЕРС еквівалентного двигуна в режимі ідеального холостого ходу, тобто $E_{\text{д max}} = 2U_{\text{н}} = 440 \text{ В}$, тоді

$$k_{\text{н}} = \frac{24}{440} = 0,05454.$$

Зіставляючи передавальну функцію (2.50) з передавальною функцією типового ПІ – регулятора, отримуємо наступні рівняння

$$R_{\text{ззе}} C_{\text{ззе}} = 4T_{\text{мe}}, \quad (2.53)$$

$$\frac{R_{3\text{э}}}{R_{\text{зе}}} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\text{мe}} k_{\text{н}} R_{\Sigma}} . \quad (2.54)$$

Задаємося величиною ємності $C_{3\text{э}} = 2 \cdot 10^{-6} \text{ Ф}$, тоді з (2.53) отримуємо

$$R_{3\text{э}} = \frac{4T_{\text{мe}}}{C_{3\text{э}}} , \quad (2.55)$$

$$R_{3\text{э}} = \frac{4 \cdot 0,045}{2 \cdot 10^{-6}} = 90 \cdot 10^3 \text{ Ом}$$

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

$$k_{\text{н}} = k_{\text{ддв}} \frac{R_{3\text{э}}}{R_{\text{н}}} \quad (2.56)$$

визначаємо

$$R_{\text{н}} = \frac{k_{\text{ддв}} R_{3\text{э}}}{k_{\text{н}}} \quad (2.57)$$

$k_{\text{ддв}}$ – коефіцієнт посилення ділянки напруги двигуна, визначений при
розрахунку контура струму

$$R_{\text{н}} = \frac{0,05454 \cdot 90 \cdot 10^3}{0,05454} = 90 \cdot 10^3 \text{ Ом} .$$

Із співвідношення (2.27) знаходимо

$$R_{\text{зе}} = \frac{2T_{\text{мe}} k_{\text{н}} R_{\Sigma} R_{3\text{э}}}{T_{\text{м}} k_{\text{т}}} . \quad (2.58)$$

$$R_{\text{н}} = \frac{2 \cdot 0,045 \cdot 0,05454 \cdot 0,824 \cdot 90 \cdot 10^3}{0,186 \cdot 0,168} = 21 \cdot 10^3 \text{ Ом}$$

Коефіцієнт посилення ланцюга струмової компенсації:

$$k_{\text{TK}} = k_{\text{H}} R_{\text{ЯК}} , \quad (2.59)$$

$$k_{\text{TK}} = 0,05454 \cdot 0,038 = 2,09 \cdot 10^{-3} .$$

З іншого боку.

$$k_{\text{TK}} = k_{\text{Ш}} k_{\text{ДТ}} \frac{R_{\text{Зе}}}{R_{\text{TK}}} \quad (2.60)$$

звідки

$$R_{\text{TK}} = \frac{k_{\text{Ш}} k_{\text{ДТ}} R_{\text{Зе}}}{k_{\text{TK}}} . \quad (2.61)$$

$$R_{\text{TK}} = \frac{0,53 \cdot 10^{-3} \cdot 320 \cdot 21 \cdot 10^3}{2,09 \cdot 10^{-3}} = 91,32 \cdot 10^3 \text{ Ом}$$

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

$$\frac{R_{\text{Зе}}}{4} C_{\text{Зе}} = \frac{R_{\text{H}}}{4} C_{\text{H}} = T_a , \quad (2.62)$$

$$C_{\text{Зе}} = \frac{4T_a}{R_{\text{Зе}}} , \quad (2.63)$$

$$C_{\text{Зе}} = \frac{4 \cdot 0,045}{21 \cdot 10^3} = 8,57 \cdot 10^{-6} \text{ Ф}$$

$$C_{\text{H}} = \frac{4T_a}{R_{\text{H}}} , \quad (2.64)$$

$$C_{\text{H}} = \frac{4 \cdot 0,045}{90 \cdot 10^3} = 2 \cdot 10^{-6} \text{ Ф}$$

Передавальні функції системи, оптимізованої по симетричному критерію, мають вигляд:

$$W_{\omega_d, U_{ze}}(p) = \frac{\omega_d(p)}{U_{ze}(p)} = \frac{k_d}{k_H} \cdot \frac{B(p)}{A(p)}; \quad (2.65)$$

$$W_{\omega_d, I_{ct}}(p) = \frac{\omega_d(p)}{I_{ct}(p)} = -\frac{8T_{\mu e}^2 k_d R_{\Sigma}}{T_M} \cdot \frac{p(1 + T_{\mu e} p)}{A(p)}; \quad (2.66)$$

$$W_{I_d, U_{ze}}(p) = \frac{I_d(p)}{U_{ze}(p)} = \frac{T_M}{k_H R_{\Sigma}} \cdot \frac{pB(p)}{A(p)}; \quad (2.67)$$

$$W_{I_d, I_{ct}}(p) = \frac{I_d(p)}{I_{ct}(p)} = \frac{B(p)}{A(p)} \quad (2.68)$$

де $A(p)$ і $B(p)$ - поліноми від p що мають наступний вигляд:

$$A(p) = 8T_{\mu e}^3 p^3 + 8T_{\mu e}^2 p^2 + 4T_{\mu e} p + 1; \quad (2.69)$$

$$B(p) = 4T_{\mu e} p + 1. \quad (2.70)$$

Алгоритмічна схема системи показана на рисунку 2.9.

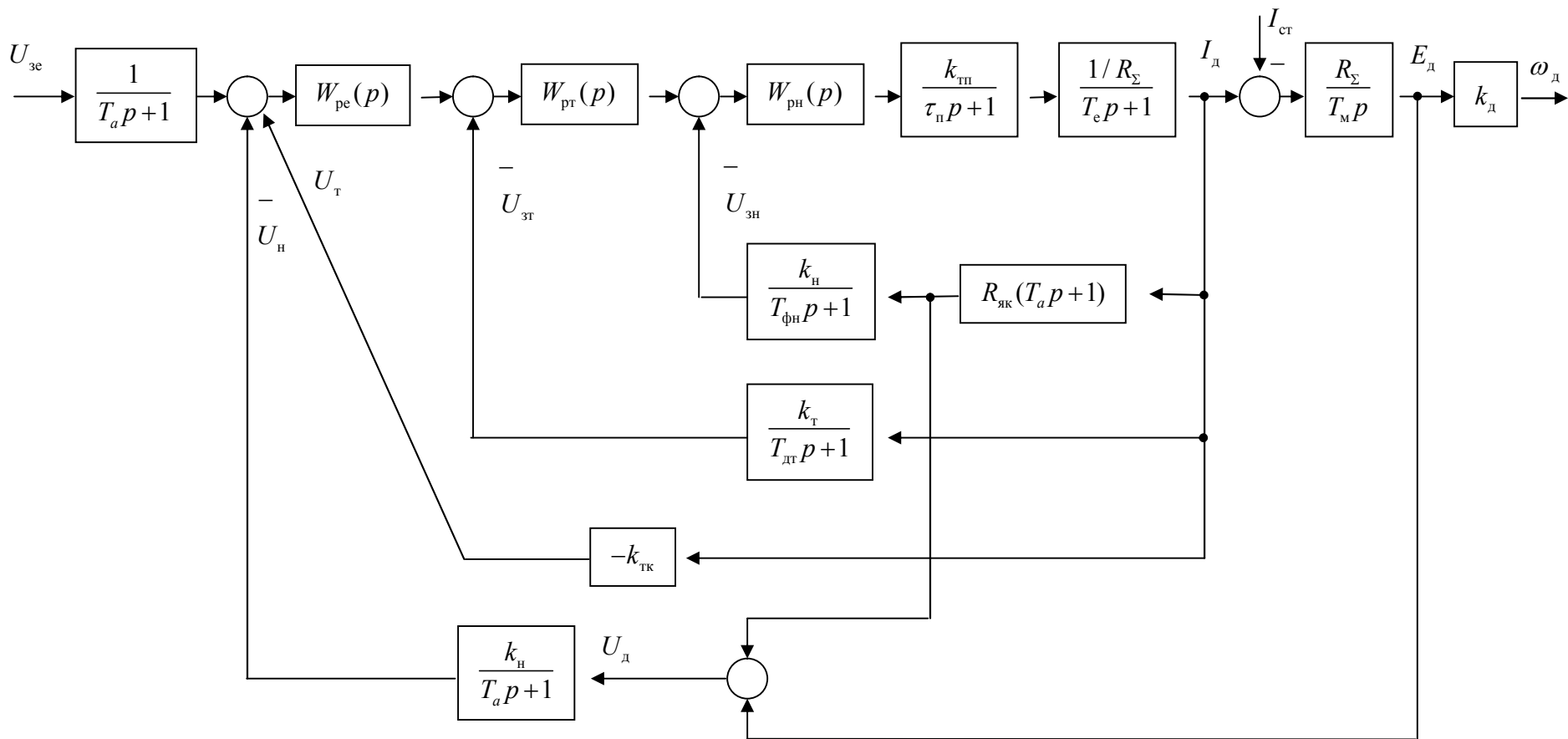


Рисунок 2.9 – Алгоритмічна схема системи

2.8 Дослідження динамічних характеристик системи

Для побудови перехідних процесів системи була розроблена на ПЕВМ за допомогою пакету MATLAB в режимі SIMULINK. Схема моделі показана на рисунку 2.10. Перехідні процеси досліджувалися також по векторний-матричному рівнянню системи, порядок складання якого дається в подальших розділах. На рисунку 2.11 приведені графіки перехідних процесів швидкості і моменту двигуна по задаючій дії. При настроюванні системи на симетричний оптимум є значне (43,3%) перерегулювання швидкості. Для зменшення перерегулювання на вихід системи доцільно включити фільтр з постійною часу, рівною $4T_{\mu e}$, унаслідок чого перерегулювання знижується до 8,1%, проте при цьому збільшується час першого узгодження. Криві $\omega_d = f(t)$ і $M_d = f(t)$ з фільтром і без фільтру позначені 1 і 2 відповідно.

На рисунку 2.12 показані графіки перехідних процесів швидкості і моменту електроприводу по збурюючій дії.

Як видно з графіків, при використанні фільтру на вході системи перехідні процеси для даних випадків мають задовільний характер.

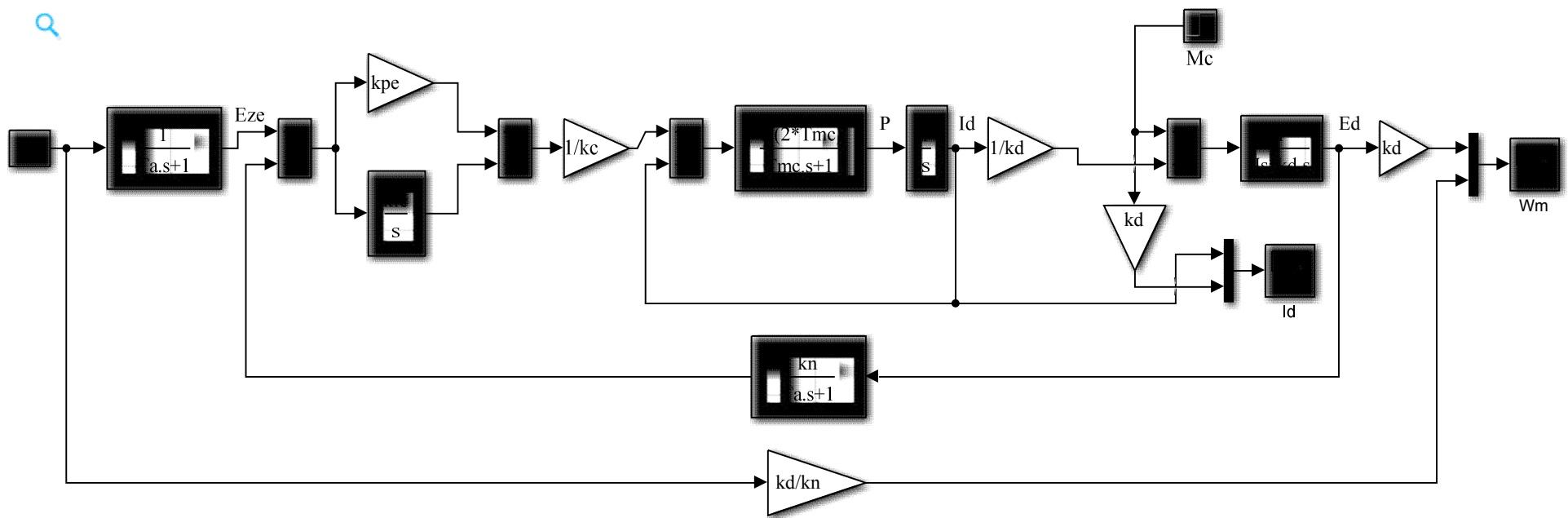


Рисунок 2.10 – Схема моделі системи, розроблена на ЕОМ

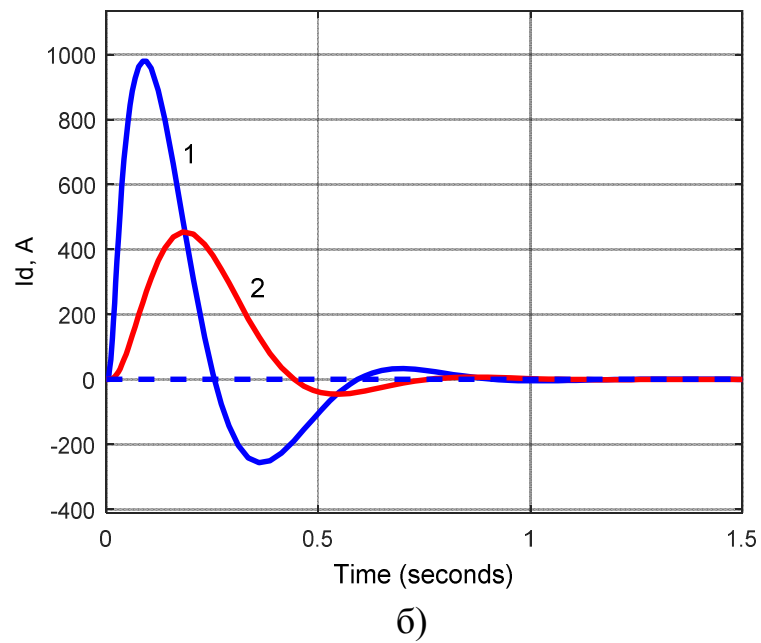
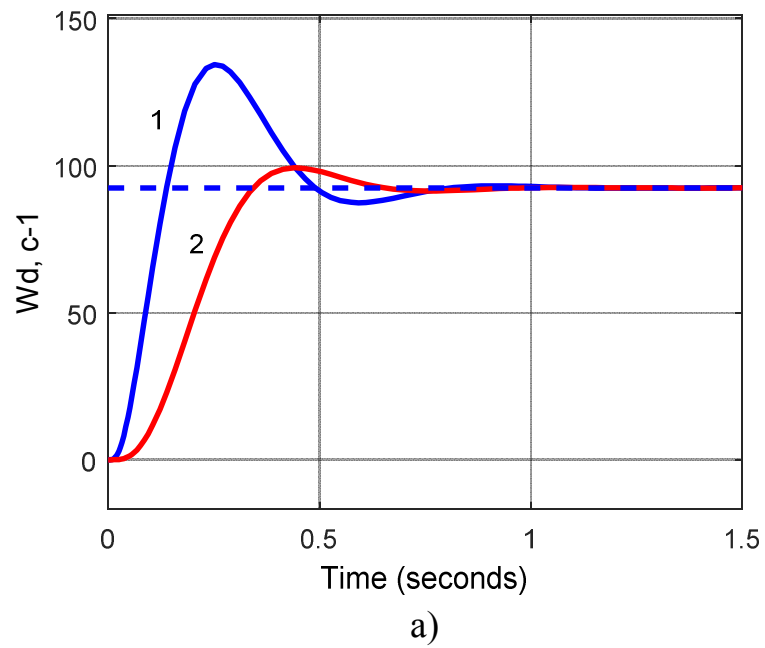
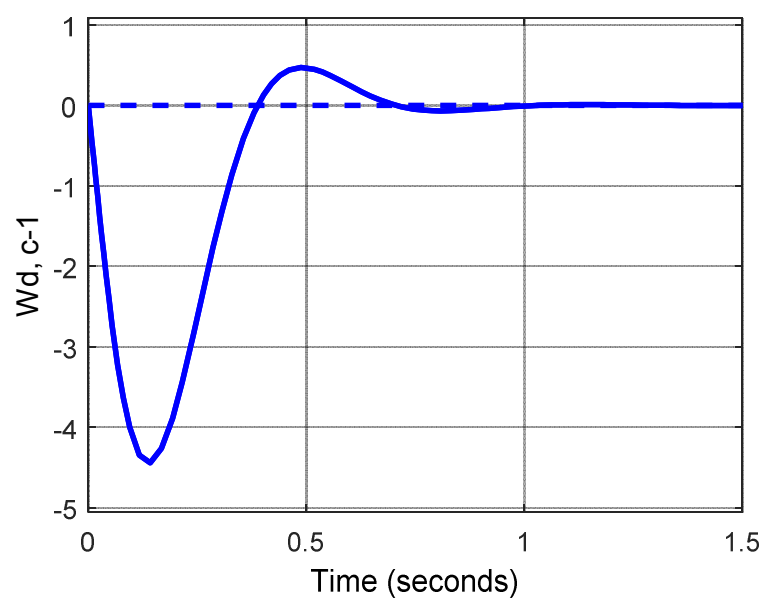
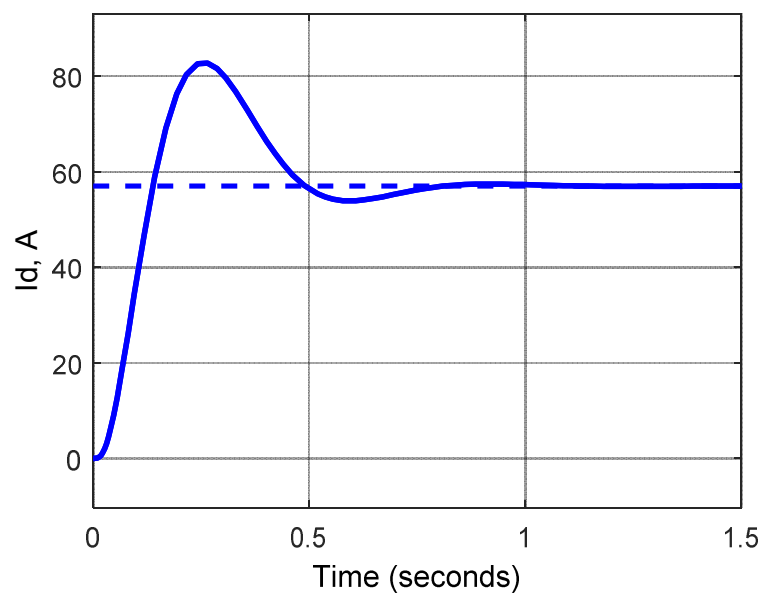


Рисунок 2.11 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ (а)
і струму $I_d = f(t)$ (б) по задаючій дії



a)



б)

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

ВИСНОВКИ ДО РОЗДІЛУ 2

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

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

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

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

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

6. Проведено моделювання синтезованої системи на ЕОМ з використанням пакету прокладних програм MATLAB. В режимі SIMULINK системи MATLAB розроблена схема моделі системи, що дозволяє виконати дослідження перехідних процесів в системі по задаючій і збурюючій діях. Перехідні процеси досліджувалися також по векторно-матричному рівнянню системи і з використанням передатних функцій системи. Під час налаштування системи на симетричний оптимум спостерігається значне (43,3%) перерегулювання швидкості. Щоб зменшити перерегулювання, до виходу системи додано фільтр з постійною часу, рівною $4T_{\mu e}$, унаслідок чого перерегулювання знижене до 8,1%.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

У попередніх розділах було проведено дослідження динамічних характеристик системи електроприводу механізму переміщення моста в припущенні нескінченної жорсткості всіх зв'язків між елементами механічної частини системи. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер руху елементів тільки в середньому, т.д. не враховує вплив пружності реальних механічних зв'язків. Внаслідок кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, додаток до якої дії (моменту двигуна), що управляє, або збурюючої дії (моменту статичного навантаження) викликає коливання зв'язаних мас, що збільшують максимальні навантаження зв'язків і відробітки необхідних рухів і переміщень, що ускладнюють точність.

Податливість механізму переміщення моста пов'язана як з кінцевою жорсткістю зв'язку між двигуном і мостом, так і розгойдуванням вантажу.

Така система може розглядатися як трьохмасова. Проте оскільки жорсткість в передачах у багато разів вище, ніж жорсткість при розгойдуванні вантажу, коливання вантажу при аналізі навантажень передач можна не враховувати, вважаючи його відхилення постійним в процесі розгону приводу, тобто розглядати двомасовою системою.

Для електроприводу механізму пересування моста використовуються два двигуни. Така схема багатодвигунного електроприводу в загальному випадку приводиться до розгалуженої розрахункової схеми, показаної на рис.3.1.

На схемі позначено:

$J_{д1}$ і $J_{д2}$ – моменти інерції двигунів;

c_1 , c_2 – приведені жорсткості кожного приводу;

$J_{\text{м}}$ – приведений момент інерції механізму; тобто приведений до валу двигуна момент інерції моста.

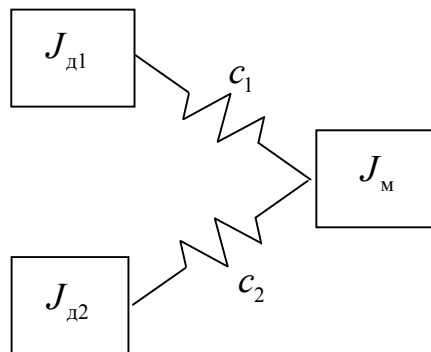


Рисунок 3.1 – Розгалужена розрахункова схема механічної частини електроприводу

Оскільки двигуни приводу рухаються синфазно, розгалужену схему рис.3.1 можна звести до двомасової розрахункової механічної схеми рис.3.2,

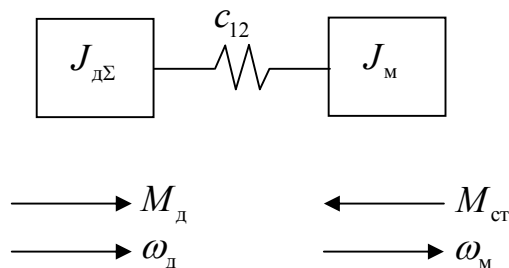


Рисунок 3.2 – Двомасова розрахункова схема

На схемі позначено:

$J_{\text{д}\Sigma}$ – сума моменту двигуна і жорстко пов'язаних з ним елементів механічної частини.

$J_{\text{м}}$ – момент інерції моста, приведений до швидкості валу двигуна рівний:

c_{12} – приведена жорсткість зв'язку між масами

$$c_{12} = c_1 + c_2. \quad (3.1)$$

Рівняння динаміки, що описують процеси з двомасовою системою при приведенні всіх елементів до швидкості валу двигуна, мають вид:

$$\begin{cases} J_{\text{м}} \frac{d\omega_{\text{м}}}{dt} = M_{\Sigma} - M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12}(\omega_{\text{д}} - \omega_{\text{м}}); \\ J_{\text{д}\Sigma} \frac{d\omega_{\text{д}}}{dt} = k\Phi_{\text{н}} I_{\text{я}} - M_{\Sigma}, \end{cases} \quad (3.2)$$

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

$$M_{\Sigma} = M_{\text{пр}} + \beta(\omega_{\text{д}} - \omega_{\text{м}}), \quad (3.3)$$

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

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

$M_{\text{ст}}$ – момент статичного навантаження, прикладений до другої маси.

Розглянемо двомасову систему підлеглого регулювання швидкості з контуром струму, настроєним по модульному критерію і контуром ЕРС, настроєним по симетричному критерію. Алгоритмічна схема двомасової системи показана на рис. 3.3.

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

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

Одномасова система підлеглого регулювання швидкості з контуром струму, настроєним по модульному критерію і контуро ЕРС, настроєним по

симетричному критерію описується наступною системою диференціальних рівнянь, записаних у формі Коші:

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

При записі системи (3.5) прийняті наступні позначення.

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

P – ривок.

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

$$W_{\text{пе}}(p) = \frac{T_{\text{м}} k_{\text{T}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} \cdot \frac{4T_{\mu\text{е}} p + 1}{4T_{\mu\text{е}} p} = \frac{T_{\text{м}} k_{\text{T}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} + \frac{T_{\text{м}} k_{\text{T}}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}} \frac{1}{p} = k_{\text{пе}} + \frac{k_{\text{іе}}}{p}, \quad (3.6)$$

$$k_{\text{пе}} = \frac{T_{\text{м}} k_{\text{T}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}}, \quad k_{\text{іе}} = \frac{T_{\text{м}} k_{\text{T}}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}}. \quad (3.7)$$

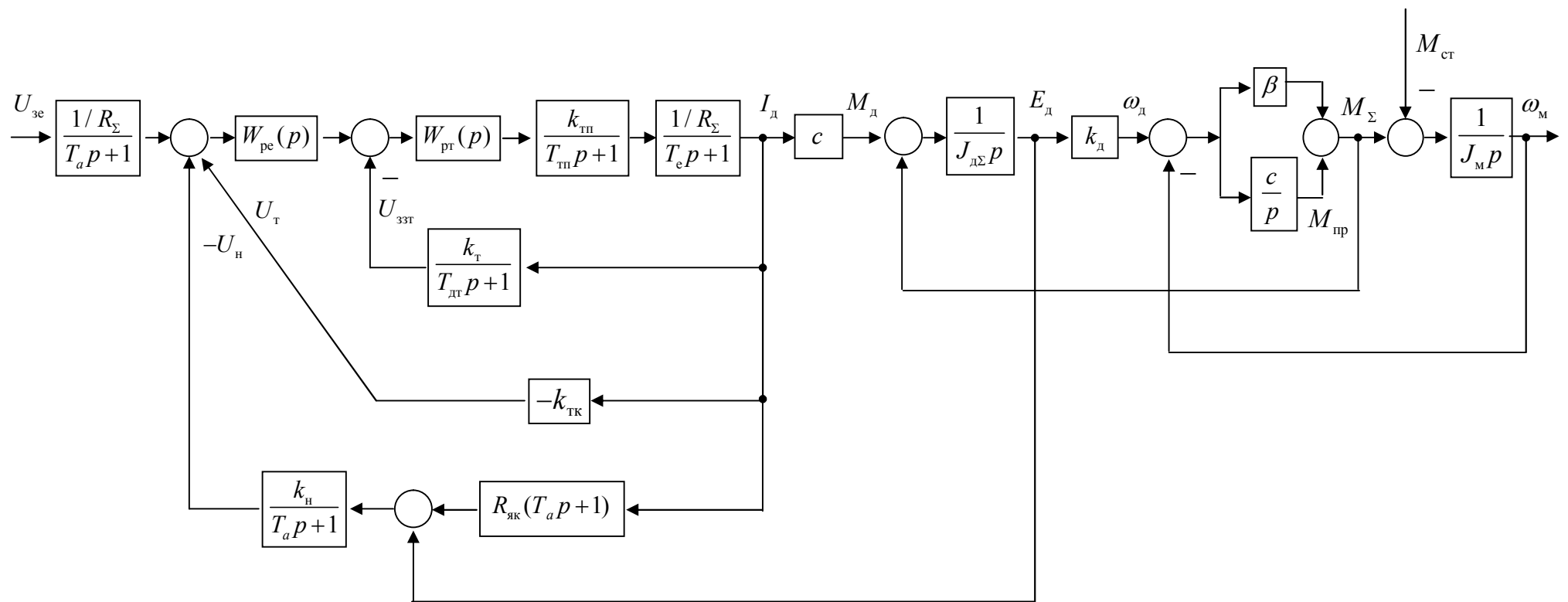


Рисунок 3.3 – Алгоритмічна схема двомасової системи

При записі системи (3.5) замість T_m записано відповідний вираз

$$T_m = \frac{R_\Sigma \cdot J_{\Sigma}}{c^2} = R_\Sigma J_{\Sigma} \cdot k_d^2; \quad k_d = 1/c.$$

Об'єднаємо системи (3.4) і (3.5), одержимо наступну систему рівнянь стану двомасової системи підлеглого регулювання швидкості із зворотним зв'язком по ЕРС двигуна:

$$\left\{ \begin{array}{l} \frac{d\omega_m}{dt} = \frac{1}{J_m} M_{\text{пр}} + \frac{\beta}{J_m} \omega_d - \frac{\beta}{J_m} \omega_m - \frac{1}{J_m} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_d - c_{12} \omega_m; \\ \frac{d\omega_d}{dt} = \frac{1}{J_{\Sigma}} M_d - \frac{1}{J_{\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\Sigma}} \omega_d - \frac{\beta}{J_{\Sigma}} \omega_m; \\ \frac{dM_d}{dt} = \frac{1}{k_d} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_T} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_T} U_{\text{ззе}} - \frac{k_d}{2T_{\mu T}^2} M_d - \frac{1}{T_{\mu T}} P + \frac{1}{T_{\mu T}^2 k_T} U_{\text{іе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_a} U_{\text{зе}} - \frac{1}{T_a} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_n}{T_a k_d} \omega_d - \frac{1}{T_a} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}. \end{array} \right. \quad (3.8)$$

Алгоритмічна схема двомасової системи, що відповідає рівнянням (3.8), показана на рис. (3.4). На схемі оптимізований по модульному критерію контур струму представлений у вигляді послідовного з'єднання аперіодичної і інтегруючої ланки, охоплених одиничним зворотним зв'язком. Сигнал на виході першої ланки пропорційний ривку електроприводу P .

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

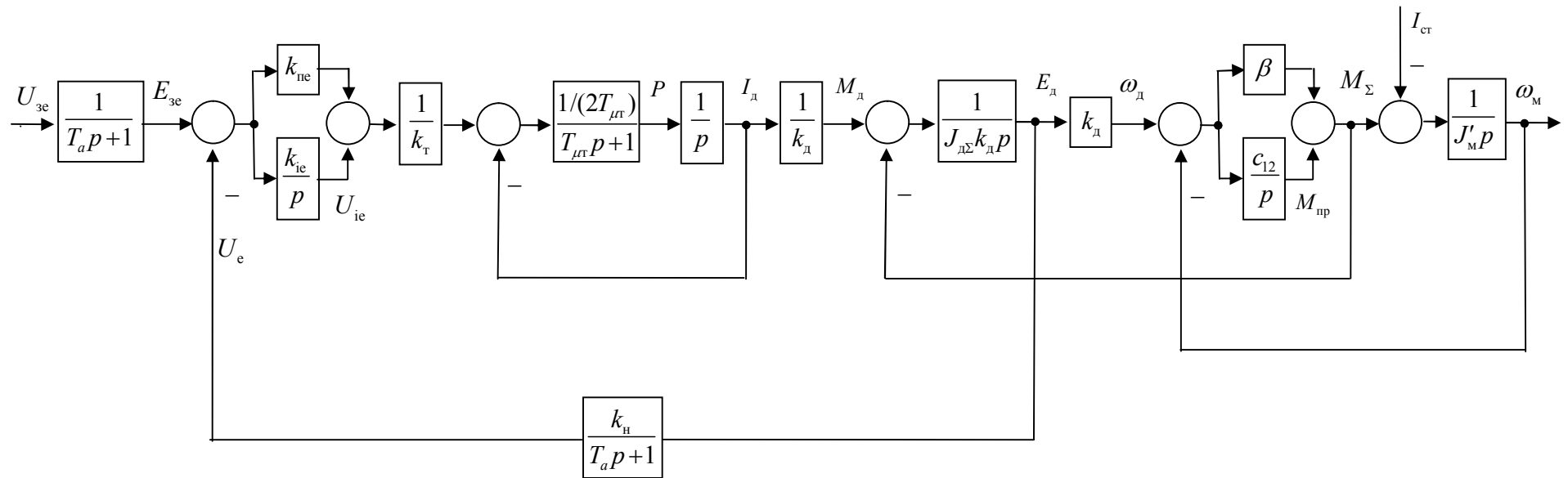


Рисунок 3.4 – Спрощена алгоритмічна схема двомасової системи

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

$$\vec{X}(t) = \{ \omega_{\text{м}}(t), M_{\text{пр}}(t), \omega_{\text{д}}(t), M_{\text{д}}(t), P(t), E_{\text{зе}}(t), U_{\text{ззе}}(t), U_{\text{іс}}(t) \}^T \quad (3.9)$$

$$\vec{U}(t) = \{ U_{\text{зе}}(t), M_{\text{ст}}(t) \}^T \quad (3.10)$$

де T – знак транспонування.

Рівняння стану системи і рівняння виходу має вигляд

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (3.11)$$

де $\mathbf{A}$ – матриця стану;

$\mathbf{B}$ – матриця управління;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ – вектор виходу.

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

$$\mathbf{A} = \begin{matrix} & \begin{matrix} \omega_{\text{м}} & M_{\text{пр}} & \omega_{\text{д}} & M_{\text{д}} & P & E_{\text{зе}} & U_{\text{ззе}} & U_{\text{іс}} \end{matrix} \\ \begin{matrix} -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 & 0 & 0 & 0 \\ -c_{12} & & c_{12} & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{k_{\text{д}}}{2T_{\text{мт}}^2} & -\frac{1}{T_{\text{мт}}} & \frac{k_{\text{пе}}}{2T_{\text{мт}}^2 k_{\text{т}}} & -\frac{k_{\text{пе}}}{2T_{\text{мт}}^2 K_{\text{т}}} & \frac{1}{2T_{\text{мт}}^2 k_{\text{т}}} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} & 0 & 0 \\ 0 & 0 & \frac{k_{\text{н}}}{T_{\text{а}} K_{\text{д}}} & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} & 0 \\ 0 & & 0 & 0 & 0 & k_{\text{іс}} & -k_{\text{іс}} & 0 \end{matrix} & \end{matrix} \quad \mathbf{B} = \begin{matrix} & \begin{matrix} U_{\text{зе}} & M_{\text{ст}} \end{matrix} \\ \begin{matrix} 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_{\text{а}}} & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} & \end{matrix}$$

3.2 Розрахунок перехідних процесів двомасової системи на ЕОМ

Визначимо параметри, необхідні для подальших розрахунків.

Сумарний моменту двигуна і жорстко пов'язаних з ним елементів механічної частини.

$$J_{\Sigma} = 1,2(J_{\Sigma 1} + J_{\Sigma 2}) = 1,2 \cdot 2 \cdot J_{\Sigma}, \quad (8.12)$$

$$J_{\Sigma} = 1,2 \cdot 2 \cdot 0,425 = 1,02 \text{ кг} \cdot \text{м}^2$$

Момент інерції моста, приведений до швидкості валу двигуна рівний:

$$J_{\Sigma} = \frac{G_{\Sigma} \cdot V_{\Sigma}^2}{\omega_{\Sigma}^2}, \quad (8.13)$$

G_{Σ} – маса механізму: $G_{\Sigma} = G + Q_{\Sigma}$

$$J_{\Sigma} = \frac{(37 + 42) \cdot 10^3 \cdot 0,85^2}{83,7^2} = 4,06 \text{ кг} \cdot \text{м}^2$$

Решта параметрів визначена вище.

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

Текст програми, що містить початкові дані, матриця **A**, **B** і **C** приведений в додатку Д.

На рис. 3.6-3.9 приведені графіки перехідних процесів двомасової системи при урахування пружності зв'язку між двигуном і мостом. Як видно, графіки мають коливальний характер. Слід зазначити, що унаслідок обмеженого об'єму пояснювальної записки з 8 змінних стану системи приводяться перші чотири, як найбільш важливі для характеристики процесів в системі.

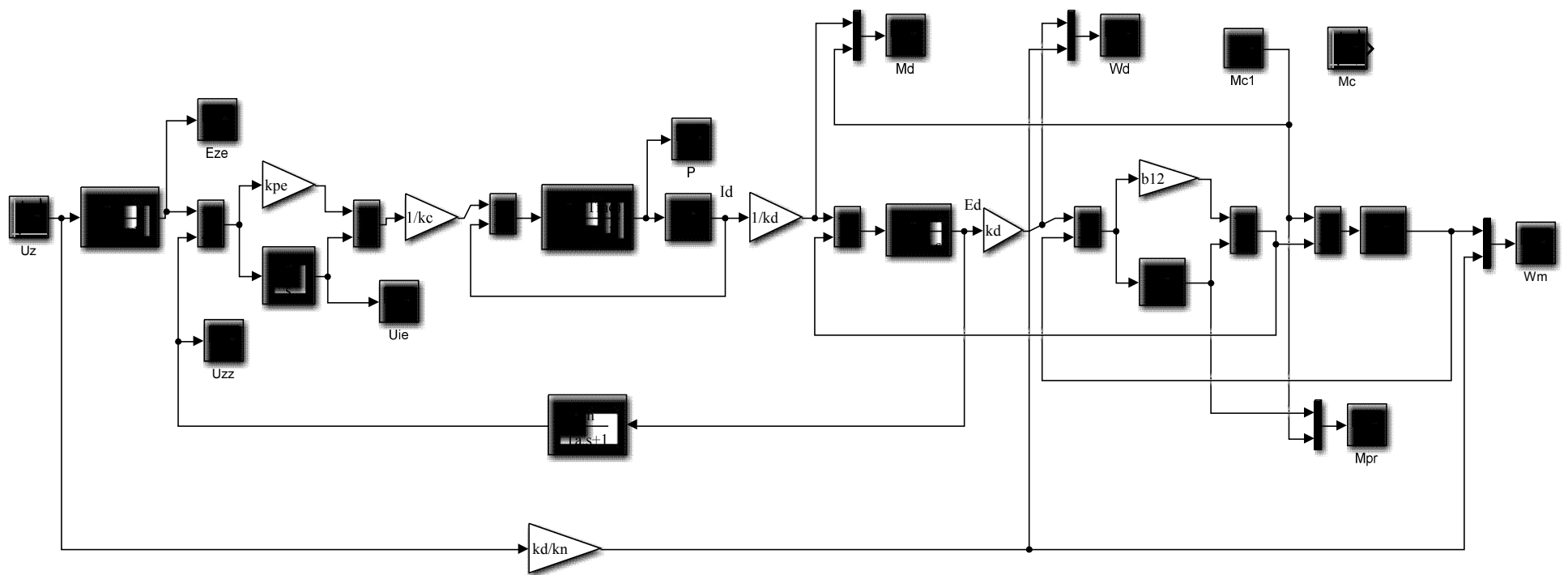


Рисунок 3.5 – Схема моделі двомасової системи, розроблена на ЕОМ

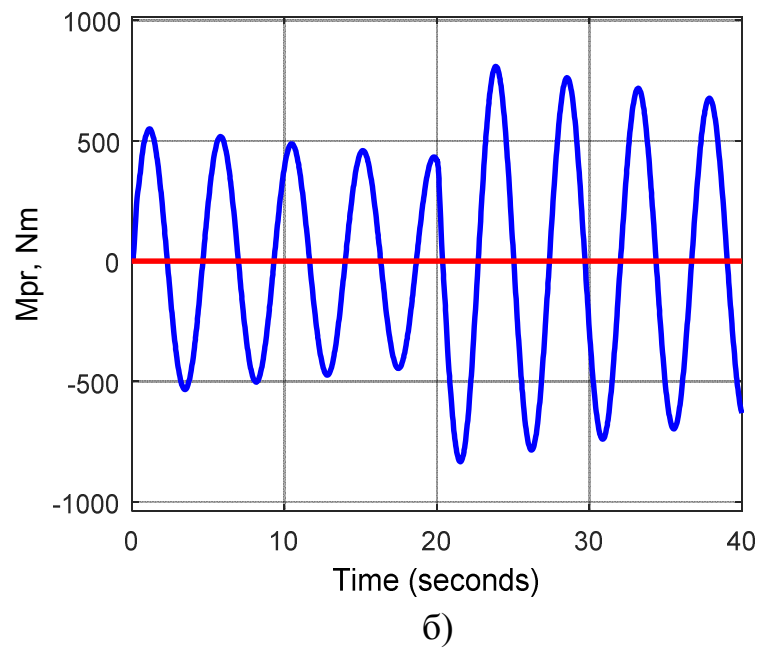
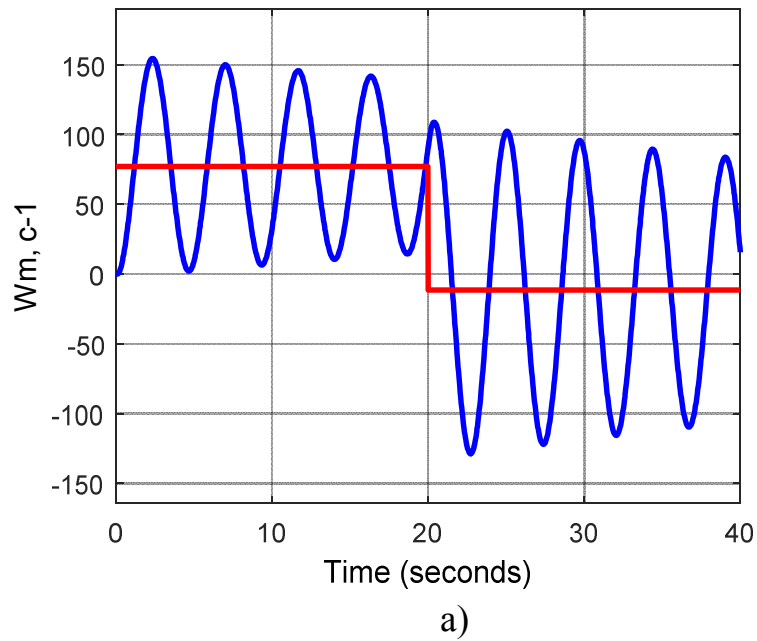


Рисунок 3.6 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії

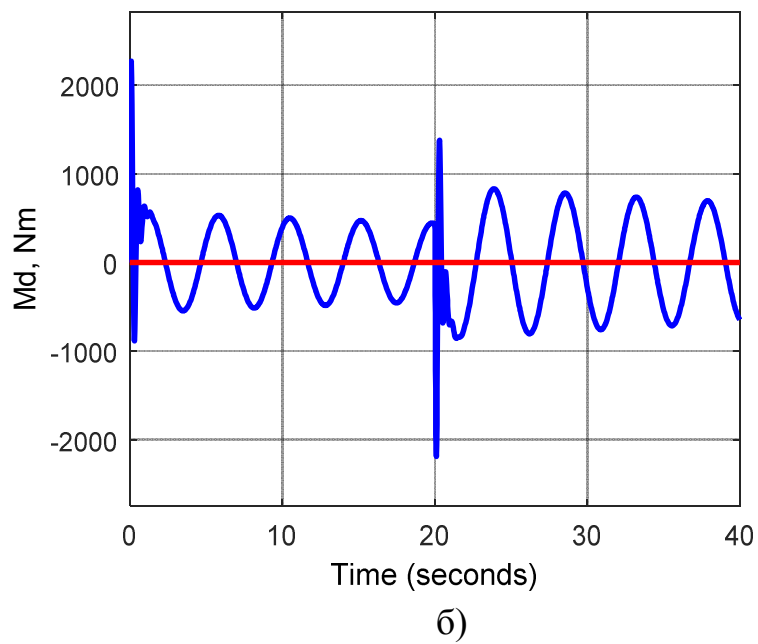
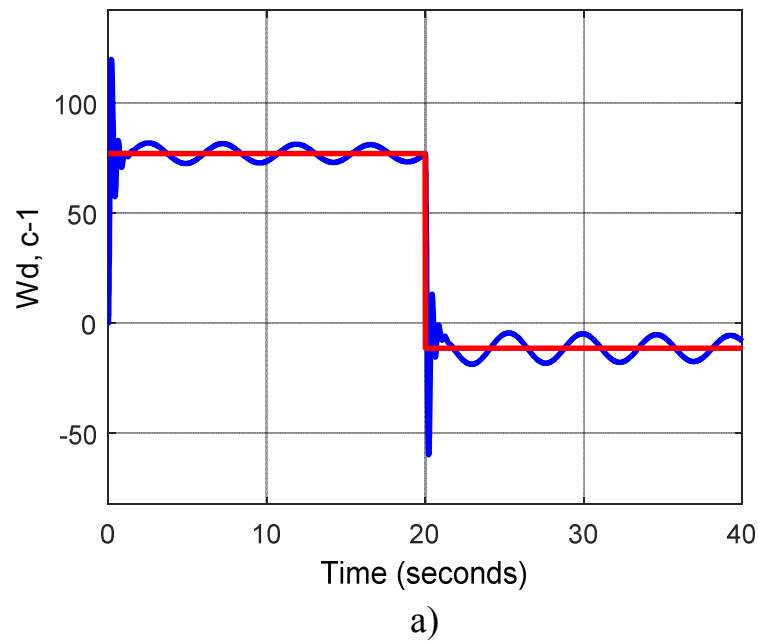


Рисунок 3.6 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по задаючій дії

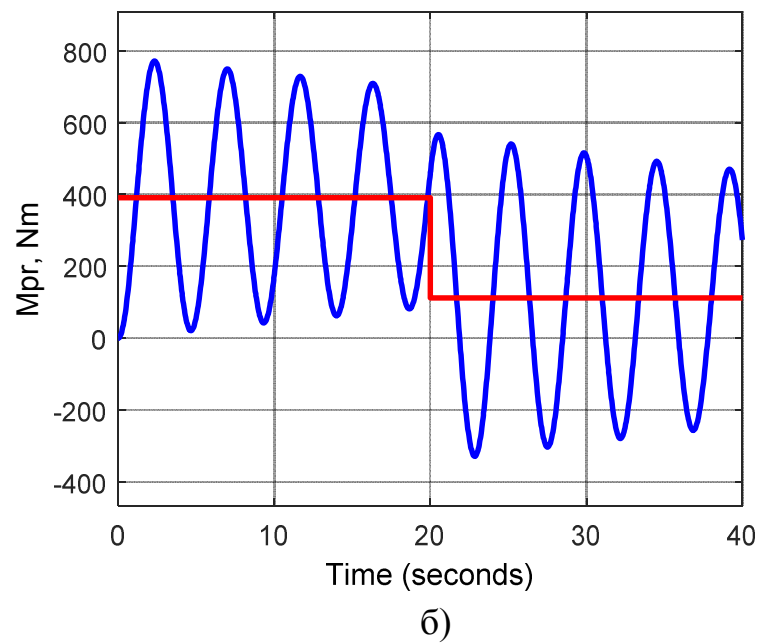
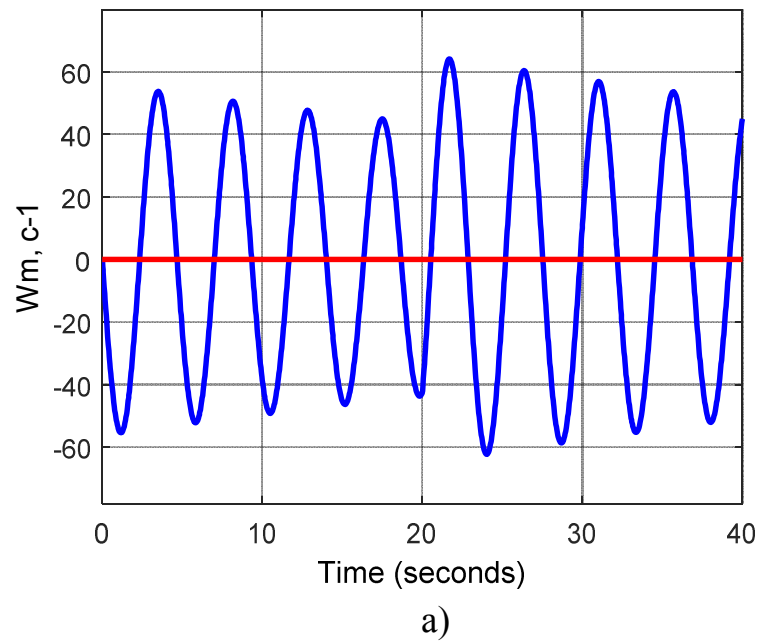


Рисунок 3.8 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по збурюючій дії

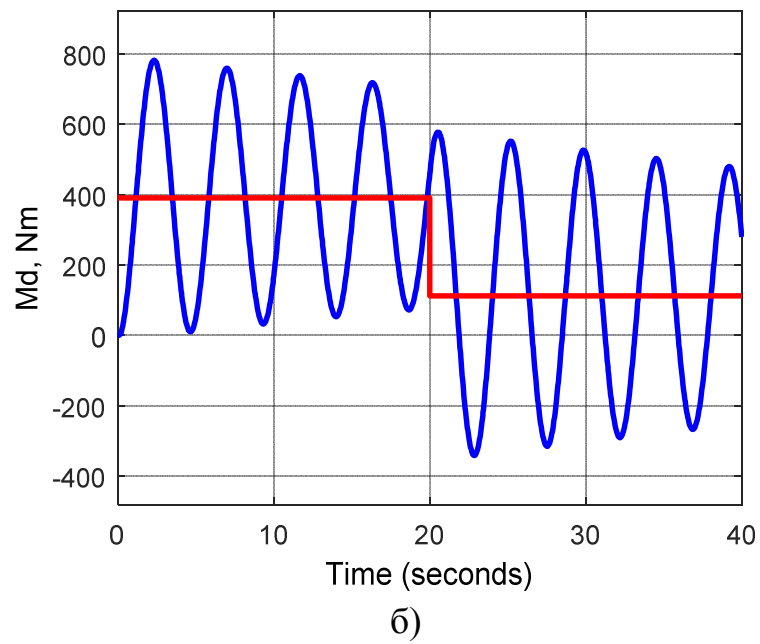
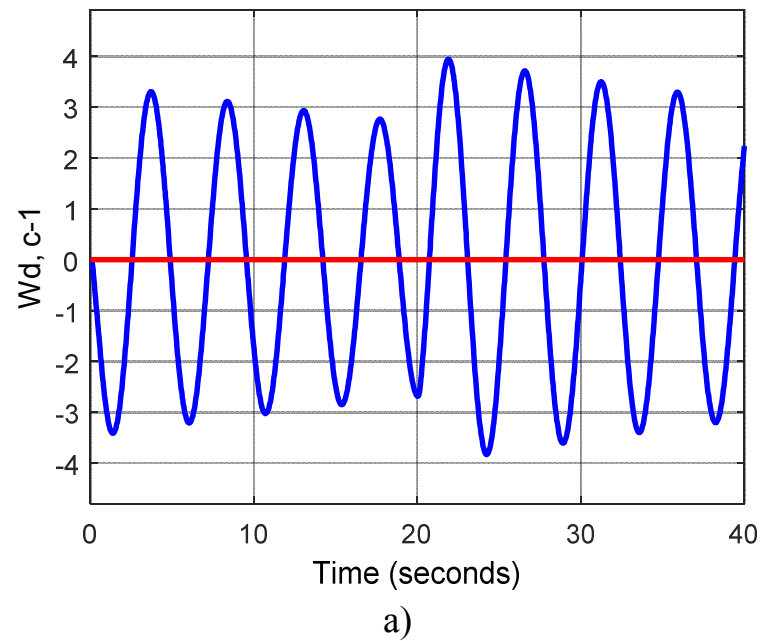


Рисунок 3.9 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму переміщення моста з урахуванням кінцевої жорсткості зв'язку між двигунами і мостом. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер рухів елементів тільки в середньому, оскільки не враховує вплив пружності реальних механічних зв'язків. Із-за кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, додаток до якої управляючої дії або збудуючої дії (моменту статичного навантаження), викликає коливання зв'язаних мас, що збільшують максимальні навантаження зв'язків і ускладнюють точність обробки необхідних рухів і переміщень. Показано, що кінематична схема механічної частини системи управління механізму переміщення моста з урахуванням податливості елементів механічної частини електроприводу може розглядатися як двомасова.

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

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

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

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

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

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

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

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

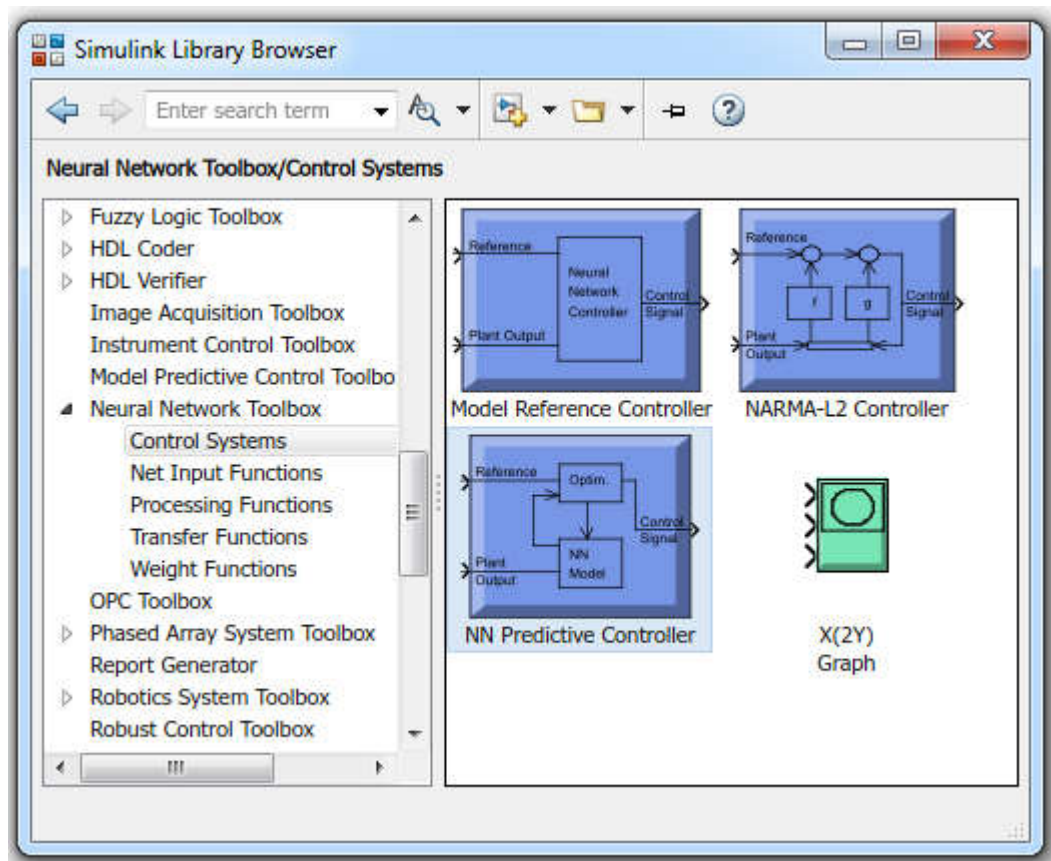


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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

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

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

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

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

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

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

4.2.1 Ідентифікація об'єкту управління.

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

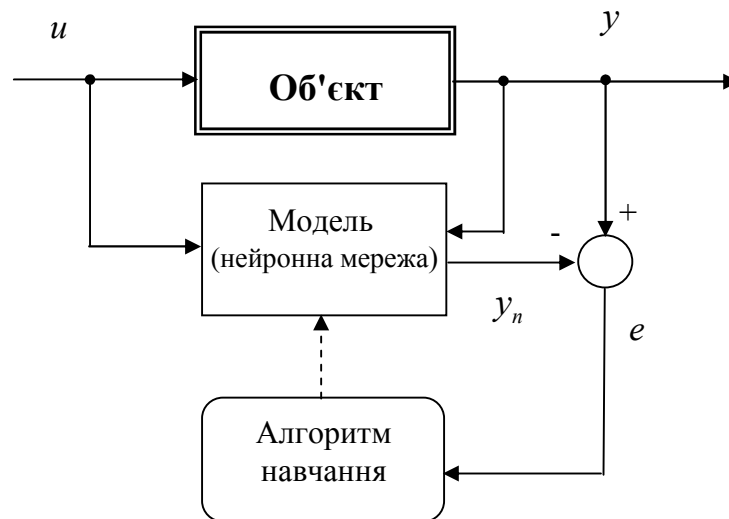


Рисунок 4.2 – Схема підсистеми ідентифікації.

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

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

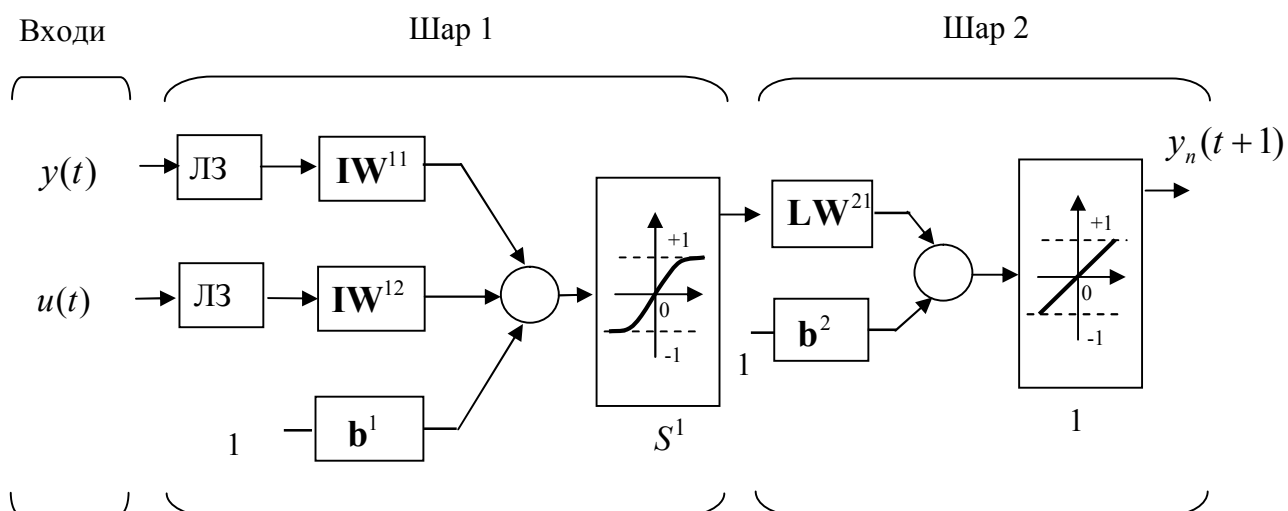


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

4.2.2 Схема управління з прогнозом.

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

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

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

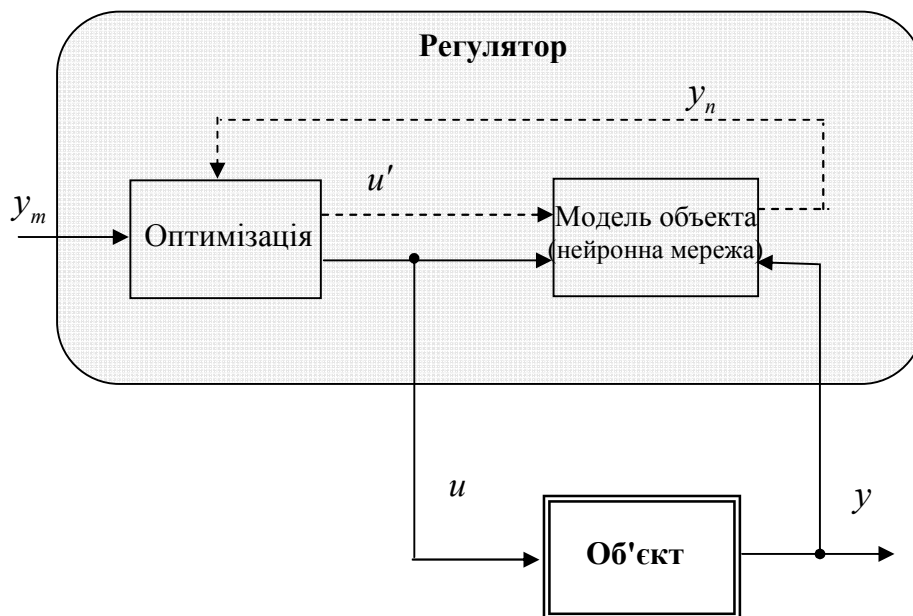


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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

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

Функції одновимірної оптимізації:

Csrchbac – пошук із зворотним прогоном;

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

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjjdjj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

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

Допоміжні функції:

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

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

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

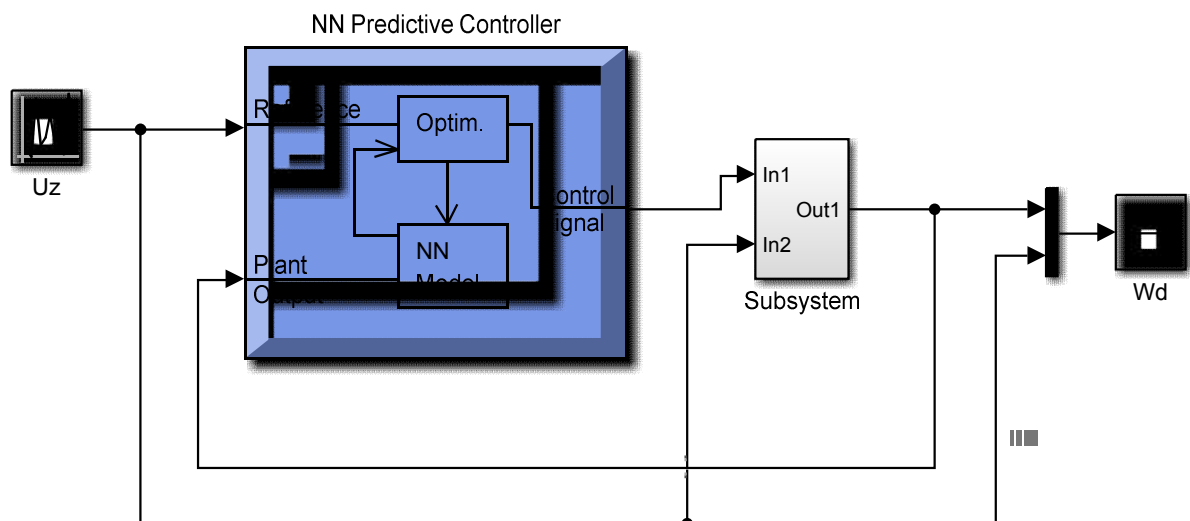


Рисунок 4.5 – Схема системи управління з нейрорегулятором

NN Predictive Controller

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

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

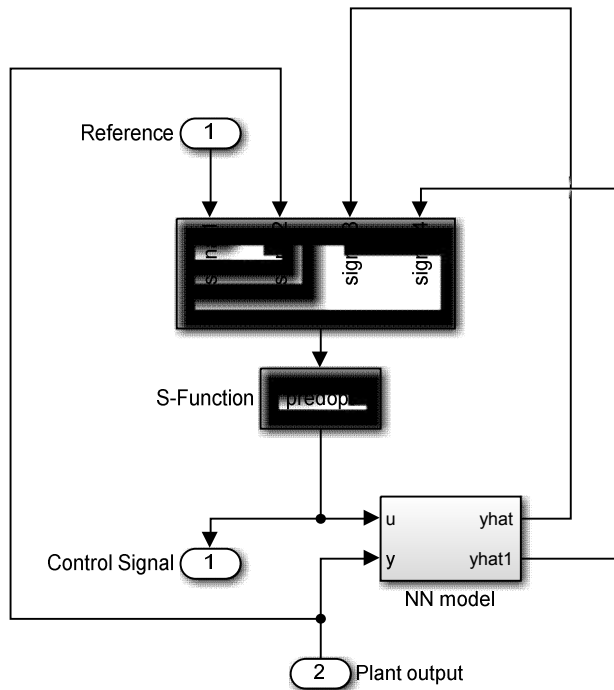


Рисунок 4.6 – Структурна схема нейрорегулятора

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

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

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

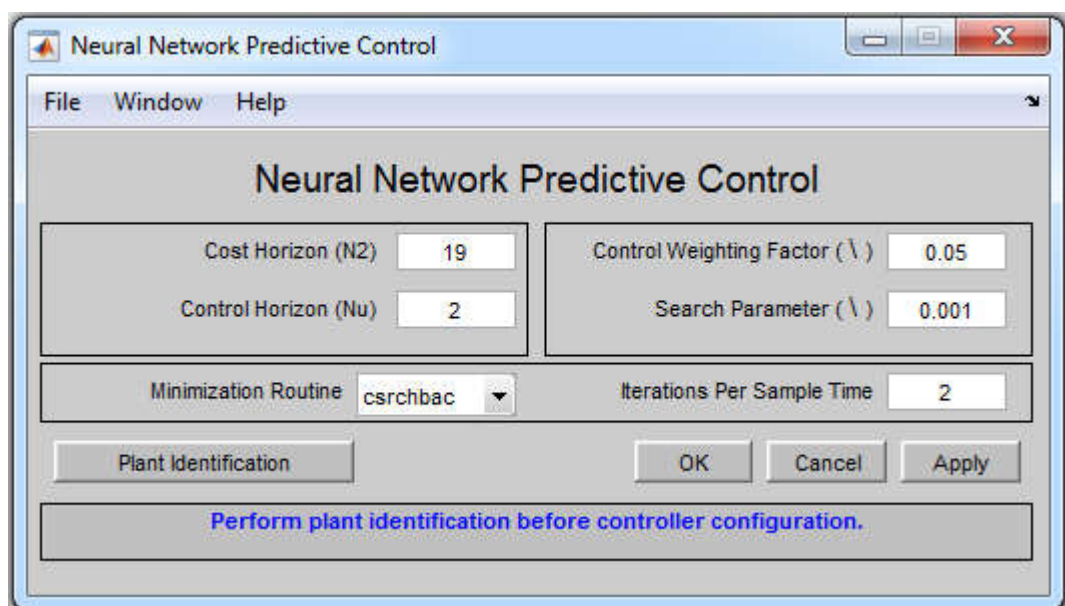


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

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

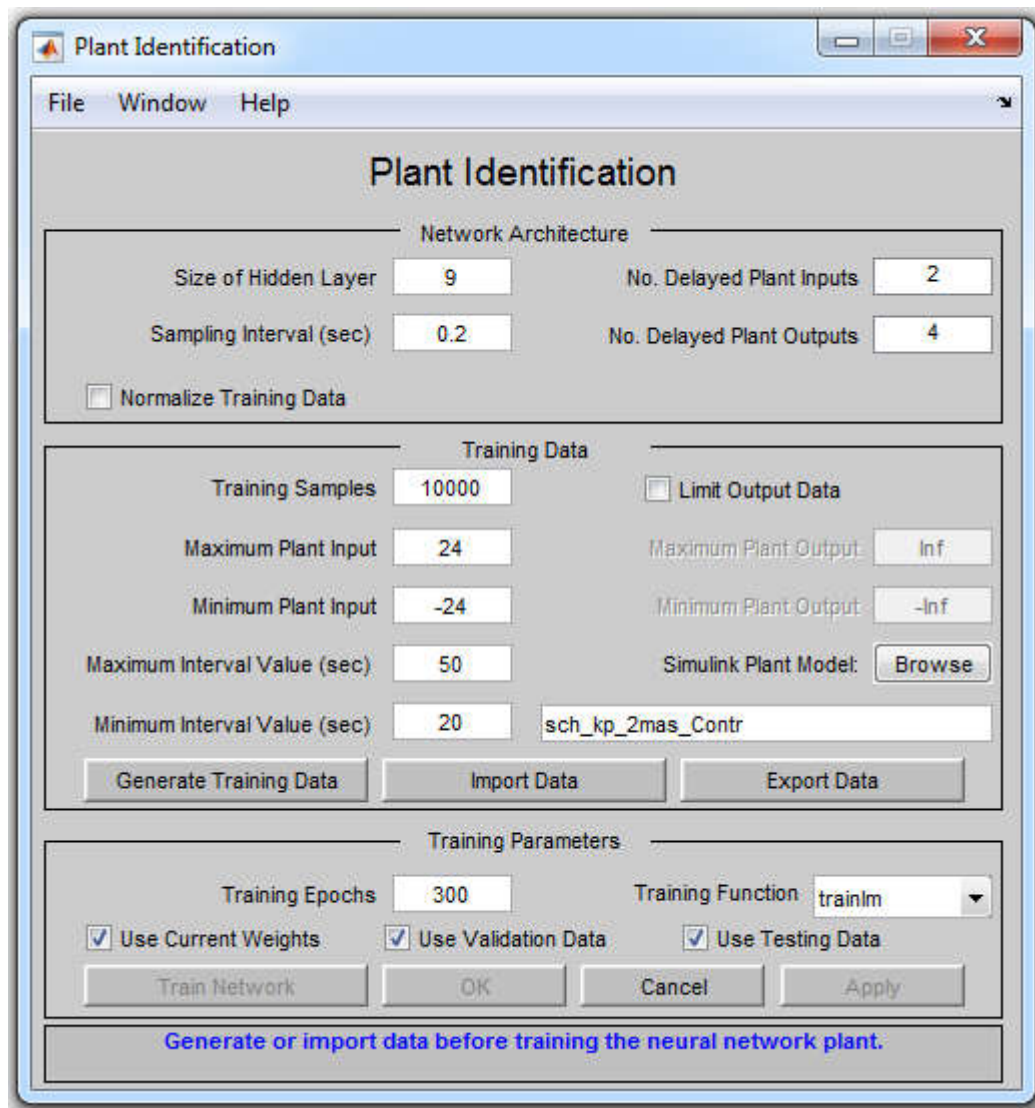


Рисунок 4.8 – Вікно ідентифікації керованої системи

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

Опис вікна Plant Identification

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Maximum Plant Output. Максимальне значення вихідного сигналу.

Minimum Plant Output. Мінімальне значення вихідного сигналу.

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

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

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

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

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

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

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

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

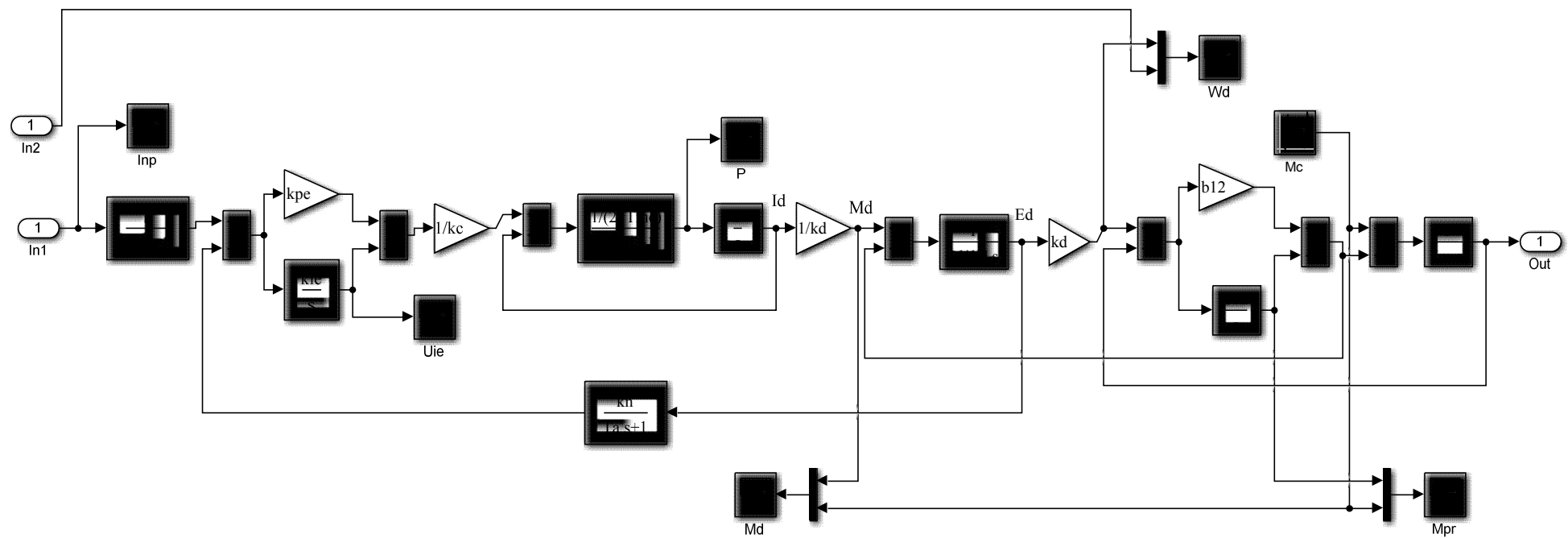


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

Секція Training Parameters (параметри навчання). Training Parameters

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

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

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

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

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

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

найменших квадратів Левенберга-Марквардта; trainbr - байєсівський регулятор.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

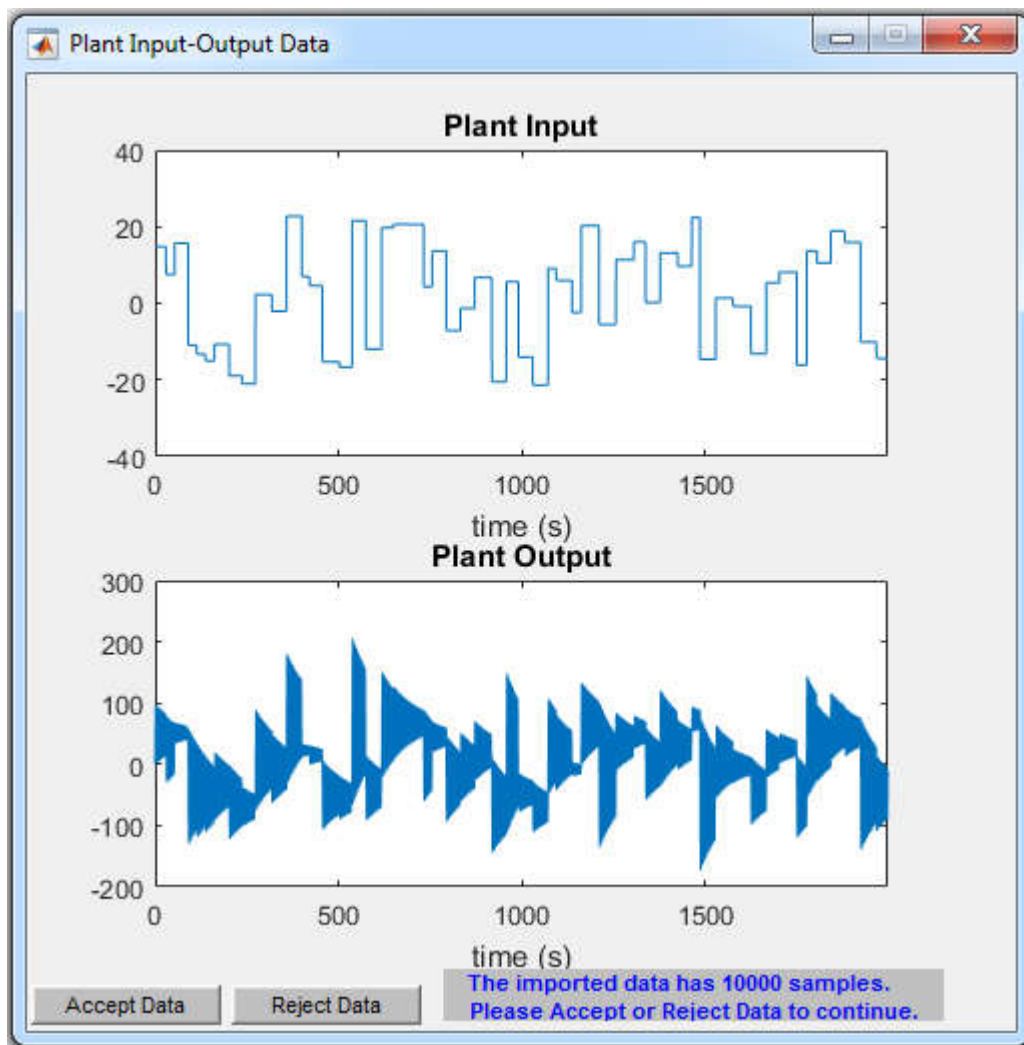


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

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

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

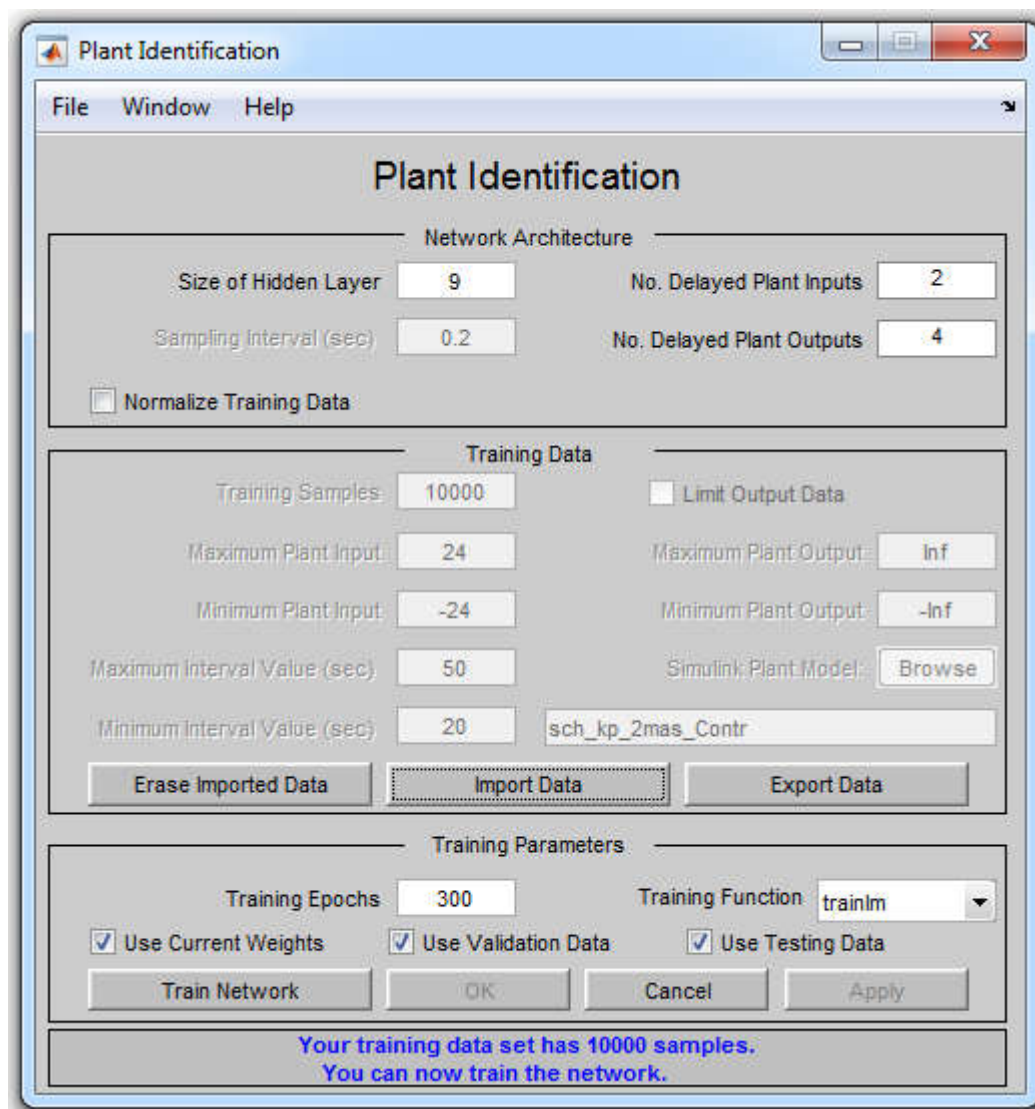


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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

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

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

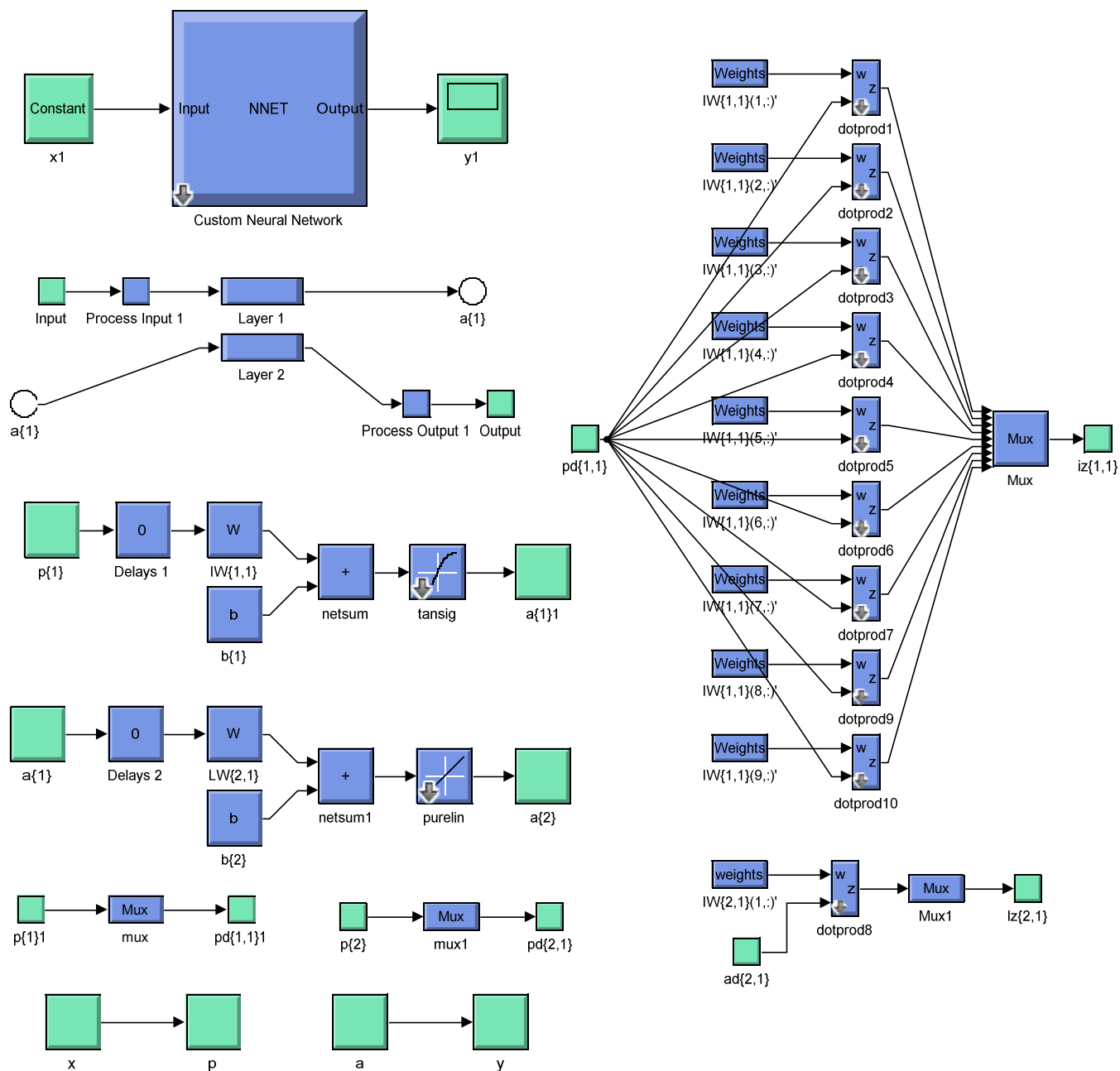


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

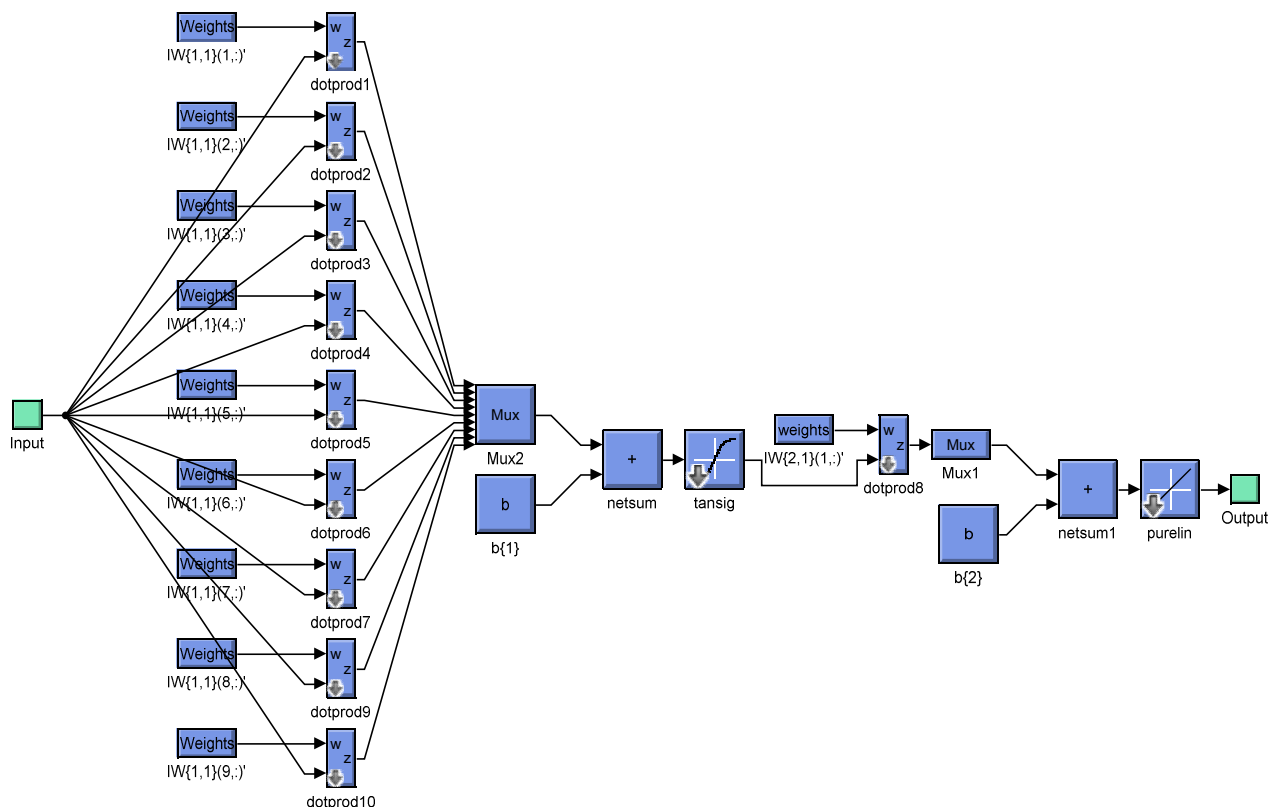


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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

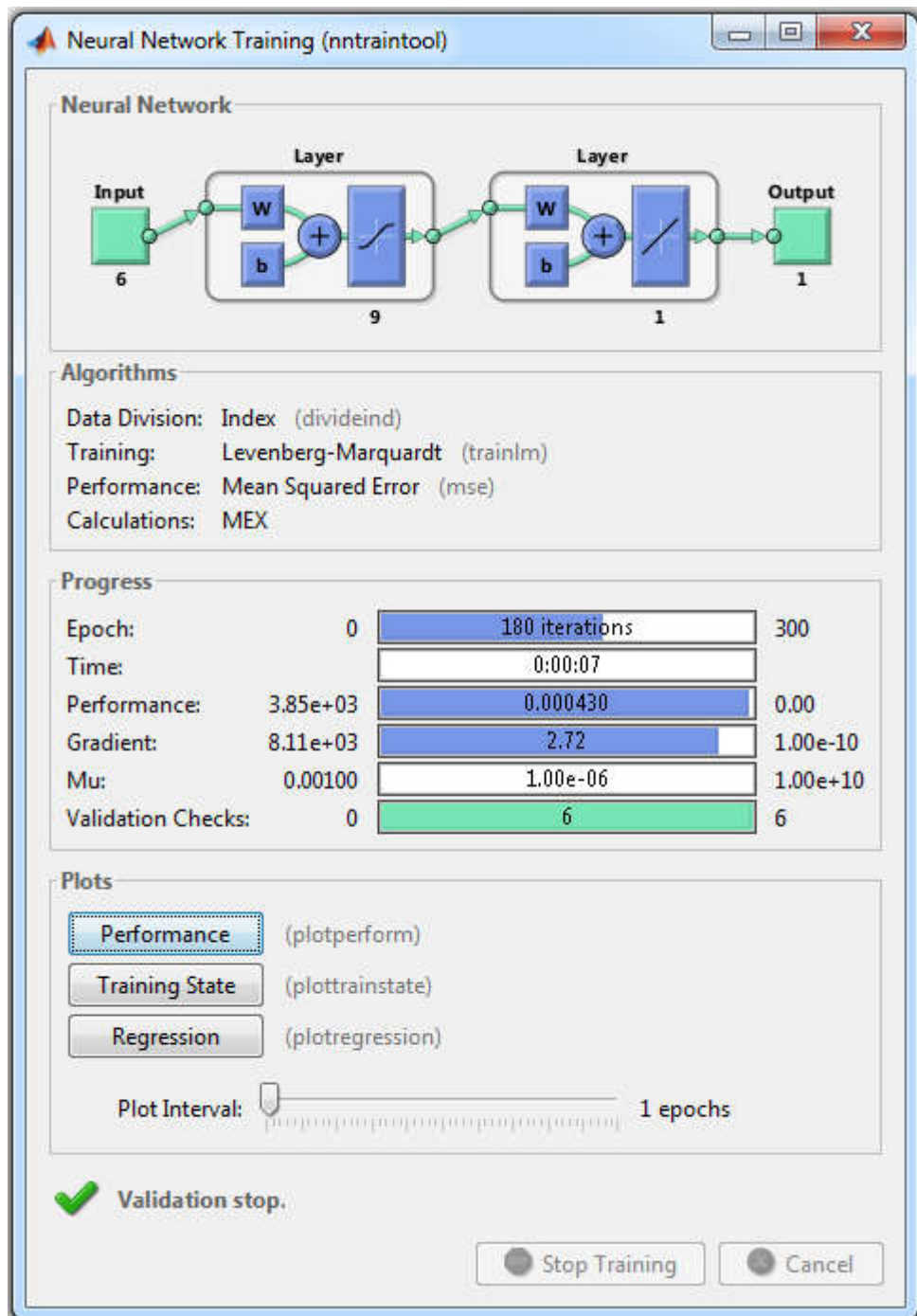


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

Вікно Neural Network Training (nntraintool) має наступні секції.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

на наборі даних для перевірки 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.

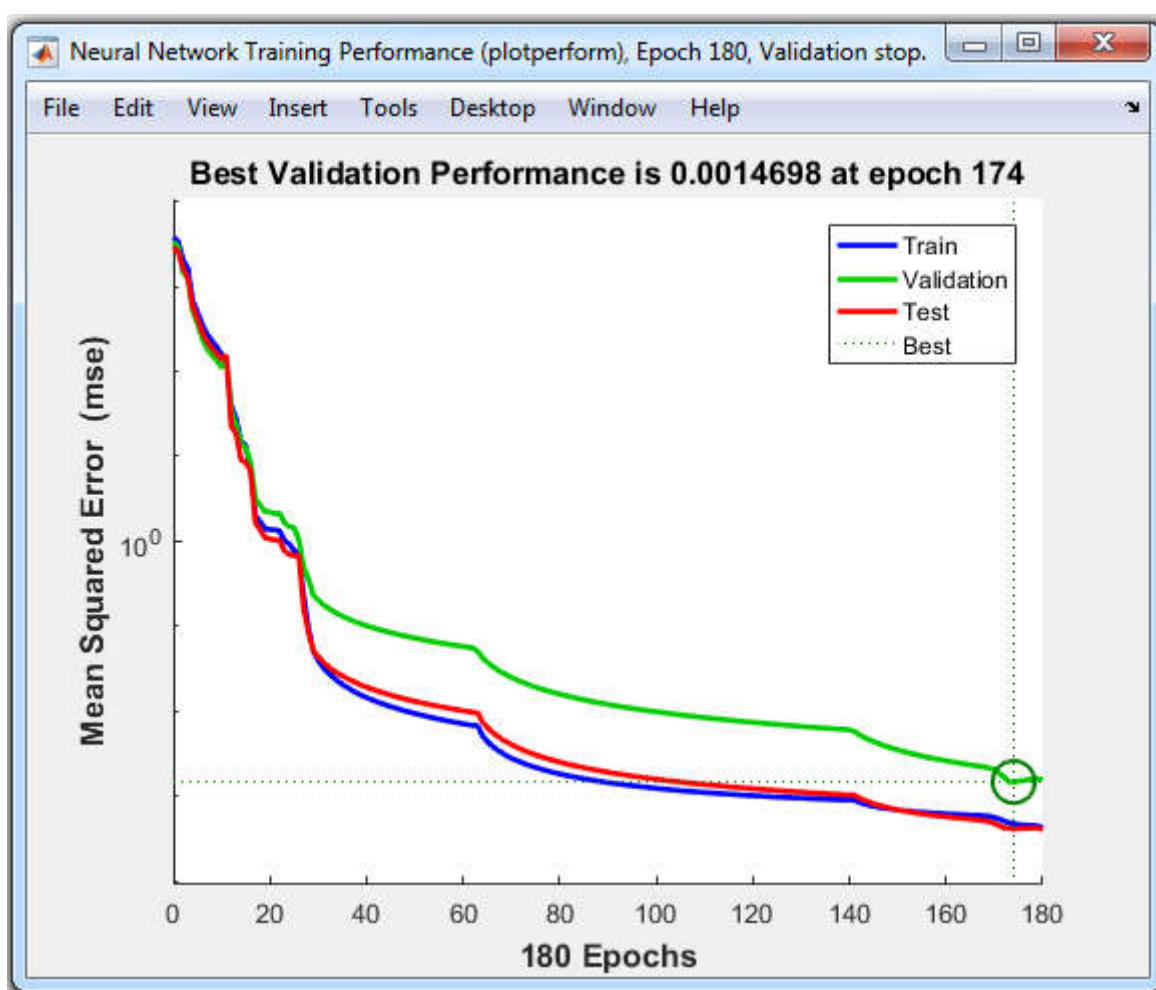


Рисунок 4.15 – Вікно контролю процесу навчання

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.

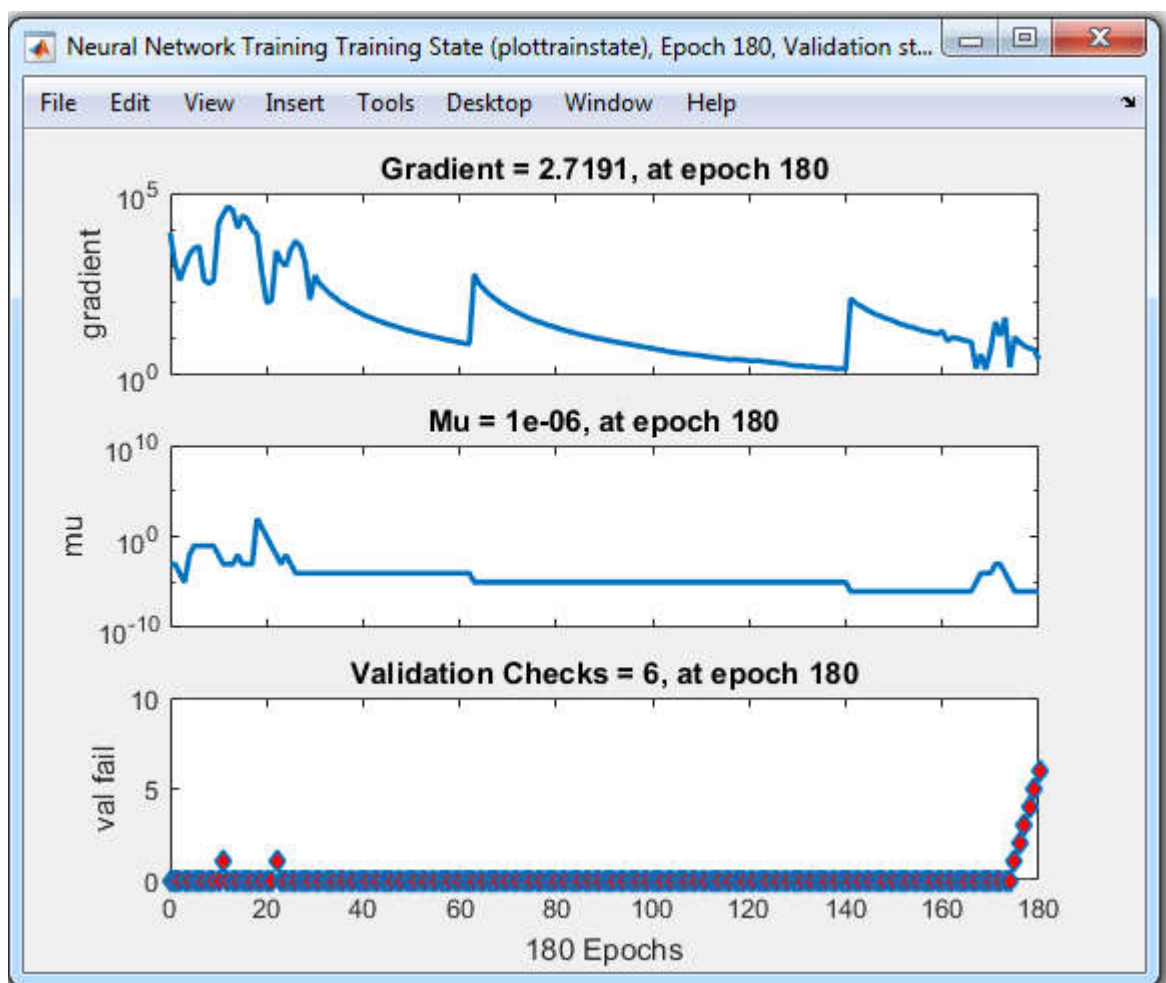


Рисунок 4.16 - Вікно Neural Network Training State

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 регуляризацію.

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

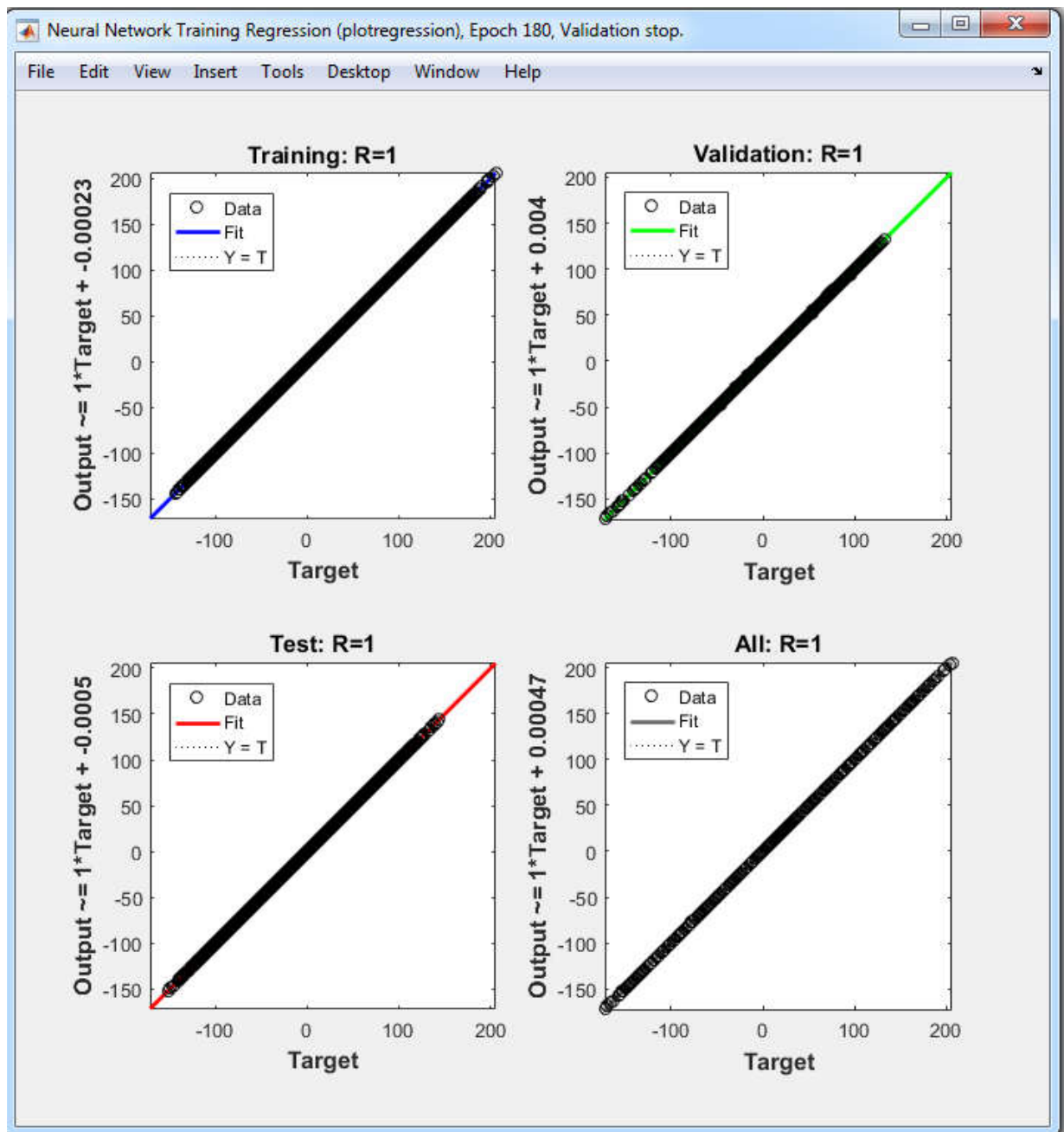


Рисунок 4.17 - Вікно Neural Network Training Regression

Onuc вікна Neural Neural Network Training Regression

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

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

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

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

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

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

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

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

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

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

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

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

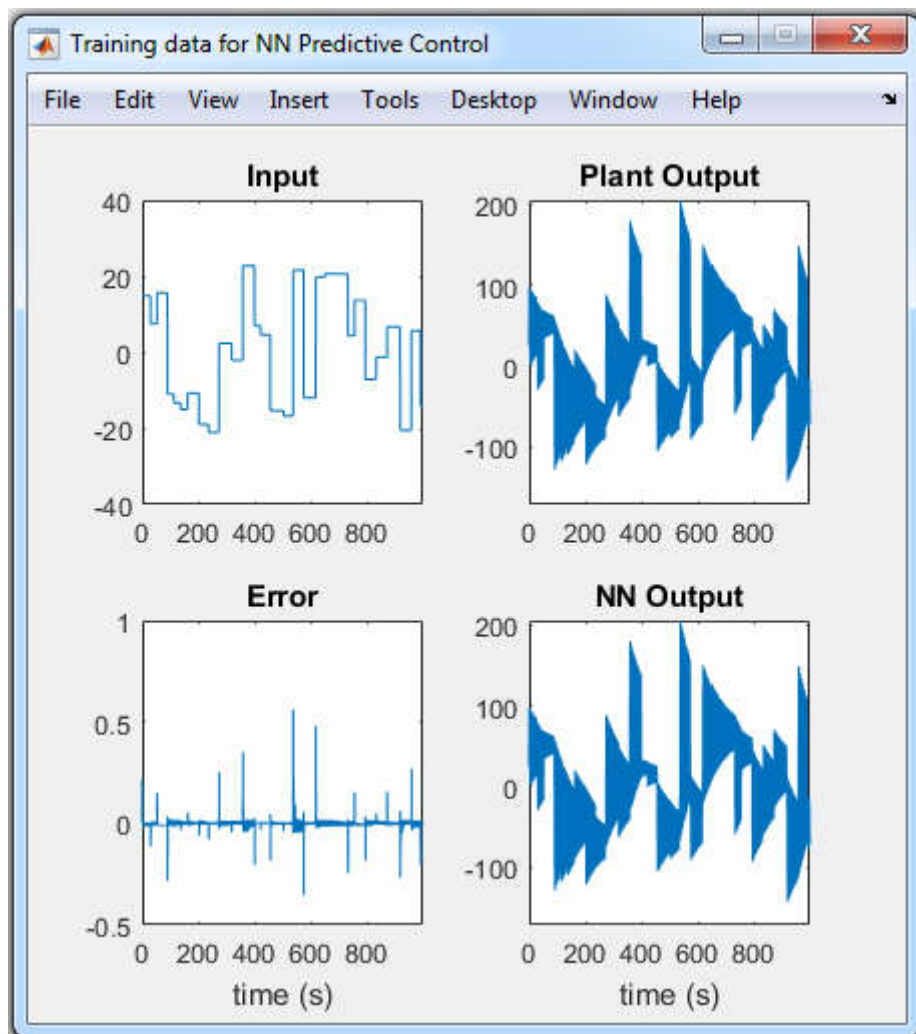


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

динаміку системи під впливом вхідних сигналів.

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

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

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

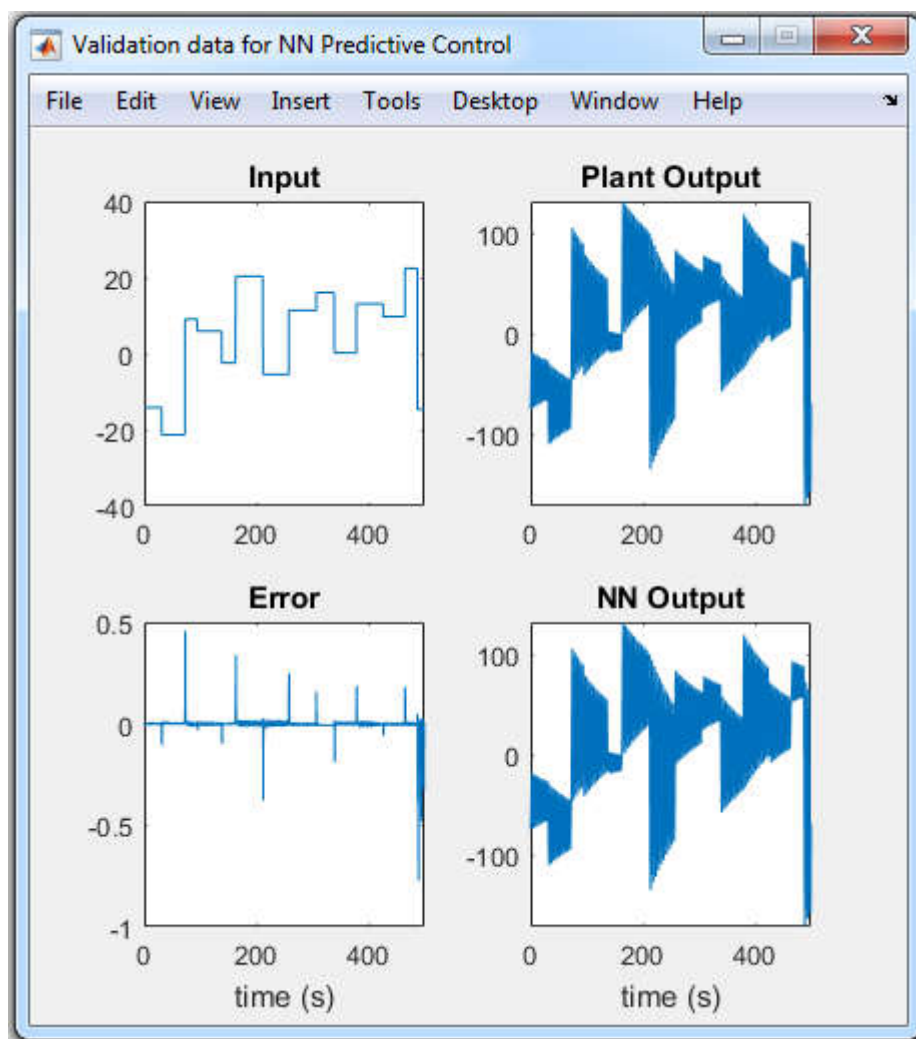


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

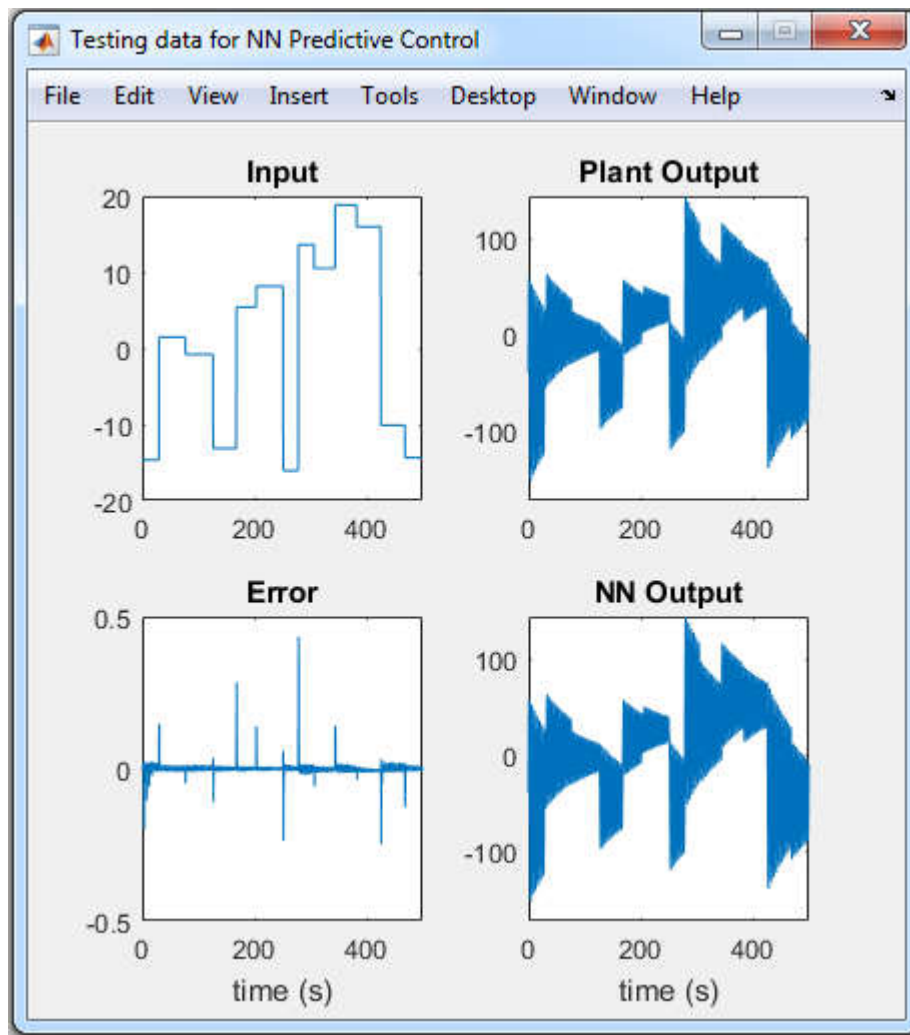


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

Продовження опису вікна Plant Identification.

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

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

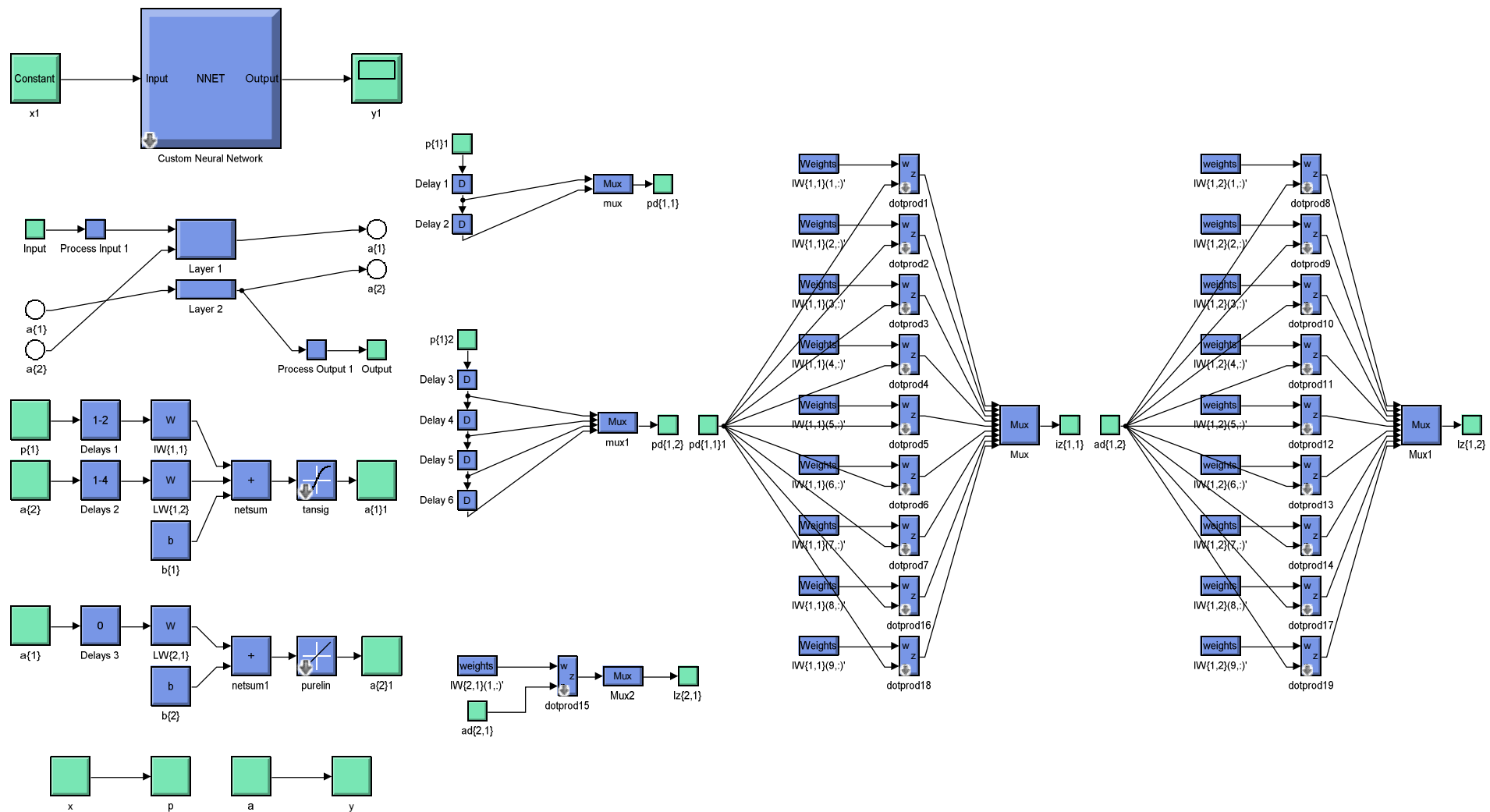


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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

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

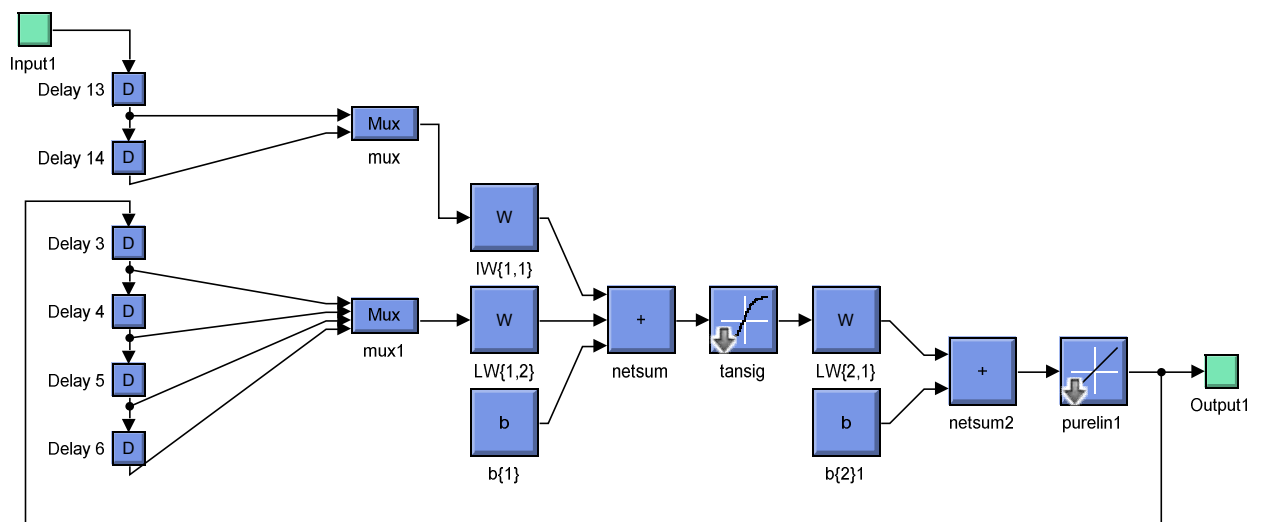


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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

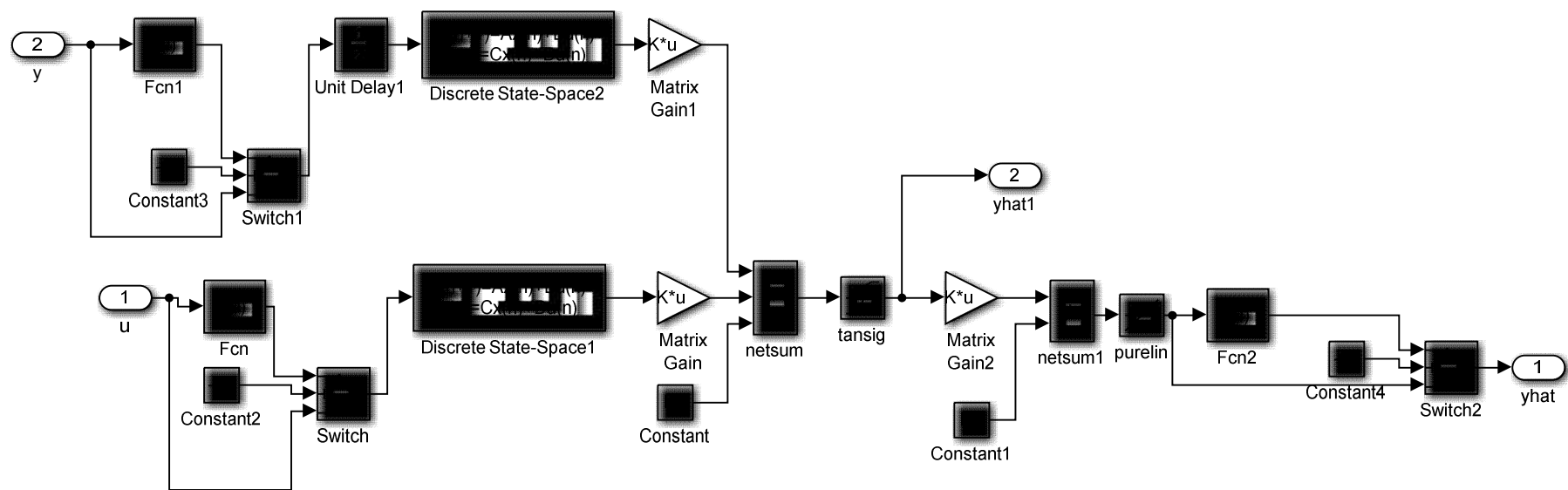


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

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

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

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

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

Discrete State Space 1

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

Discrete State Space 2

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

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

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

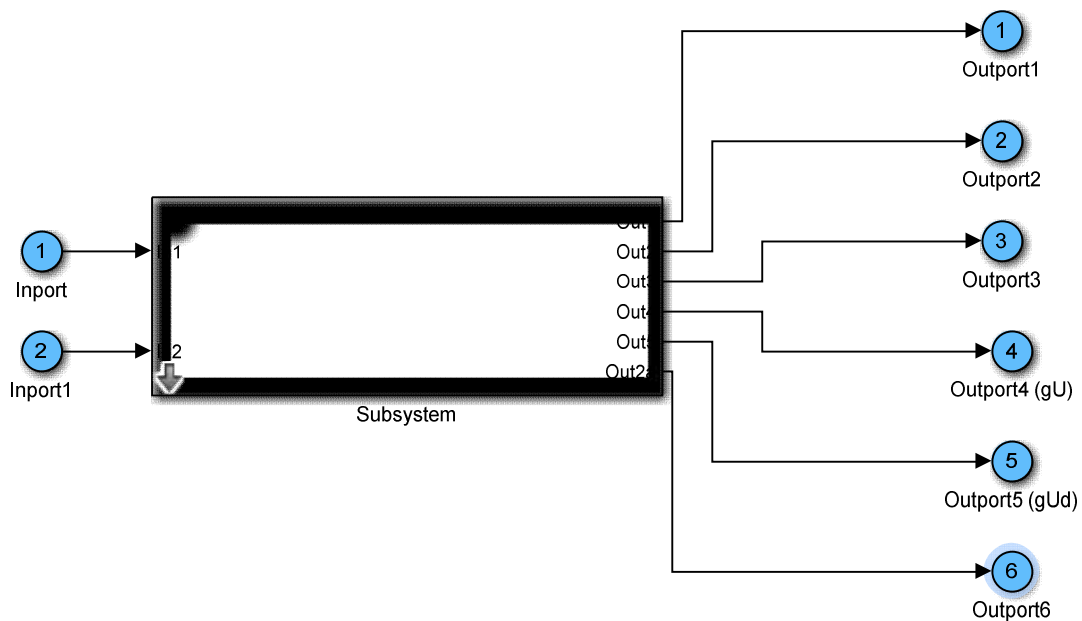


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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

Продовження опису вікна Neural Network Predictive Controller

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

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

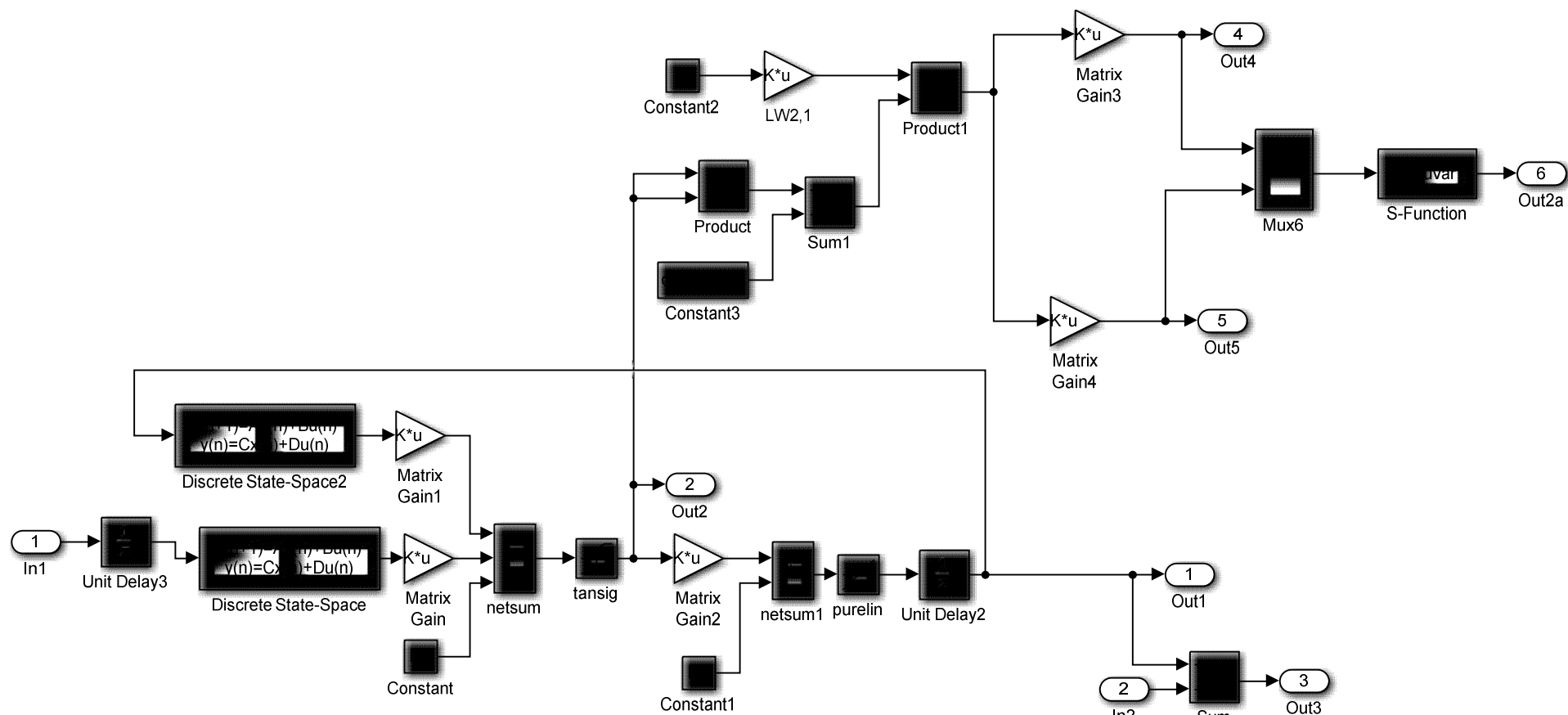


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

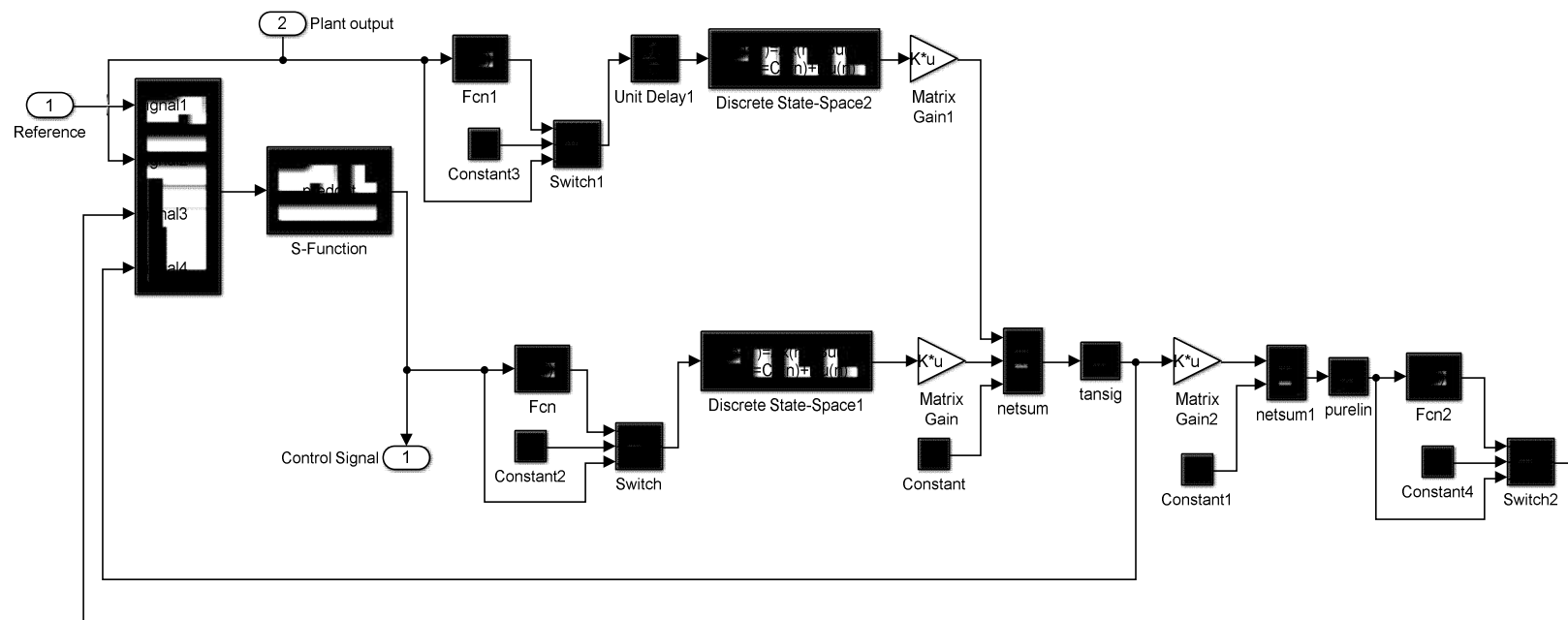


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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

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

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

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

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

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

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

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

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

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

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

Minimization Routine. Вибір процедури одновимірного пошуку.

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

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

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

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

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

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

4.4 Вибір параметрів нейрорегулятора NN PREDICTIVE CONTROLLER

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

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

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

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

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

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

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

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

Як навчальну функцію використано **trainlm**.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller

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

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

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

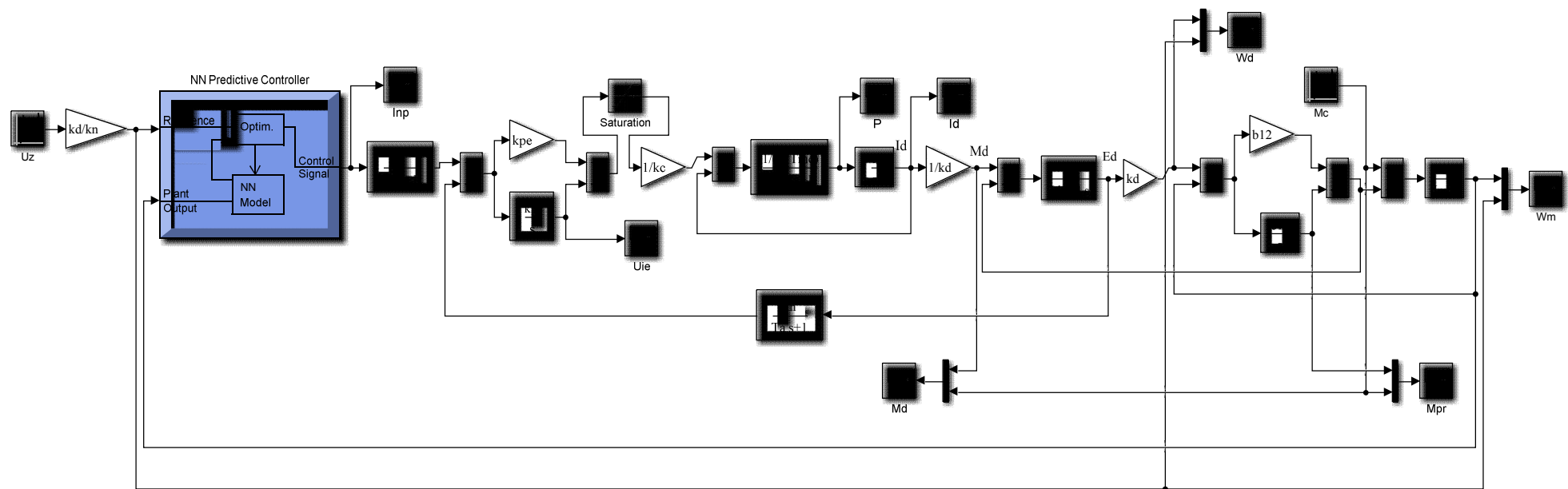


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

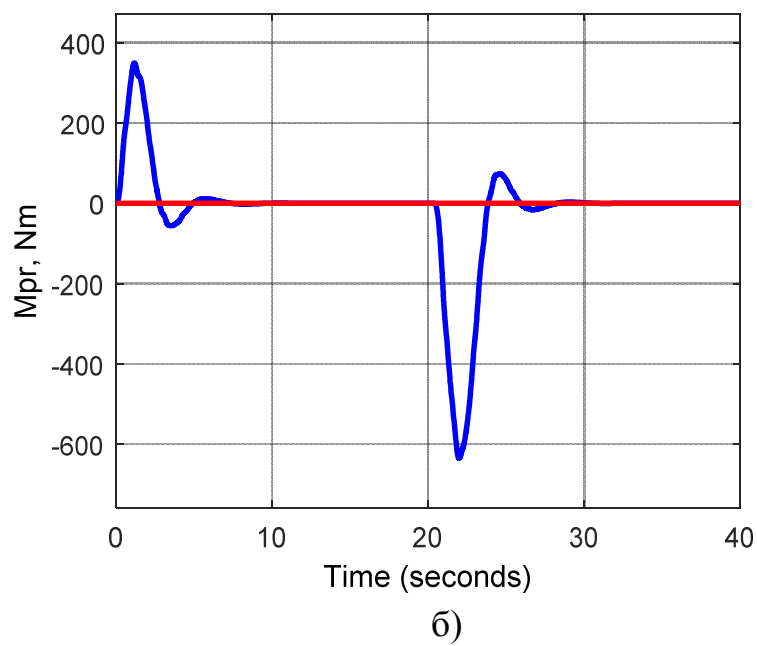
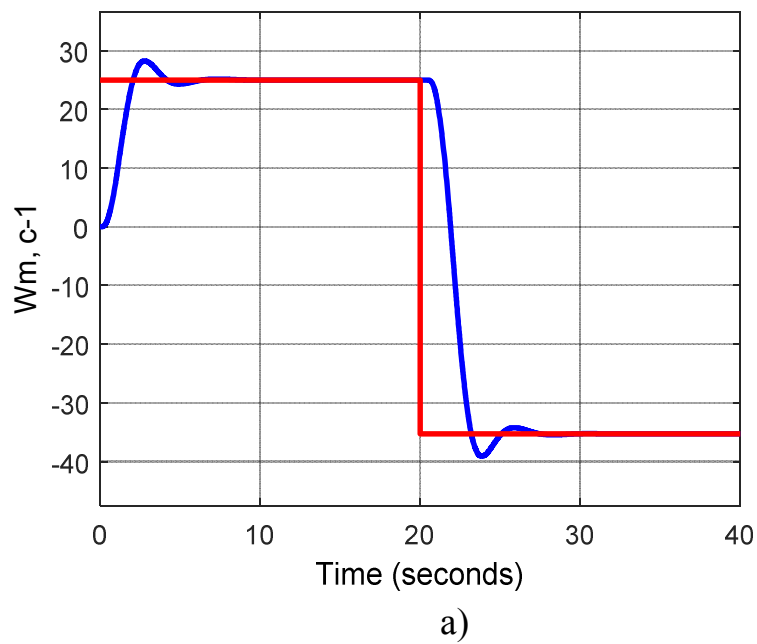


Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

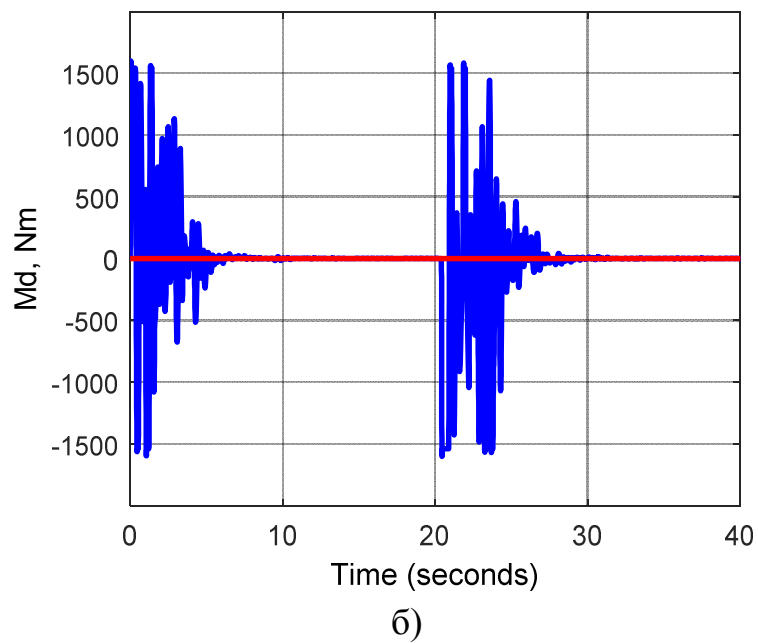
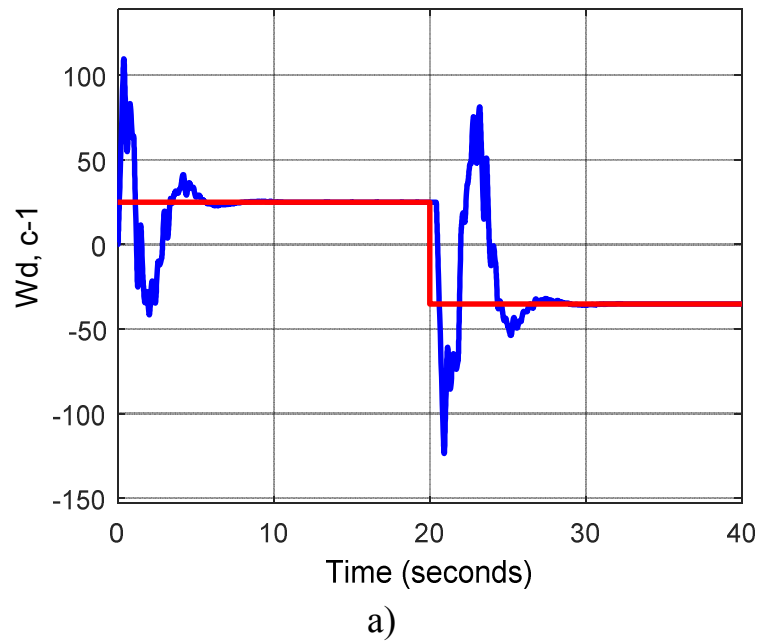


Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

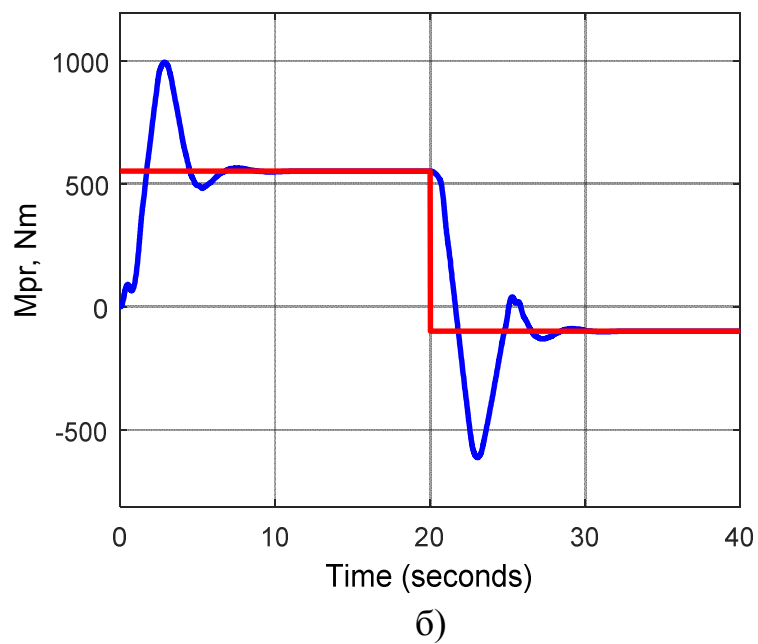
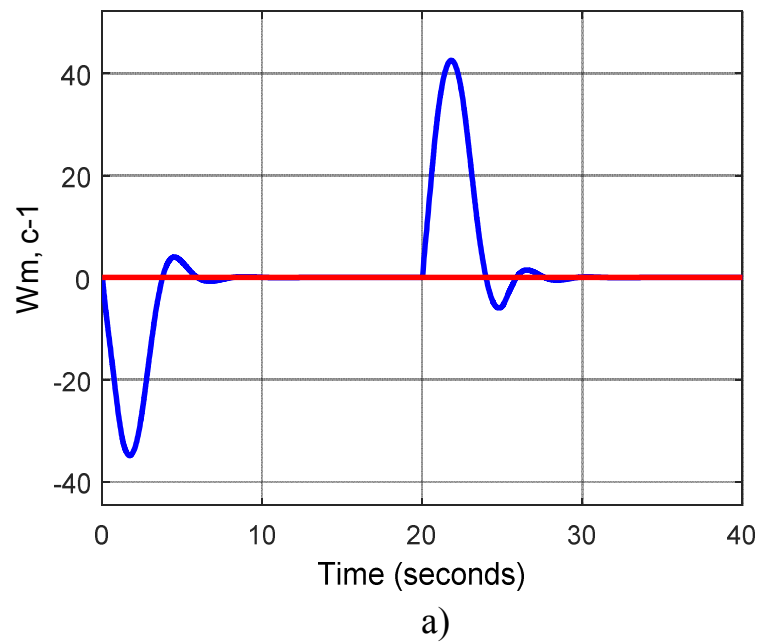


Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

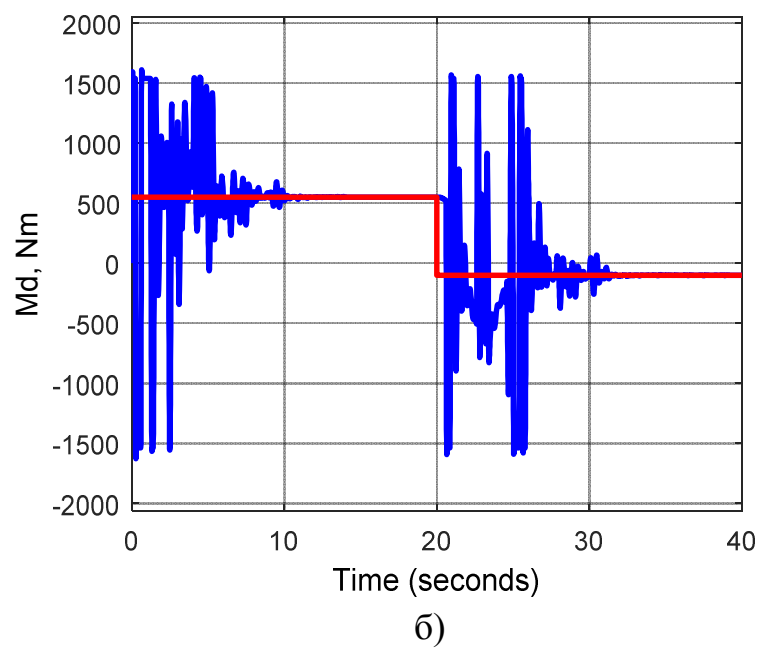
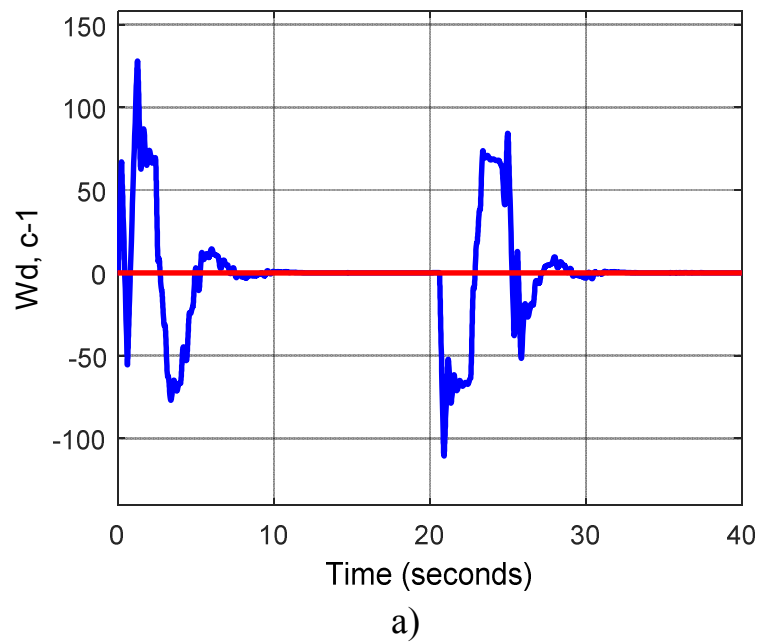


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

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

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

3. Шляхом варіювання параметрів навчальної послідовності, з використанням моделі виконавчого пристрою, виконана генерація навчальної послідовності, яка дозволила виконати тренування мережі з високим ступенем точності. Оптимальні параметри навчальної послідовності наступні: довжина навчальної вибірки 10000; інтервал між двома послідовними моментами знімання даних 0,2с; мінімальний інтервал ідентифікації 20 с; максимальний інтервал ідентифікації 50 с.

4. Створено двошарову нейронну мережу з прямою передачею сигналу і виконано навчання мережі з використанням алгоритму Левенберга-Марквардта. Визначено оптимальну кількість нейронів прихованого шару. Помилка навчання мережі має порядок 10^{-3} , миттєві помилки на навчальній, контрольній і тестовій множині не перевищують 1. Коефіцієнт кореляції R дорівнює 1:

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

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

6. Проведено моделювання нейромережевої системи з синтезованим нейрорегулятором. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи в системі з нейрорегулятором NN Predictive Controller в режимі пуску і набросу навантаження мають задовільний характер. Система є астатичною. Перерегулювання складає 8%, час регулювання складає 6 с, що відповідає завданню.

ВИСНОВКИ

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

Основні результати роботи полягають у наступному.

1. В результаті аналізу систем управління основних механізмів мостового крана, що забезпечують високу якість регулювання і з урахуванням вимог, що пред'являються до сучасних систем управління, науково обґрунтовано перспективність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму пересування моста мостового крана.

2. На підставі аналізу технічних характеристик мостових кранів і вимог до електроприводів основних механізмів вибраний електропривод постійного струму, виконаний за системою ТП-Д. Виконано вибір електродвигуна і основного електроустаткування силового ланцюга електроприводу.

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

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

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

5. Для додання системі бажаних динамічних характеристик виконаний синтез системи управління з нейрорегулятором NN Predictive Controller, реалізованим в пакеті прикладних програм Neural Network Toolbox системи MATLAB. Розглянутий принцип побудови і функціонування нейромережевого регулятора, приведений порядок синтезу регулятора, побудовані процеси в двомасовій системі з нейрорегулятором. Аналіз результатів моделювання показав, що реакція системи на ступінчасту дію з випадковою амплітудою цілком задовільна, має коливальний характер з достатньо швидким загасанням. Система є астатичною. Перерегулювання складає 8%, час регулювання складає 6 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. Василюк, 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. Василюк, 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 the Ideas Immanent in Nervous Activity // Bulletin of 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 prac-

tice a survey // Automatica. – 1989. – Vol.25. – P.335-348.

41. Kandel, Eric R. *In Search of Memory: The Emergence of a New Science of Mind*. W.W. Norton & Company, 2006.

42. Li Y., Sundararajan N., Saratchandran P. (2001), Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks, Automatica, Vol. 37, No 8, pp. 1293-1301.

43. Gundy-Burlet K., Krishnakumar K., Limes G., Bryant D. (2004), Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft, J. of Aerospace Computing, Information, and Communication, Vol. 1, No.12, pp. 526-542.

44. Kondratev A. I., Tyumentsev Yu. V. (2010), “Neyrosetevoe adaptivnoe otkazoustoychivoe upravlenie dvizheniem manevrennogo samoleta”, [Neural network adaptive fault-tolerant control of the movement of a maneuverable aircraft], XII All-Russian Scientific and Technical Conference "Neuroinformatics – 2010", Part 2, NIYaU MIFI, Moscow, pp. 262-273.

45. Nikiforova L. N., Petrosyan E. A., Yakemenko G. V. (2000), “Neyrokompyuteryi v upravlenii vertoletami”, [Neurocomputers in helicopter control], Artificial Intelligence, No. 3, pp. 290-298.

46. Kupin A. I. (2008), “Intelektualna Identifikatsiya ta keruvannya v umovah protsesiv zbagachuvalnoyi tehnologiyi”, [Intelligent identification and control in the processes of enrichment technology], KTU, Krivoy Rog, 204 p.

47. D. Gu and H. Hu. (2002), Neural Predictive Control for a Car-like Mobile Robot, International Journal of Robotics and Autonomous Systems, Vol. 39 (2), pp. 73-86.

48. Danil V. Prokhorov. (2008), Toyota Prius HEV Neurocontrol and Diagnostics, Neural Networks, No. 21, pp. 458-465.

49. Zmeu K. V., Markov N. A., Shipitko I. A., Notkin B. S. (2009), “Bezmodelnoe prognoziruyshee inversnoe neyro-upravlenie s regeneriruemym etalonnyim perehodnyim protsessom”, [A modelless predictive inverse neuropravlenie with a regenerated reference transient process], Intelligent Systems, No. 3, pp. 109-117.

50. Kuznetsov B. I. , Vasilets T. Yu., Varfolomiev O. O. (2015), “Neyromerezheva

система наведення і стабілізації з регулятором на основі еталонної моделі”, [Neural network aiming and stabilization system with the reference model controller], Electrical engineering and electromechanics, No. 4, pp.35-39.

51. Dzyuba D. A., Chernodub A. N. (2010), “Primenenie metoda kontroliruemogo vozmuscheniya dlya modifikatsii neyrokontrollerov v realnom vremeni”, [The application of the controlled disturbance method for the modification of real time neural controllers], Mathematical machines and systems, No. 4, pp. 20-28.

52. Dias F.M., Mota A.M. (2001), Comparison between Different Control Strategies using Neural Networks, 9th Mediterranean Conference on Control and Automation, Croatia, Dubrovnik.

53. Venayagamoorthy G.K., Harley R.G., Wunsch D.C. (2003), Implementation of Adaptive Criticbased Neurocontrollers for Turbogenerators in a Multimachine Power System”, IEEE Transactions on Neural Networks, Vol. 14, Issue 5, pp. 1047–1064.

54. D’Emilia G., Marrab A., Natalea E. (2007), Use of neural networks for quick and accurate autotuning of PID controller, Robotics and Computer-Integrated Manufacturing, Vol. 23, pp. 170 – 179.

55. Яуре А.Г., Певзнер Е.М. Крановый электропривод: Справочник. – М.: Энергоатомиздат, 1988. – 314 с.

56. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.

57. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.

58. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.

59. 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.

60. 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.

61. 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.

62. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.

63. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.

64. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.

65. Автоматизований електропривод ч. 2 [Електронний ресурс]: навчальний посібник для студентів освітньої програми «Електромеханічні системи автоматизації, електропривод та електромобільність» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / В.І. Теряєв. – Київ: КПІ ім. Ігоря Сікорського, 2022.– 204 с

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

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

68. 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.

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

70. 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.
71. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
72. 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.
73. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 с.

ДОДАТОК А

СТРУКТУРА ШТУЧНОГО НЕЙРОНА

Актуальність досліджень штучних нейронних мереж підтверджується різноманіттям їх можливих застосувань. Теорія штучних нейронних мереж є алгоритмічним базисом розвитку нейрокомп'ютерів, подібно до тому, яка булева алгебра більше 50 років була алгоритмічним базисом однопроцесорних і багатопроцесорних ЕОМ.

Сучасні цифрові обчислювальні машини перевершують людину по здатності проводити числові і символічні обчислення. Проте людина може без зусиль вирішувати складні задачі сприйняття зовнішніх даних (наприклад, дізнатися в натовпі знайомого тільки по його обличчю, що промайнуло) з такою швидкістю і точністю, що найпотужніший в світі комп'ютер в порівнянні з ним виявиться безнадійним тугодумом. Причина такої значної відмінності в їх продуктивності заключається в тому, що архітектура біологічної нейронної системи абсолютно не схожа на архітектуру машини фон Неймана (табл. А.1), а це істотно впливає на типи функцій, які ефективніше використовуються кожною з моделей.

Нейронна мережа – це мережа з кінцевим числом шарів з однотипних елементів – аналогів нейронів з різними типами зв'язків між шарами. При цьому число нейронів в шарах вибирається виходячи з необхідності забезпечення заданої якості рішення задачі, а число шарів нейронів – як можна менше для скорочення часу рішення задачі.

Розвиток штучних нейронних мереж надихався і надихається біологією. Проте наші знання про роботу мозку обмежені, і розробникам штучних ШНМ доводиться виходити за межі сучасних біологічних знань у пошуках структур, що дозволяють виконувати корисні функції. Так часто створюються мережі, що неможливі в живій матерії або такі, що вимагають неправдоподібно великих допущень про анатомію і функціонування мозку.

Нервова система людини, побудована з елементів, званих нейронами, дуже складна. Близько 10^{11} нейрона беруть участь в приблизно 10^{15} передаючих

зв'язках. Унікальними здібностями нейронів є прийом, обробка і передача електрохімічних сигналів по нервових шляхах, які утворюють комунікаційну систему мозку.

Таблиця А.1 – Машина фон Неймана в порівнянні з біологічною нейронною системою

Параметри порівняння	Машина фон Неймана	Біологічна нейронна система
	1	2
Процесор	Складний	Простий
	Високошвидкісний	Низькошвидкісний
	Один або кілька	Велика кількість
Пам'ять	Відокремлена від процесора	Інтегрована в процесор
	Локалізована	Розподілена
	Адресація не за змістом	Адресація за змістом
Надійність	Висока уразливість	Живучість
Обчислення	Централізовані	Розподілені
	Послідовні	Паралельні
	Програми, що зберігаються	Самонавчання
Спеціалізація	Чисельні і символічні операції	Проблеми сприйняття
Середовище функціонування	Строго визначена	Погано визначена
	Строго обмежена	Без обмежень

Не дивлячись на величезну кількість нейронів, їх тіла займають всього декілька відсотків загального об'єму мозку. Решта майже всього простору зайнята міжнейронними зв'язками. Число зв'язків кожного нейрона не має аналогів в сучасній техніці. Розуміння того, що міжнейронні зв'язки відносяться до основних структурних компонентів мозку, що в першу чергу визначає його функціональні характеристики, є одним з найбільш істотних висновків, зроблених нейрофізіологами. В підтвердження можна привести вислів відомого нейрофізіолога Э. Кендела: «На переконання багатьох нейробіологів врешті-решт буде

доведено, що унікальні властивості кожної людини – здатність відчувати, думати, навчатися і пам'ятати – поміщені в строго організованих мережах синаптичних взаємозв'язків між нейронами головного мозку» [41].

На рис. А.1 приведена структура типового біологічного нейрона. Дендрити йдуть від тіла нервової клітини до інших нейронів, де вони приймають сигнали в точках з'єднання, званих синапсами. Прийняті синапсом вхідні сигнали підводяться до тіла нейрона. Тут вони підсумовуються, причому одні входи прагнуть збудити нейрон, а інші перешкодити його збудженню. Відповідно до цього розрізняють процеси синаптичного збудження і гальмування нейрона.

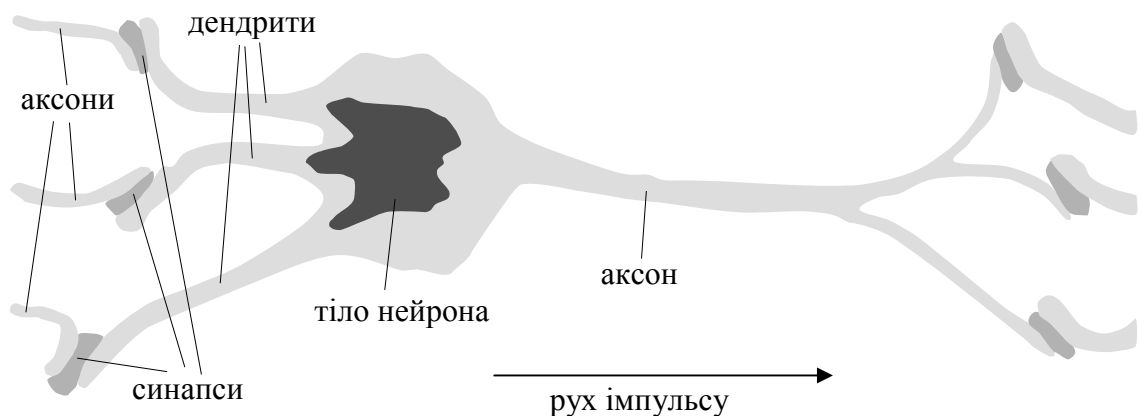


Рисунок А.1 – Біологічний нейрон

Коли сумарне збудження в тілі нейрона перевищує деякий поріг, нейрон збуджується, посылаючи по аксону, який на кінці сильно гілкується, сигнал іншим нейронам. Нервовий імпульс володіє здатністю розповсюджуватися по нервових волокнах без загасання, аж до синаптичних закінчень із швидкістю розповсюдження біля 10 м/с .

Синаптичне гальмування знижує збудження нейрона і тим самим утрудняє перехід збудження нейрона в нервовий імпульс.

Після генерації імпульсу настає так званий рефрактерний період, під час якого нейрон відновлює здібність до генерації наступного імпульсу. Тривалість рефрактерного періоду менш 1 мс . Тому максимальна частота генерації імпульсів нейронів біля 1000 с^{-1} .

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. Він володіє групою синапсів – однонаправлених вхідних зв'язків, сполучених з виходами інших нейронів, а також має аксон – вихідний зв'язок даного нейрона, з яким сигнал (збудження або гальмування) поступає на синапси наступних нейронів. Кожен вхід множиться на відповідну вагу, аналогічну синаптичній силі, і всі добутки підсумовуються, визначаючи рівень активації нейрона. Загальний вид нейрона, що реалізовує ці властивості, приведений на рис. А.2.

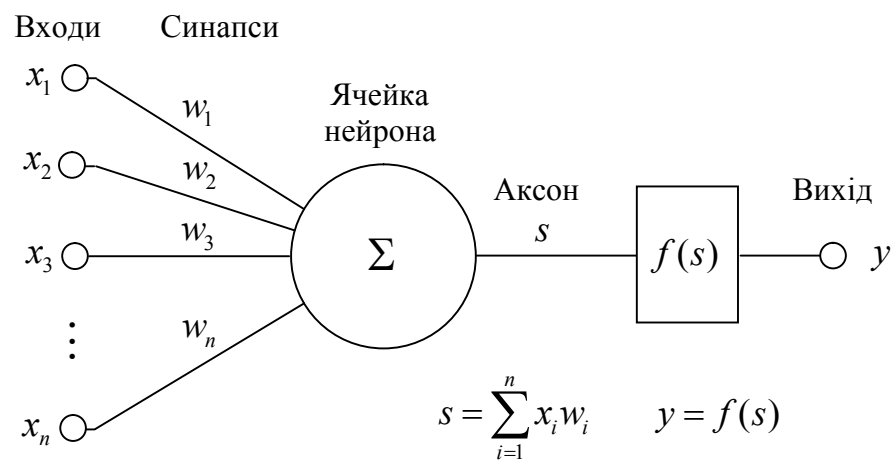


Рисунок А.2 – Штучний нейрон

Множина вхідних сигналів позначається вектором $\mathbf{X}$. Кожна вага w_i відповідає "силі" одного біологічного синаптичного зв'язку. Множина вагів в сукупності позначається вектором $\mathbf{W}$.

Блок, що підсумовує, відповідний тілу біологічного елементу, алгебраїчно складає зважені входи

$$s = \sum_{i=1}^n x_i w_i . \quad (3.1)$$

Вихід нейрона є функцією його стану:

$$y = f(s) \quad (3.2)$$

Нелінійна функція f називається активаційною і може мати різний вигляд, що, зокрема, показано на рис. А.3.

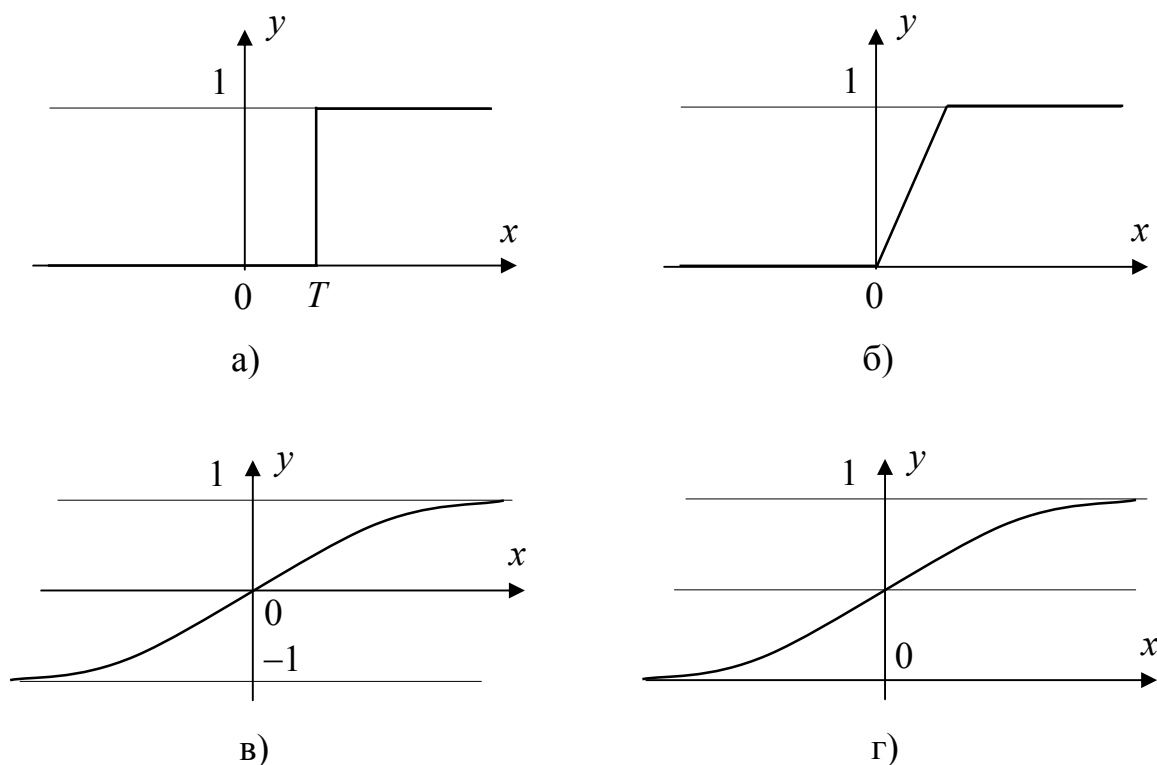


Рисунок А.3 – Типи активаційних функцій: а – функція одиничного стрибка;
 б – лінійний поріг (гістерезис); в – сигмоїд – гіперболічний тангенс;
 г – сигмоїд (формула (1.3))

У разі, коли функція активації одна і та ж для всіх нейронів мережі, мережу називають однорідною (гомогенною). Якщо ж активаційна функція залежить ще від одного або декількох параметрів, значення яких міняються від нейрона до нейрона, то мережу називають неоднорідною (гетерогенною).

Однією з найбільш поширених є нелінійна функція з насиченням, так звана логістична функція або сигмоїд (тобто функція S -подібного вигляду):

$$f(x) = \frac{1}{1 + e^{-\alpha x}}. \quad (\text{A.3})$$

При зменшенні α сигмоїд стає пологішим, в межі при $\alpha = 0$ вироджується в горизонтальну лінію на рівні 0,5. При збільшенні α сигмоїд наближається на вигляд до функції одиничного стрибка з порогом T у точці $x = 0$. З виразу для сигмоїда очевидно, що вихідне значення нейрона лежить в діапазоні $[0,1]$. Одна з цінних властивостей сигмоїдної функції – простий вираз для її похідної, застосування якого буде розглянуто надалі

$$f'(x) = \alpha f(x)(1 - f(x)). \quad (\text{A.4})$$

Слід зазначити, що сигмоїдна функція диференціюється на всій осі абсцис, що широко використовується в багатьох алгоритмах навчання. Крім того, вона володіє властивістю підсилювати слабкі сигнали краще, ніж сильні, і запобігає насиченню від сильних сигналів, оскільки вони відповідають областям аргументів, де сигмоїд має пологий нахил. Іншою широко використовуваною активаційною функцією є гіперболічний тангенс. На відміну від логістичної функції гіперболічний тангенс приймає значення різних знаків, що для ряду мереж виявляється доцільнішим.

Кажучи про можливу класифікацію ШНМ, важливо відзначити існування бінарних і аналогових мереж. Перші з них оперують з двійковими сигналами, і вихід кожного нейрона може приймати тільки два значення: логічний нуль ("загальмований" стан) і логічна одиниця ("збуджений" стан). У аналогових мережах вихідні значення нейронів здатні приймати безперервні значення. Ще одна класифікація ділить ШНМ на синхронні і асинхронні. У першому випадку в кожен момент часу свій стан міняє лише один нейрон. У другому – стан міняється відразу у цілої групи нейронів, як правило, у всього шару. Для програмних імітаторів нейронних мереж на цифрових ЕОМ, питання пов'язані з синхронізацією вирішуються комп'ютером, на якому реалізуються ШНМ.

Розглянута проста модель штучного нейрона істотно спрощує ряд властивостей свого біологічного двійника. Наприклад, вона не приймає до уваги затримки в часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. І, що важливіше, вона не враховує дію синхронізуючої функції біологічного нейрона, яку ряд дослідників вважає вирішальною. Не дивлячись на ці обмеження, мережі, побудовані з цих нейронів, виявляють властивості, що сильно нагадують біологічну систему.

При складанні нейронних мереж (при їх конструкції) строгих обмежень немає. Завдання полягає лише в тому, щоб на виходах формувалися необхідні сигнали. Можливості в цьому питанні безмежні, але зазвичай використовують декілька стандартних архітектур, з яких при деяких невеликих модифікаціях будують більшість використовуваних мереж.

ДОДАТОК Б

АРХІТЕКТУРА ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Б.1 Архітектури найбільш поширених ШНМ

Хоча один нейрон і здатний виконувати прості процедури розпізнавання, сила нейронних обчислень виникає від з'єднань нейронів в мережах.

Широкий круг завдань, вирішуваний ШНМ, не дозволяє в даний час створювати універсальні, потужні мережі, вимушуючи розробляти спеціалізовані ШНМ, що функціонують по різних алгоритмах.

Вибір структури ШНМ здійснюється відповідно до особливостей і складності завдання. Якщо ж завдання не може бути зведене ні до одного з відомих типів, розробникові доводиться вирішувати складну проблему синтезу нової конфігурації. При цьому він керується декількома основоположними принципами: можливості мережі зростають із збільшенням числа осередків мережі, щільності зв'язків між ними і числом виділених шарів; введення зворотних зв'язків разом із збільшенням можливостей мережі піднімає питання про динамічну стійкість мережі; складність алгоритмів функціонування мережі (зокрема, наприклад, введення декількох типів синапсів – збудливих, гальмуючих і ін.) також сприяє посиленню потужності ШНМ. Питання про необхідні і достатні властивості мережі для вирішення того або іншого роду завдань є цілим напрямом нейрокомп'ютерної науки. Оскільки проблема синтезу ШНМ сильно залежить від вирішуваної задачі, дати загальні докладні рекомендації складно. В більшості випадків оптимальний варіант вибирається на основі інтуїтивного підбору. Єдина жорстка вимога, що пред'являється архітектурою до елементів мережі, це відповідність розмірності вектора вхідних сигналів мережі числу її вхідів.

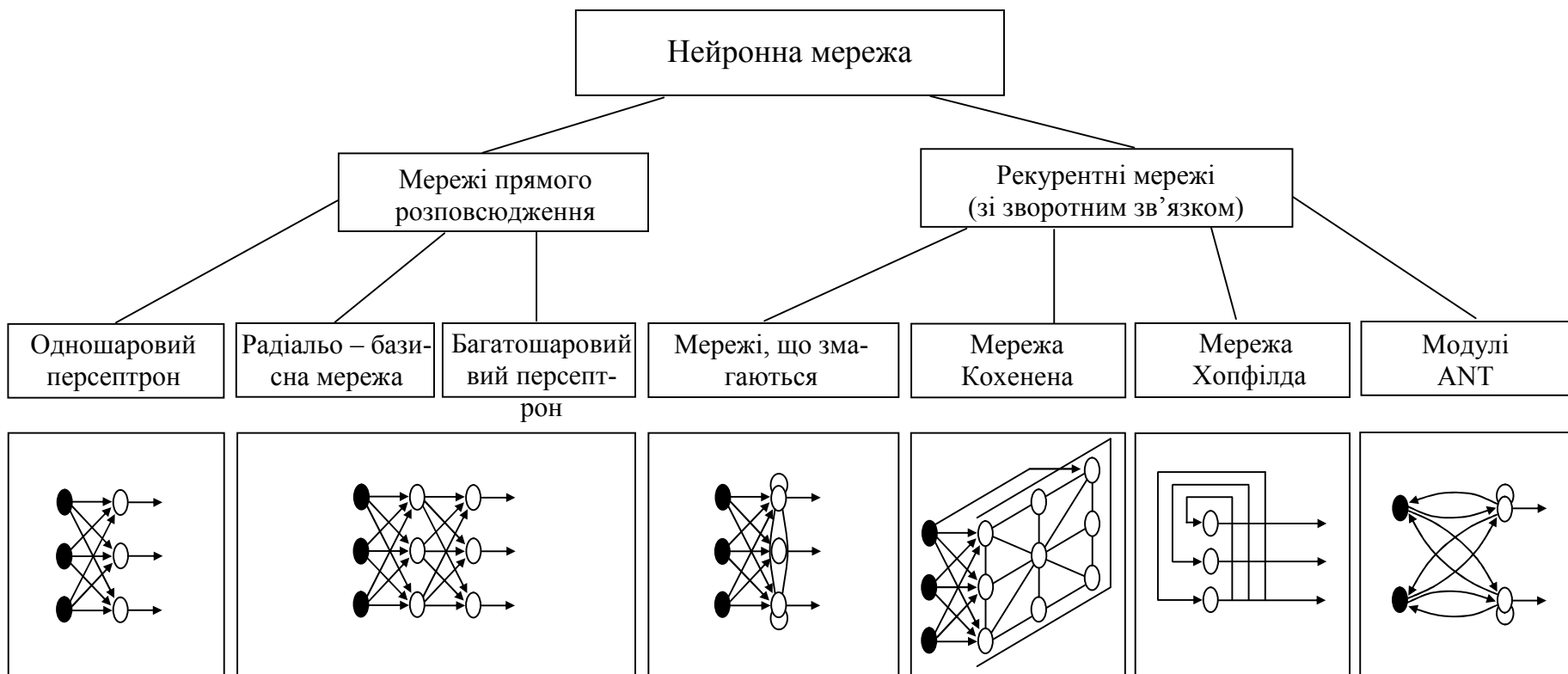


Рисунок Б.1 – Варіанти найбільш поширених архітектур нейронних мереж

Б.2 Штучні нейронні мережі прямого розповсюдження

Найбільшого застосування при побудові нейрорегуляторів систем автоматичного управління знайшли мережі прямого розповсюдження без зворотних зв'язків (багатошарові персептрони). Розглянемо їх більш детально.

Проста одношарова мережа, що складається з m нейронів, показана на рис. Б.2.

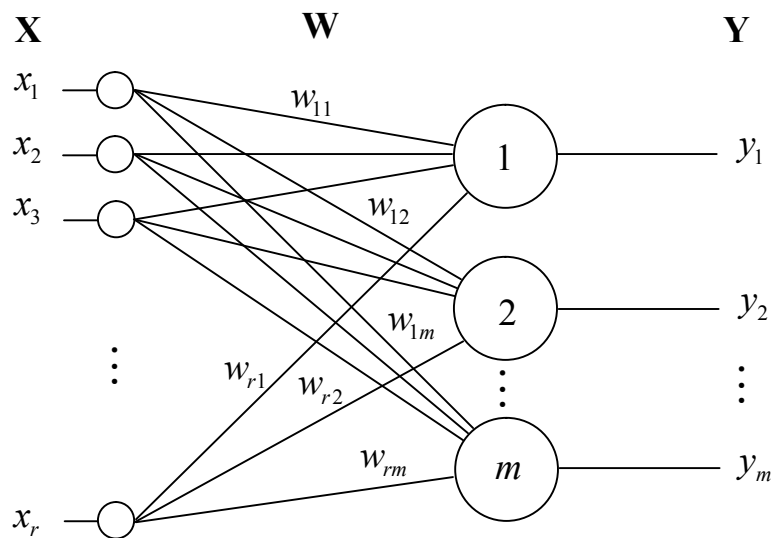


Рисунок Б.2 – Одношарова нейронна мережа

На r входів поступають сигнали, що проходять по синапсах на m нейрони, що створюють єдиний шар цієї ШНМ, які видають вихідні сигнали:

$$y_j = f \left[\sum_{i=1}^r x_i w_{ij} \right], \quad (\text{Б.1})$$

де $j = 1, 2, \dots, m$.

У штучних і природних (біологічних) мережах багато з'єднань можуть бути відсутніми, всі з'єднання показані в цілях спільності.

Очевидно, що всі вагові коефіцієнти синапсів одного шару нейронів можна звести в матрицю **W**, у якій кожен елемент w_{ij} задає величину i -го синапти-

чного зв'язку j -го нейрона. Таким чином процес, що відбувається в ШНМ, може бути записаний в матричній формі:

$$\mathbf{Y} = f(\mathbf{XW}) \quad (\text{Б.2})$$

де $\mathbf{X}$ і $\mathbf{Y}$ – відповідно вхідний і вихідний сигнальні вектори;

$f(\mathbf{V})$ – активаційна функція, вживана поелементно до компонентів вектора $\mathbf{V} = \mathbf{XW}$.

На рис. Б.3 представлена двошарова ШНМ, отримана з одношарової (див. рис. Б.2) шляхом додавання другого шару, що складається з l нейронів. Тут доречно відзначити важливість ролі розглянутої раніше нелінійності активаційної функції, оскільки, якби вона не володіла даною властивістю або не входила в алгоритм роботи кожного нейрона, результат функціонування будь-якої k -шарової ШНМ з ваговими матрицями $\mathbf{W}^{(i)}$, де $i = 1, 2, \dots, k$ для кожного шару i зводився б до перемножування вхідного вектора сигналів $\mathbf{X}$ на матрицю

$$\mathbf{W}^{(\Sigma)} = \mathbf{W}^{(1)}\mathbf{W}^{(2)} \dots \mathbf{W}^{(k)}, \quad (\text{Б.3})$$

тобто така k -шарова еквівалентна одношаровій ШНМ з ваговою матрицею єдиного шару $\mathbf{W}^{(\Sigma)}$:

$$\mathbf{X} = \mathbf{YW}^{(\Sigma)}. \quad (\text{Б.4})$$

Таким чином, для розширення обчислювальних можливостей багатошарових ШНМ в порівнянні з одношаровими ШНМ необхідно використання нелінійних активаційних функцій.

Необхідно відзначити, що нелінійність іноді може вводитися і в синаптичні зв'язки. У більшості відомих на сьогоднішній день ШНМ для знаходження зваженої суми входів нейрона використовують формулу (Б.1), проте в деяких додатках ШНМ корисно ввести інший запис, наприклад:

$$S = \sum_{i=1}^r x_i^2 w_{ij} \quad (\text{Б.9})$$

або

$$S = \sum_{i=1}^r x_i x_{(i+1) \bmod r} w_{ij} . \quad (\text{Б.10})$$

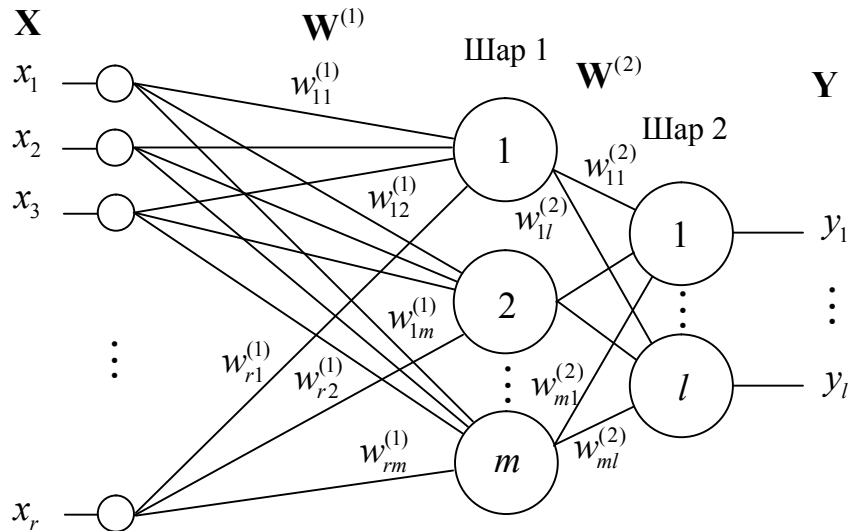


Рисунок Б.3 – Двошарова нейронна мережа

Питання в тому, щоб розробник ШНМ чітко розумів, для чого він це робить, якими цінними властивостями він тим самим додатково наділяє нейрон, і яких позбавляє. Введення такого роду нелінійностей, взагалі кажучи, збільшує обчислювальну потужність мережі, тобто дозволяє з меншого числа нейронів з "нелінійними" синапсами сконструювати ШНМ, що виконує роботу звичайної ШНМ з великим числом стандартних нейронів і складнішою конфігурацією.

У мереж, розглянутих до цих пір, не було зворотних зв'язків, тобто з'єднань, що йдуть від виходів деякого шару до входів цього ж шару або попередніх шарів. Цей спеціальний клас мереж, званих мережами без зворотних зв'язків або мережами прямого розповсюдження, представляє інтерес і широко використовується. В даній магістерській роботі використовується саме цей тип нейронних мереж.

Декілька слів необхідно сказати про необхідну потужність вихідного шару мережі, що виконує остаточну класифікацію простору станів. Річ у тому, що для розділення множини вхідних образів, наприклад, по двох класах достатньо

всього одного виходу. При цьому кожен логічний рівень – "1" і "0" – позначатиме окремий клас. На двох виходах можна закодувати вже чотири класи і т.д. Проте результати роботи мережі, організованої таким чином, недостатньо надійні. Для підвищення достовірності класифікації бажано ввести надмірність шляхом виділення кожному класу одного нейрона у вихідному шарі або, що ще краще, декілька, кожний з яких навчається визначати приналежність конкретних станів входів до певного класу з своєю мірою достовірності, наприклад високої, середньої і низької. Такі ШНМ. дозволяють проводити класифікацію входних неявно виражених станів, об'єднаних в нечіткі (розмиті або пересічні) множини. Ця властивість дозволяє широко використовувати ШНМ в практичних застосуваннях.

ДОДАТОК В

НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

В.1 Загальні питання навчання нейронних мереж

Коли йде розмова про використання ШНМ і нейромережових алгоритмів, майже завжди розуміються певні процедури їх навчання. ШНМ є адаптивною системою, життєвий цикл якої складається з двох незалежних фаз – навчання і роботи мережі. Навчання вважається закінченим, коли мережа правильно виконує перетворення на тестових прикладах і подальше навчання не викликає значної зміни вагових коефіцієнтів, що настроюються. Далі мережа виконує перетворення раніше невідомих їй даних на основі сформованої нею в процесі навчання нелінійної моделі процесу. Мережа успішно працює до тих пір, поки істотно не зміниться реальна модель явища (наприклад, у разі виникнення ситуації, інформація про яку ніколи не пред'являлася мережі при навчанні), що відображається. Після цього мережа може дообучатися з урахуванням нової інформації, причому при дообучанні попередня інформація не втрачається, а узагальнюється з урахуванням інформації, що знов поступила. При "пошкодженні" частини вагових коефіцієнтів ШНМ її властивості можуть бути повністю відновлені в процесі дообучання. Від того, наскільки якісно буде виконаний етап навчання ШНМ, залежить здатність мережі вирішувати поставлені перед нею проблеми під час експлуатації. Теорія навчання розглядає три фундаментальні властивості, пов'язаних з навчанням по прикладах: місткість, складність зразків і обчислювальна складність. Під місткістю розуміється, скільки зразків може запам'ятати мережу і які функції і межі ухвалення рішень можуть бути на ній сформовані. Складність зразків визначає число навчальних прикладів, необхідних для досягнення здатності мережі до узагальнення. Важливою характеристикою є час, що витрачається на навчання. Як правило, час навчання і якість навчання зв'язані зворотною залежністю і вибирати ці параметри доводиться на основі компромісу.

Існують три парадигми навчання "з вчителем", "без вчителя" (самонавчання) і змішана.

У свою чергу множина різних алгоритмів навчання діляться на два великі класи детерміністських і стохастичних алгоритмів. В першому з них підстроювання вагів є жорсткою послідовністю дій, в другому – воно проводиться на основі дій, що підкоряються деякому випадковому процесу

Навчання з вчителем припускає, що для кожного вхідного вектора існує цільовий вектор, що є необхідним виходом. Разом вони називаються навчальною парою. Зазвичай мережа навчається на деякому числі таких навчальних пар.

Коли в мережі тільки один шар, алгоритм її навчання з вчителем досить очевидний, оскільки правильні вихідні стани нейронів єдиного шару заздалегідь відомі, і підстроювання синаптичних зв'язків йде в напрямі, що мінімізує помилку на виході мережі.

У багатошарових же мережах оптимальні вихідні значення нейронів всіх шарів, окрім останнього, як правило, не відомі, і двох або більш шарову ШНМ вже неможливо навчити, керуючись тільки величинами помилок на виходах цієї мережі. Один з варіантів вирішення цієї проблеми – розробка наборів вихідних сигналів, відповідних вхідним, для кожного шару ШНМ, що, звичайно, є дуже трудомісткою операцією і не завжди здійсненою. Другий варіант – динамічне підстроювання вагових коефіцієнтів синапсів, в ході якої вибираються, як правило, найбільш слабкі зв'язки, які змінюються на малу величину в ту або іншу сторону. Зберігаються ж тільки ті зміни, які спричинили зменшення помилки на виході всієї мережі. Очевидно, що даний метод "проб", не дивлячись на свою простоту, що здається, вимагає громіздких рутинних обчислень. І, нарешті, третій, більш прийнятний варіант – розповсюдження сигналів помилки від виходів ШНМ до її входів, в напрямі зворотному прямому розповсюдженню сигналів в звичайному режимі роботи. Цей алгоритм навчання ШНМ отримав назву процедури зворотного розповсюдження (back-propagation algorithm) і є найширше використовуваним. Саме він буде докладніший розглянутий надалі, оскільки використовується в магістерській роботі.

В.2 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 змодельювати наявну залежність. Наприклад, одношарова лінійна мережа здатна відтворювати тільки лінійні функції. Якщо використовувати багатшарові лінійні ме-

режі, 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, верояно, 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 вибирати ту, яка простіше.

Необхідність багатократних експериментів веде до того, що контрольна множина починає грати ключову роль у виборі моделі нейронної мережі, тобто стає частиною процесу навчання. Тим самим його роль як незалежного критерію якості моделі ослабляється, оскільки при великому числі експериментів ви-

никає 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.3 Властивість узагальнення

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

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

Класичним прикладом непередставницької моделі нейронної мережі є наступна ситуація. При проектуванні системи машинного зору, призначеної для автоматичного розпізнавання цілей, мережа навчалася на 100 картинках, що містять зображення танків, і на 100 інших картинках, де танків не було. Після навчання мережі був досягнутий стовідсотково "правильний" результат. Але коли на вхід мережі були подані нові дані, вона безнадійно провалилася. У чому ж була причина? З'ясувалося, що фотографії з танками були зроблені в похмурий, дощовий день, а фотографії без танків – в сонячний день. Мережа навчилася уловлювати різницю в загальній освітленості. Щоб мережа могла результативно працювати, її слід було навчати на даних, де б були присутні всі погодні умови і типи освітлення, при яких мережу передбачається використовувати, і це не кажучи ще про рельєф місцевості, вугілля і дистанцію зйомки і т.д.

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

го постаратися зробити так, щоб спостереження різних типів були представлені рівномірно, і відповідно цьому інтерпретувати результати, які видає мережа.

Здатність мережі, навченої на деякій множині 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 відповідність між входами і бажаними значеннями на виході. Така процедура навчання носить назву адаптації і достатньо широко застосовується для настройки параметрів нейронних мереж.

В.4 Метод зворотного розповсюдження помилки

Згідно методу найменших квадратів, цільовою функцією помилки ШНМ, що мінімізується, є величина

$$E(w) = \frac{1}{2} \sum_{jp} (y_{jp}^{(N)} - d_{jp})^2, \quad (\text{В.1})$$

де $y_{jp}^{(N)}$ – реальний вихідний стан нейрона j у вихідному шару N нейронної мережі при подачі на її входи p -го образу;

d_{jp} – ідеальний (бажаний) вихідний стан цього нейрона.

Підсумовування ведеться по всіх нейронах вихідного шару і по всіх оброблюваних мережею образах. Мінімізація ведеться методом градієнтного спуску, що означає підстроювання вагових коефіцієнтів таким чином

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}}. \quad (\text{В.2})$$

Тут w_{ij} – ваговий коефіцієнт синаптичного зв'язку, що сполучає i -й нейрон шару $(n-1)$ з j -м нейроном шару n

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}}. \quad (\text{B.3})$$

Тут під y_j як і раніше, розуміється вихід нейрона j , а під s_j – зважена сума його вхідних сигналів, тобто аргумент активаційної функції. Множник $\frac{\partial y_j}{\partial s_j}$ є похідною цієї функції по її аргументу. З цього виходить, що похідна активаційною функція повинна бути визначена на всій осі абсцис U зв'язку з цим функція одиничного стрибка і інші активаційні функції з неоднорідностями не підходять для даних ШНМ. У них застосовуються такі гладкі функції, як гіперболічний тангенс або класичний сигмоїд з експонентою. У разі гіперболічного тангенса

$$\frac{\partial y}{\partial s} = 1 - s^2. \quad (\text{B.4})$$

Третій множник $\frac{\partial s_j}{\partial w_{ij}}$, очевидно, рівний виходу нейрона попереднього шару $y_i^{(n-1)}$.

Що стосується першого множника в (B.3), він легко розкладається таким чином

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_k} \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_k} w_{jk}^{(n+1)}. \quad (\text{B.5})$$

Тут підсумовування по k виконується серед нейронів шару $(n+1)$. Ввівши нову змінну

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j}, \quad (\text{B.6})$$

отримаємо рекурсивну формулу для розрахунків величин $\delta_j^{(n)}$ шару n з величин $\delta_k^{(n+1)}$ більш старшого шару $(n+1)$

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right] \frac{\partial y_j}{\partial s_j}. \quad (\text{B.7})$$

Для вихідного ж шару

$$\delta_j^{(N)} = (y_j^{(N)} - d_j) \frac{\partial y_j}{\partial s_j}. \quad (\text{B.8})$$

Тепер можна записати (B.2) в розкритому вигляді

$$\Delta w_{ij}^{(n)}(t) = -\eta \delta_j^{(n)} y_i^{(n-1)}. \quad (\text{B.9})$$

Іноді для додання процесу корекції вагів деякої інерційності, що згладжує різкі скачки при переміщенні по поверхні цільової функції, (B.1) доповнюється значенням зміни ваги на попередній ітерації

$$\Delta w_{ij}^{(n)}(t) = -\eta [\mu w_{ij}^{(n)}(t-1) + (1-\mu) \delta_j^{(n)} y_i^{(n-1)}] \quad (\text{B.10})$$

де μ – коефіцієнт інерційності t – номер поточної ітерації.

Таким чином, повний алгоритм навчання ШНМ за допомогою процедури зворотного розповсюдження будується в наступному порядку [45].

1. Подати на входи мережі один з можливих образів і в режимі звичайного функціонування ШНМ, коли сигнали розповсюджуються від входів до виходів, розрахувати значення останніх. Як було відмічено раніше

$$s_j^{(n)} = \sum_{i=1}^M y_i^{(n-1)} w_{ij}^{(n)}, \quad (\text{B.11})$$

де M – число нейронів в шарі $(n-1)$ з урахуванням нейрона з постійним вихідним станом $+1$, що задає зсув;

$$y_i^{(n-1)} = x_{ij}^{(n)} - i\text{-й вхід нейрона } j \text{ шару } n$$

$$y_j^{(n)} = f(s_j^{(n)}) \quad (\text{B.12})$$

$f()$ – сигмоїд.

$$y_q^{(0)} = I_q \quad (\text{B.13})$$

I_q – q -а компоненту вектора вхідного образу.

2. Розрахувати $\delta^{(N)}$ для вихідного шару по формулі (B.8).

Розрахувати по формулі (B.9) або (B.10) зміни вагів $\Delta w^{(N)}$ шару N .

3. Розрахувати по формулах (B.7) і (B.9) (або (B.7) і (B.10)) відповідно $\delta^{(n)}$ і $\Delta w^{(n)}$ для решти всіх шарів $n = N - 1, \dots, 1$.

4. Скоректувати всі ваги в ШНМ

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \quad (\text{B.14})$$

5. Якщо помилка мережі істотна, перейти на крок 1, інакше – кінець.

На кроці 1, на входи мережі поперемінно у випадковому порядку пред'являються всі тренувальні образи, щоб мережа, образно кажучи, не забувала одні у міру запам'ятовування інших.

З виразу (B.9) виходить, що, коли вихідне значення $y_i^{(n-1)}$ прагне до нуля, ефективність навчання помітно знижується. При двійкових вхідних векторах в середньому половина вагових коефіцієнтів не коректуватиметься [45], тому область можливих значень виходів нейронів $[0,1]$ бажано зрушити в межі $[-0.5, +0.5]$, що досягається простими модифікаціями логістичних функцій. Наприклад, сигмоїд з експонентою перетвориться до вигляду

$$f(x) = -0,5 + \frac{2}{2 + e^{-\alpha x}}. \quad (\text{B.15})$$

Даний алгоритм навчання ШНМ має декілька "вузьких місць". По-перше, в процесі навчання може виникнути ситуація, коли великі позитивні або негативні значення вагових коефіцієнтів змістять робочу точку на сигмоїдах багатьох нейронів в область насичення. Малі величини похідної від логістичної функції приведуть у відповідність з (B.15) і (B.16) до зупинки навчання, що паралізує ШНМ. По-друге, застосування методу градієнтного спуску не гарантує,

що буде знайдений глобальний, а не локальний мінімум цільової функції. Ця проблема пов'язана ще з однією, а саме – з вибором величини швидкості навчання.

Доказ збіжності навчання в процесі зворотного розповсюдження заснований на обчисленні похідних, тобто прирості вагів i , отже, швидкість навчання повинна бути нескінченно малою, проте в цьому випадку навчання відбуватиметься неприйнятно поволі. З іншого боку, дуже великі корекції вагів можуть привести до постійної нестійкості процесу навчання. Тому в якості η зазвичай вибирається число менше 1, але не дуже маленьке, наприклад 0,1, і воно, в загальному випадку, може поступово зменшуватися в процесі навчання. Крім того, для виключення випадкових попадань в локальні мінімуми іноді, після того, як значення вагових коефіцієнтів за стабілізуються, η короткочасно сильно збільшують, щоб почати градієнтний спуск з нової точки. Якщо повторення цієї процедури кілька разів приведе алгоритм в один і той же стан ШНМ, можна більш менш упевнено сказати, що знайдений глобальний мінімум.

Алгоритми, що модифікують процедуру зворотного розповсюдження, змінюють процедуру вибору функції критерію, що оптимізується, вибір оптимального кроку і правила корекції параметрів навчаної мережі детально описані в [46, 47].

Відзначимо, що всі ці алгоритми не узгоджуються з даними нейробіології, де показано, що сигнали в біологічних нейронних мережах можуть розповсюджуватися тільки в одному, прямому напрямі.

Запропоновані також алгоритми настройки багат шарової ШНМ без використання процедури зворотного розповсюдження помилки, які отримали назву «Автономні алгоритми». Ці алгоритми засновані на тому факті, що для багат шарової ШНМ існує деяка множина значень параметрів, що настроюються, на якій вектор корекції параметрів, що настроюються, обчислюється досить просто. Ці алгоритми за інших рівних умов гірше вирішують задачу навчання, ніж алгоритми зворотного розповсюдження помилки, проте вони є структурно незалежними і дуже простими з обчислювальної точки зору.

ДОДАТОК Г

ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ

Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання нелінійних динамічних систем є отримання їх адекватних математичних моделей, який базується, як правило, на теоретичному і експериментальному аналізі властивостей цих систем. Теоретичний аналіз, заснований на вивченні фізичних і хімічних процесів, які відбуваються в системі, дозволяє отримати математичний опис, наприклад, диференціальних рівнянь. При експериментальному аналізі на основі спостережень вхідних і початкових сигналів системи отримують або її параметричну, або непараметричну модель. Найбільшого розповсюдження придбали параметричні моделі, які вимагають рішення задач структурної і параметричної ідентифікації і що використовують обмежену кількість параметрів. Не дивлячись на величезну кількість робіт, різноманіття видів нелінійностей не дозволяє створити єдину теорію ідентифікації нелінійних систем. Вживаний найчастіше класичний підхід заснований на апроксимації нелінійностей, наприклад рядами Вольтерра, Гаммерштейна, Вінера, поліномами Колмогорова-Габора і ін., проте сфера застосування таких моделей обмежена. Крім того, додаткові труднощі отримання адекватного математичного опису зумовлює наявність в реальних сигналах завад, яка вимагає попередньої фільтрації сигналів.

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

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

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

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

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

ξ – завада вимірювання вихідного сигналу.

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

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

Тоді рівняння (Г.1) може бути записане таким чином:

$$\tilde{y}(k) = f[x(k), k] + \xi(k). \quad (\Gamma.3)$$

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

Схема ідентифікації нелінійного об'єкту (Г.1) на підставі ШНМ приведена на рис. Г.1.

Найбільшого застосування для вирішення завдання ідентифікації нелінійних динамічних об'єктів (Г.1) сьогодні отримали багат шаровий персептрон (рис. Г.2) і радіально-базисні мережі (РБМ) (рис. Г.3), що використовують апроксимацію нелінійного оператора $f(\cdot)$ нейронною мережею.

Рівняння БП має вигляд

$$\hat{y}(k) = f^q \left[(\omega^q)^T f^{q-1} \left[(\omega^{q-1})^T f^{q-2} \left[\dots f^1 \left[(\omega^1)^T \right] x(k) \right] \right] \right], \quad (\Gamma.4)$$

де q – кількість куль в мережі;

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

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

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

$$\hat{y}(k) = \sum_{i=1}^N c_i \Phi_i(x), \quad (\Gamma.5)$$

де c_i – вагові коефіцієнти, які підлягають визначенню.

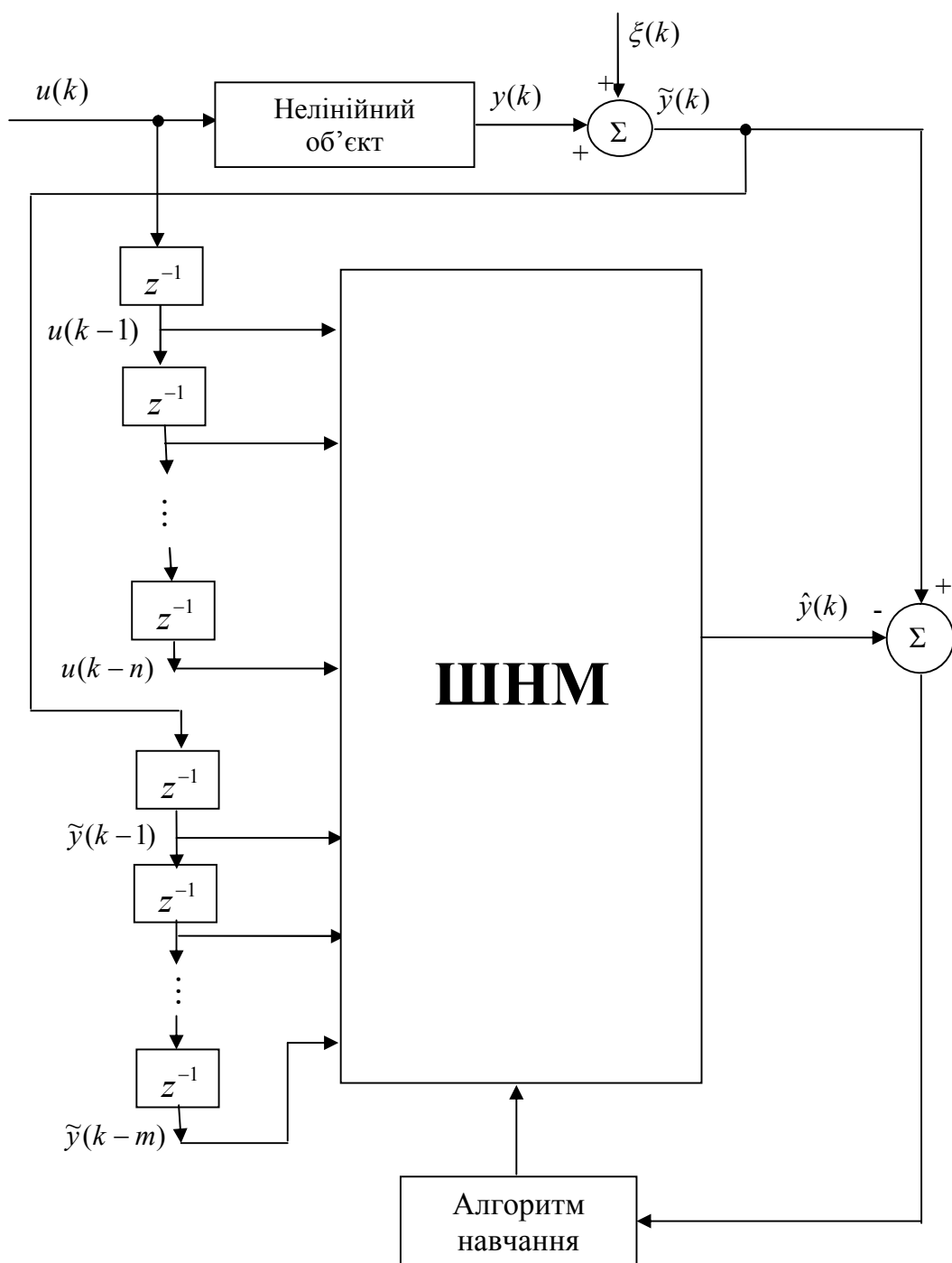


Рисунок Г.1 – Схема ідентифікації динамічного об'єкту

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

$$\Phi_i(x) = \exp \left\{ -\frac{\|x - \mu_i\|^2}{\sigma_i^2} \right\}, \quad (\text{Г.6})$$

де μ_i, σ_i – центри і радіуси базисних функцій відповідно;

$\|\cdot\|$ – евклідова норма, то визначенню підлягає вектор

$$\omega(k) = (c_0(k), c_1(k), \mu_1^T(k), \sigma_1(k), \dots, c_N(k), \mu_N^T(k), \sigma_N(k))^T,$$

який містить всі невідомі параметри мережі (зазвичай приймають $\Phi_0(x) = 1$).

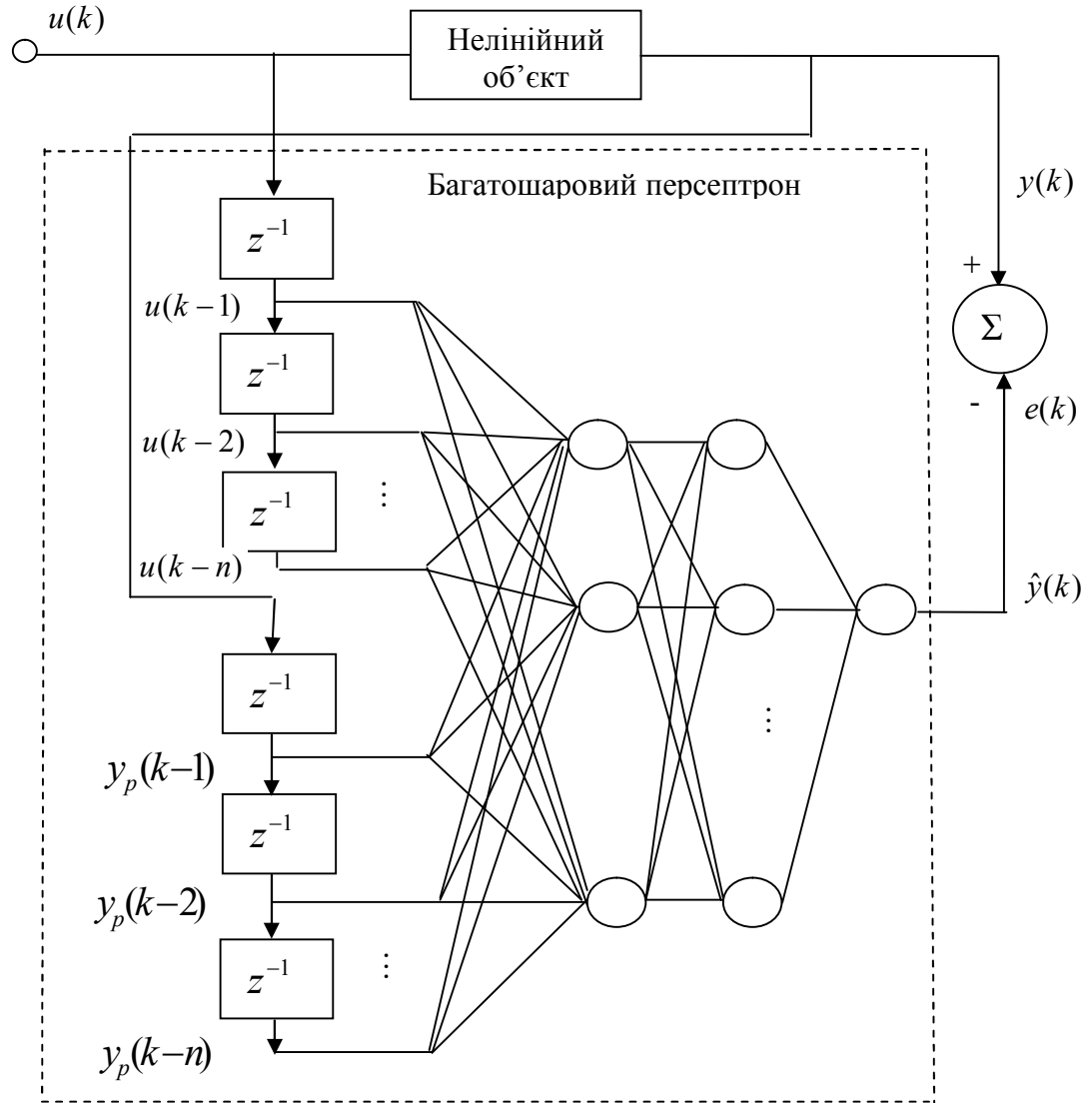


Рис Г.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового перцептрона

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

$$J = e^2(k) = M \left\{ [\tilde{y}(k) - \hat{y}(k)]^2 \right\}. \quad (\Gamma.7)$$

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

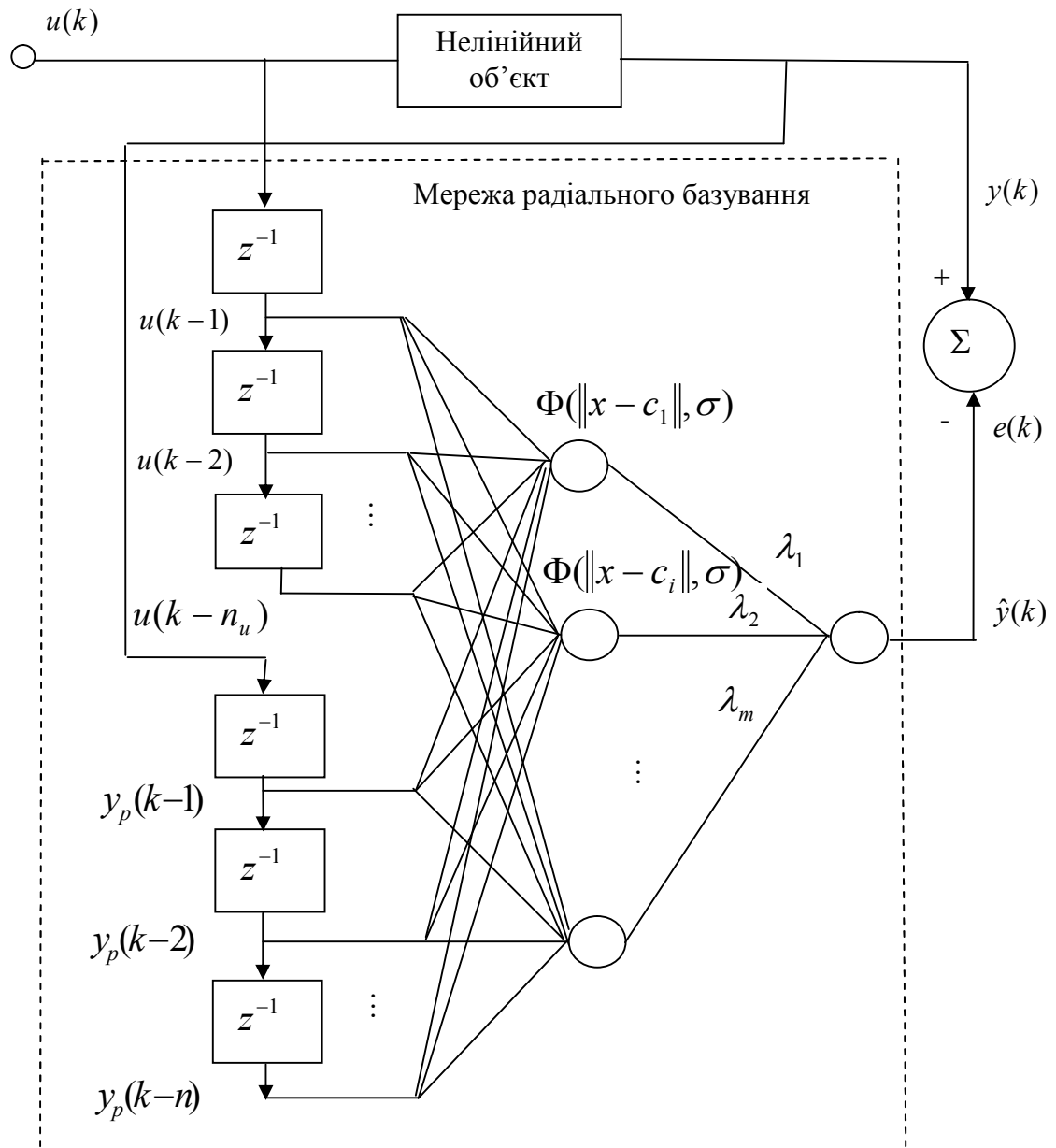


Рисунок Г.3 – Ідентифікація динамічного об'єкту за допомогою РБМ

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% 2024 рік
% Система управління електроприводом переміщення моста
% мостового крана

% Вихідні дані
kd=0.42/2;
In=57;
Mn=In/kd;
ktp=21.4;
kn=0.05454;
kc=0.168;
Tmc=0.005;
Ta=0.036;
Te=0.035;
Re=0.824;
b12=0 ;
c12=10;
Jd=1.02;
Jm=4.06;
Js=Jd+Jm;
Uz=24;

w0=3;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);

Tme=2*Tmc+Ta;
kic=Re/(2*Tmc*kc*ktp);
kpc=kic*Te;
kpe=Js*kd^2*kc/(2*Tme*kn);
kie=kpe/(4*Tme);

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

```

zeros(1,2) kn/(Ta*kd) zeros(1,3) -1/Ta 0
zeros(1,5) kie -kie 0];
B=[0 -1/Jm;
zeros(4,2);
1/Ta 0;
0 0
0 0];
C=zeros(6,8);
C(1,1)=1;C(2,2)=1;C(3,3)=1;C(4,4)=1; C(5,5)=1;C(6,8)=1;
%C(1,3)=1;C(2,4)=1;C(3,5)=1;C(4,8)=1;
D=zeros(6,2);

```