

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

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

КВАЛІФІКАЦІЙНА РОБОТА
магістра

на тему

СИСТЕМА УПРАВЛІННЯ РОЛЬГАНГОМ ПЕРЕД НОЖИЦЯМИ
СТАНА ГАРЯЧОЇ ПРОКАТКИ 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)

_____/ Єгор ЗІМЕНКО
(підпис) (ім'я та прізвище)

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

Рецензент _____/ Олег ЛИТВИН
(підпис) (ім'я та прізвище)

«До захисту допущено»

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

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

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

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

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

« » 2024р.

ЗАВДАННЯ

на кваліфікаційну роботу магістрського рівня вищої освіти

студенту (ці) Єгор ЗІМЕНКО
(ім'я, прізвище)

1. Тема роботи Система управління рольгангом перед ножицями стана гарячої прокатки 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 Маса зливку $m_m=9,8-13,1 \cdot 10^3$ кг; діаметр бочки ролика $d_p=0,47$ м; діаметр шийки ролика $d_{ш}=0,22$ м; довжина бочки ролика $l_p=1,25$ м; кількість роликів секції $Z_p=3$; маса ролика $m_p=1,6 \cdot 10^3$ кг; окружна швидкість на бочці ролика $V_p=0,2-2$ м/с; передавальне число редуктора $i=5,3$; двигун постійного струму Д816; тиристорний електропривод КТЕУ 1000/220; трансформатор ТСЗП-400/10У3. Згладжуючий реактор СРОСЗ-700 МУХЛ4. Перерегулювання не більше 10%, час регулювання до 5с.
4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережевих технологій для управління багатомасовою електро-механічною системою механізму рольганга перед ножицями стана гарячої прокатки. Розрахувати систему підлеглого регулювання з внутрішнім контуром струму і зовнішнім контуром ЕРС. Оптимізувати контур струму за модульним критерієм, а контур ЕРС – за симетричним критерієм. 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. Магістрська кваліфікаційна робота. Формальні ознаки. 2. Рольганг перед ножицями. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейромережеві системи управління. Структурні схеми. 6. Нейромережева ідентифікація. NARX моделі. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси. 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14,15. Магістрська кваліфікаційна робота. Висновки

6. Консультант:

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

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

Керівник

(підпис)

Тетяна ВАСИЛЕЦЬ

(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Єгор ЗІМЕНКО

(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 нейромережевих технологій для управління багатомасовою електро-механічною системою механізму рольганга перед ножицями стана гарячої прокатки.	09.09.2024-18.09.2024	
2	Розрахунок системи підлеглого регулювання координат електроприводу з оптимізацією контурів за модульним та симетричним критеріями. Моделювання системи на EOM.	19.09.2024-15.10.2024	
3	Дослідження системи з урахуванням пружних елементів. Розробка математичної моделі двомасової системи та моделювання двомасової системи на EOM.	16.10.2023-05.11.2023	
4	Синтез нейромережевої системи управління на EOM з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережевої системи.	06.11.2024-25.11.2024	
5	Оформлення роботи	26.11.2024-05.12.2024	

Студент (ка)

(підпис)

Єгор ЗІМЕНКО

(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 208 сторінок, 63 рисунків, 1 таблиці, 60 використаних джерел, 5 додатків.

Мета роботи – розробка методів синтезу системи управління електроприводом механізму рольганга за ножицями стана гарячої прокатки для підвищення показників якості функціонування системи на основі нейрорегуляторів.

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

В результаті виконання магістерської кваліфікаційної роботи науково обґрунтовано перспективність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму рольганга за ножицями стана гарячої прокатки. Розроблена система управління координат електроприводу механізму. Розраховані параметри системи автоматичного управління. Проведено моделювання системи на ЕОМ.

Розроблена математична модель системи з урахуванням пружних елементів механічної частини електроприводу. Розраховані динамічні характеристики двомасової системи.

Розглянута структура нейромережевої системи управління і виконаний синтез нейрорегулятора NN Predictive Controller в системі MATLAB. Проведено моделювання нейромережевої системи управління з нейрорегулятором на ЕОМ. Встановлено, що синтезована нейромережева система має високі показники якості функціонування.

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

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

Ключові слова: ШТУЧНИЙ НЕРОН, НЕЙРОННІ МЕРЕЖІ, ПРОКАТНИЙ СТАН, РОЛЬГАНГ, СИСТЕМА ПІДЛЕГЛОГО УПРАВЛІННЯ, ОДНОМАСОВА І ДВОМАСОВА СИСТЕМИ, НЕЙРОМЕРЕЖЕВА СИСТЕМА, НЕЙРОРЕГУЛЯТОР

ABSTRACT

An explanatory message consists of: 208 pages, 63 figures, 1 table, 60 used sources, 5 additions.

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

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

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

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

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

Keywords: ARTIFICIAL NERO, NEURAL NETWORKS, FLATTING MILL, ROLGANG, INFERIOR CONTROL SYSTEM, ONE-MASS AND TWO-MASS SYSTEM, NEURAL NETWORK SYSTEM, NEUROREGULATOR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

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

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.	
ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	14
1.1 Галузі застосування штучних нейронних мереж	14
1.2 Огляд структур систем нейрорегулювання.	
Вибір структури системи	19
1.4 Призначення і класифікація рольгангів	41
1.5 Технічні вимоги до електроприводів рольгангів	42
1.6 Постановка задачі синтезу нейромережевої системи управління	43
ВИСНОВКИ ДО РОЗДІЛУ 1	46
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ	
ЕЛЕКТРОПРИВОДОМ РОЛЬГАНГА СТАНА ГАРЯЧОЇ ПРОКАТКИ	48
2.1 Технічні дані механізму і силового електроустаткування.	
Вимоги до системи управління	48
2.2 Перевірка електродвигуна по нагріву і перевантажувальній	
здатності	50
2.3 Розрахунок параметрів силового кола електроприводу	55
2.4 Вибір структури системи автоматичного регулювання	
координат електроприводу	59
2.5 Синтез послідовного корегуючого пристрою контура струму	61
2.6 Синтез послідовного корегуючого пристрою контура ЕРС	69
2.7 Дослідження динамічних характеристик системи	77
ВИСНОВКИ ДО РОЗДІЛУ 2	81
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ	
ПРУЖНИХ ЕЛЕМЕНТІВ	83
3.1 Математична модель двомасової системи	83
3.2 Розрахунок перехідних процесів двомасової системи	91
ВИСНОВКИ ДО РОЗДІЛУ 3	97

РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	98
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	98
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	100
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	103
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	146
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	148
ВИСНОВКИ ДО РОЗДІЛУ 4	148
ВИСНОВКИ	156
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	158
ДОДАТОК А. СТРУКТУРА ШТУЧНОГО НЕЙРОНА	164
ДОДАТОК Б. КЛАСИФІКАЦІЯ НЕЙРОННИХ МЕРЕЖ І ЇХ ВЛАСТИВОСТІ.....	169
ДОДАТОК В. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ	177
В.1 Сутність процесу навчання	177
В.2 Алгоритми навчання нейронних мереж.....	179
В.3 Метод зворотного розповсюдження помилки.....	183
В.4 Перенавчання і узагальнення нейронних мереж.....	193
ДОДАТОК Г. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ	196
Г.1 Моделі систем, використовувані при нейромережевій ідентифікації	196
Г.2 Нейронні мережі як універсальні моделі	201
Г.3 Ідентифікація динамічних систем за допомогою нейронних мереж	204
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	207

ВСТУП

Актуальність теми. Темпи розвитку всього народного господарства визначаються рівнем виробництва металу. Без продукції металургійного виробництва не може існувати і розвиватися сучасна незалежна держава, а без наявності в країні високорозвинутої металургії неможливі швидкі темпи розвитку народного господарства.

Прокатне виробництво є завершальним етапом металургійного циклу. У сучасних прокатних цехах технологічні процеси організовані за потоковими та безперервними принципами, що сприяє широкому впровадженню комплексної механізації й автоматизації. У зв'язку з цим механічне обладнання прокатних цехів вирізняється високою складністю та різноманітністю як за призначенням, так і за конструкцією. Розробка нового прокатного обладнання та експлуатація складних машин і агрегатів у таких цехах вимагають використання досягнень різних наукових і технічних галузей, а також підготовки висококваліфікованих спеціалістів.

Реверсивні стани гарячої прокатки є найбільш продуктивними агрегатами металургійних заводів. Їх продуктивність зазвичай обчислюється мільйонами тонн прокатуваного металу в рік. Через обтискові, заготівки і деякі інші реверсивні стани проходить майже важ прокатуваний метал. Обтискові стани визначають в більшості випадків продуктивність прокатних цехів.

Найбільш характерною особливістю приводів реверсивних станів гарячого плющення (у тому числі і головного приводу) є велике число включень, що часто перевищує 1000 в годину. Сталі режими роботи мають неістотне значення.

Сучасний реверсивний стан гарячої прокатки - високомеханізований і автоматизований агрегат, що включає широкий комплекс автоматично керованих механізмів. Значні успіхи в створенні необхідного устаткування і систем управління реверсивними електроприводами досягнуті в останнє десятиліття. Цей

період характеризується зміною технічної оснащеності і впровадженням нових машин і способів управління ними.

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 [1-12]. 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. Навчена на обмеженій безлічі 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.

Таким чином, використання нейромережевих технологій в системах управління, надає величезні можливості для пошуку ефективних рішень самих

різних задач, багато з яких раніше не мали рішення із задовільною точністю. Тому тема роботи є актуальною.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління електроприводом механізму рольганга за ножицями стана гарячої прокатки для підвищення показників якості функціонування системи на основі нейромережових регуляторів. Досягнення поставленої мети здійснюється рішенням наступних основних задач.

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

Об'єктом дослідження є динамічні процеси в системі управління електроприводом механізму рольганга перед ножицями стана гарячої прокатки з урахуванням пружних елементів.

Предметом дослідження є моделі і методи синтезу нейромережової системи управління двомасовою електромеханічною системою механізму рольганга перед ножицями стана гарячої прокатки.

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 дало змогу оцінити якість роботи систем і підтвердило ефективність отриманих результатів.

Наукова новизна отриманих результатів. Новизна наукових результатів, досягнутих у процесі виконання роботи, полягає в наступному:

1. Вперше науково обґрунтовано доцільність застосування нейромережових технологій для управління багатомасовою електромеханічною системою механізму рольганга перед ножицями стана гарячої прокатки .

2. Вперше запропоновано і синтезовано нейромережеву систему управління електроприводом механізму рольганга перед ножицями стана гарячої прокатки з урахуванням пружних елементів, що має високі показники якості функціонування.

3. Вперше запропонована математична модель системи управління електроприводом механізму рольганга передножицями стана гарячої прокатки з урахуванням пружних елементів у вигляді багат шарової прямонаправленої мережі – багат шарового персептрона.

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

5. Одержали подальший розвиток методи синтезу нейромережових 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 рольганга перед ножицями стана гарячої прокатки , 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.

У роботі наведено посилання на авторів і відповідні джерела при використанні відомих залежностей, експериментальних даних та наукових положень.

Апробація результатів роботи. Основні положення та наукові результати були апробовані на науковій конференції студентів ННІ «УПА» ХНУ ім. В.Н. Каразіна (Харків, 2024 р.).

Структура і об'єм роботи Робота складається з вступу, 4 розділів основної частини, висновків, додатків і списку використаних літературних джерел. Повний об'єм роботи складає 208 сторінок, серед них: 63 рисунки, з яких 37 рисунків за текстом, 26 рисунків на 24 окремих сторінках, 1 таблиця, 5 додатків на 45 сторінках, список літературних джерел з 60 найменувань на 6 сторінках.

РОЗДІЛ 1

АНАЛІЗ СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Галузі застосування штучних нейронних мереж

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

Нейронні мережі — це обчислювальні структури, що імітують прості біологічні процеси, які зазвичай асоціюються з роботою людського мозку. Вони представляють собою системи, здатні навчатися шляхом аналізу позитивних і негативних зв'язків. Основним елементом таких мереж є штучний нейрон, або просто "нейрон", назва якого походить від його біологічного аналога.

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

Апроксимація функцій.

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

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

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

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

Асоціативна пам'ять.

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

Стиснення пам'яті.

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

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

Розпізнавання і класифікація/

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

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

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

Оптимізаційні завдання/

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

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

Управління складними процесами/

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

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

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

Прогнозування/

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

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

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

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

Нейрокомп'ютери.

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

Таблиця 1.1 – Порівняльні характеристики комп'ютера і ШНМ

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

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

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

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

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

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

1.2 Огляд структур систем нейрорегулювання. Вибір структури системи

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

1.2.1 Послідовна схема нейроуправління

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

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

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

похідна яких обчислюється за виразом

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

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

Мета управління полягає в мінімізації квадратичної функції помилки

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

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

Вихідний сигнал мережі може бути обчислений таким чином:

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

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

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

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

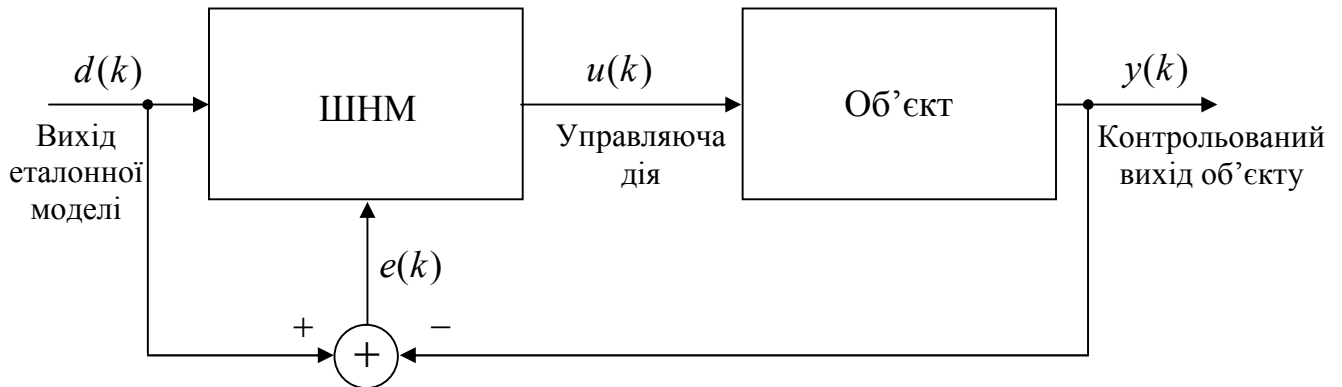


Рисунок 1.1 – Спеціалізоване нейроуправління

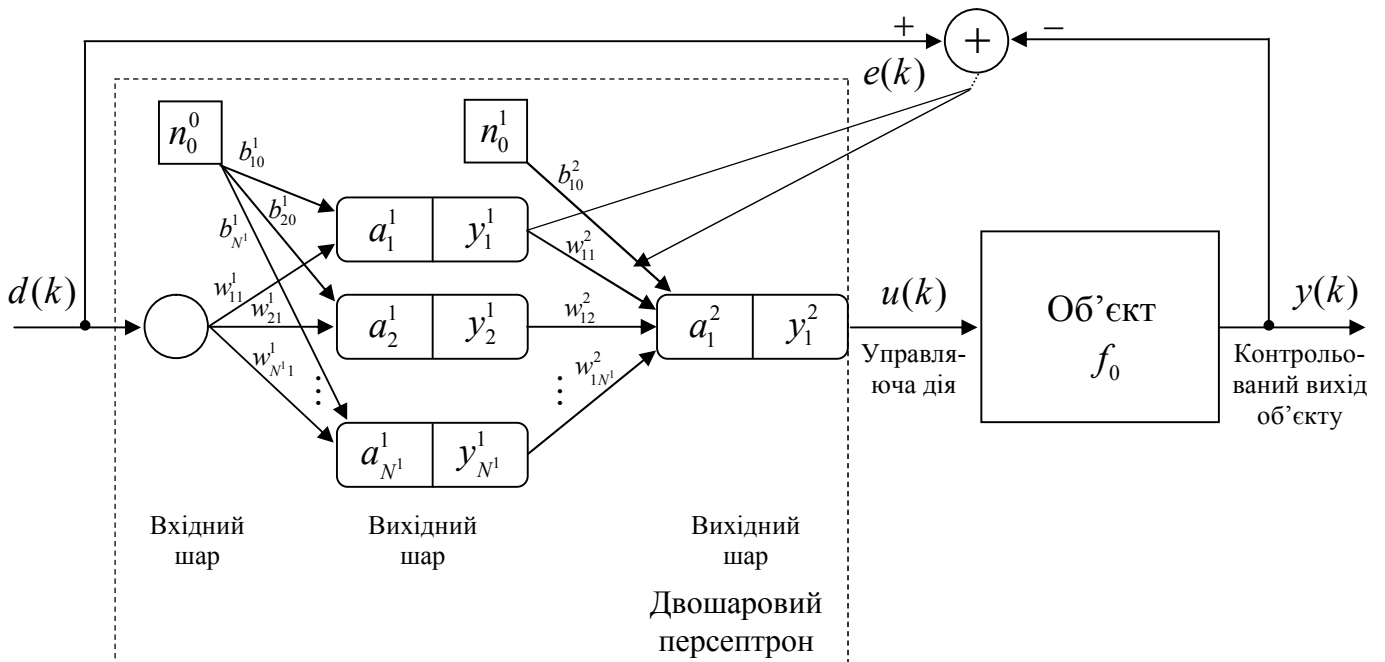


Рисунок 1.2 – Структура нейрорегулятора на основі двошарового персептрону

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

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

З (1.5) отримуємо

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

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

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

Тепер можемо записати загальну формулу настройки вагів вихідного шару

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

Аналогічно для нейронів прихованого шару, можемо записати

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

де

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

Таким чином, алгоритм настройки параметрів нейрорегулятора має вигляд:

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

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

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

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

де

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

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

1.2.2 Схема зворотного розповсюдження в часі

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

об'єкту, а надалі використовується для зворотного розповсюдження помилки на другому етапі функціонування системи – управлінні об'єктом. Використання нейроемулятора дозволяє обчислити Якобіан системи, замінюючи об'єкт його моделлю. В цьому випадку отримаємо:

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

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

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

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

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

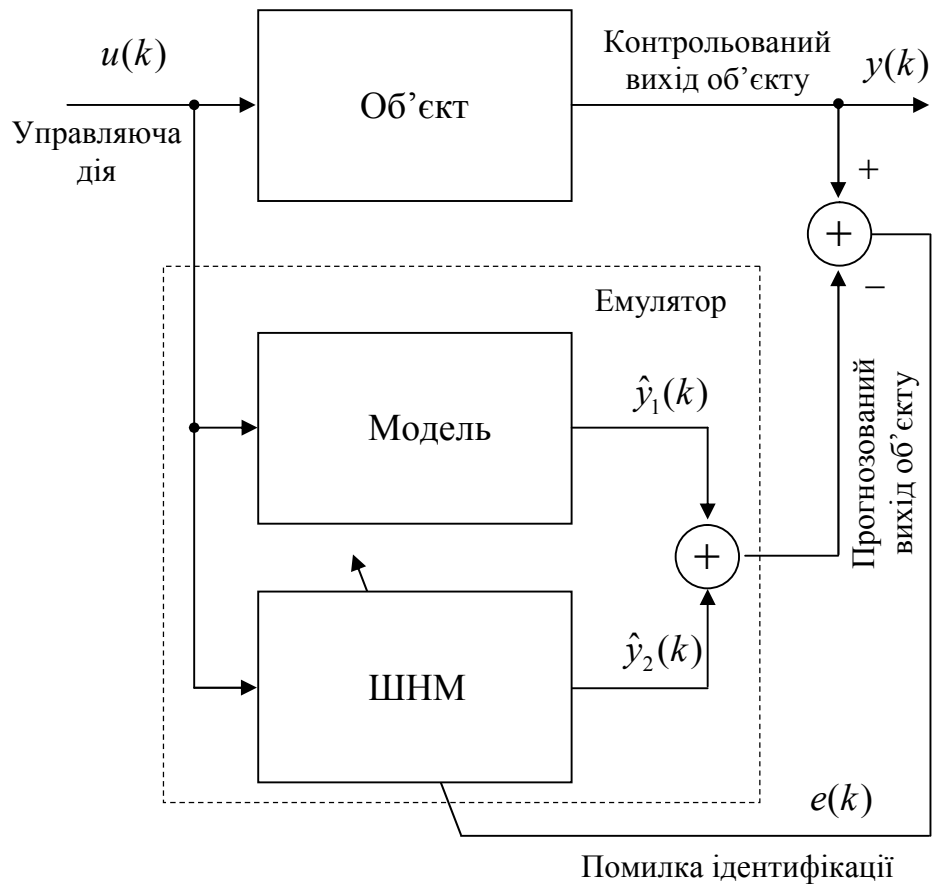


Рисунок 1.3 – Структура нейроемулятора з умовною моделлю

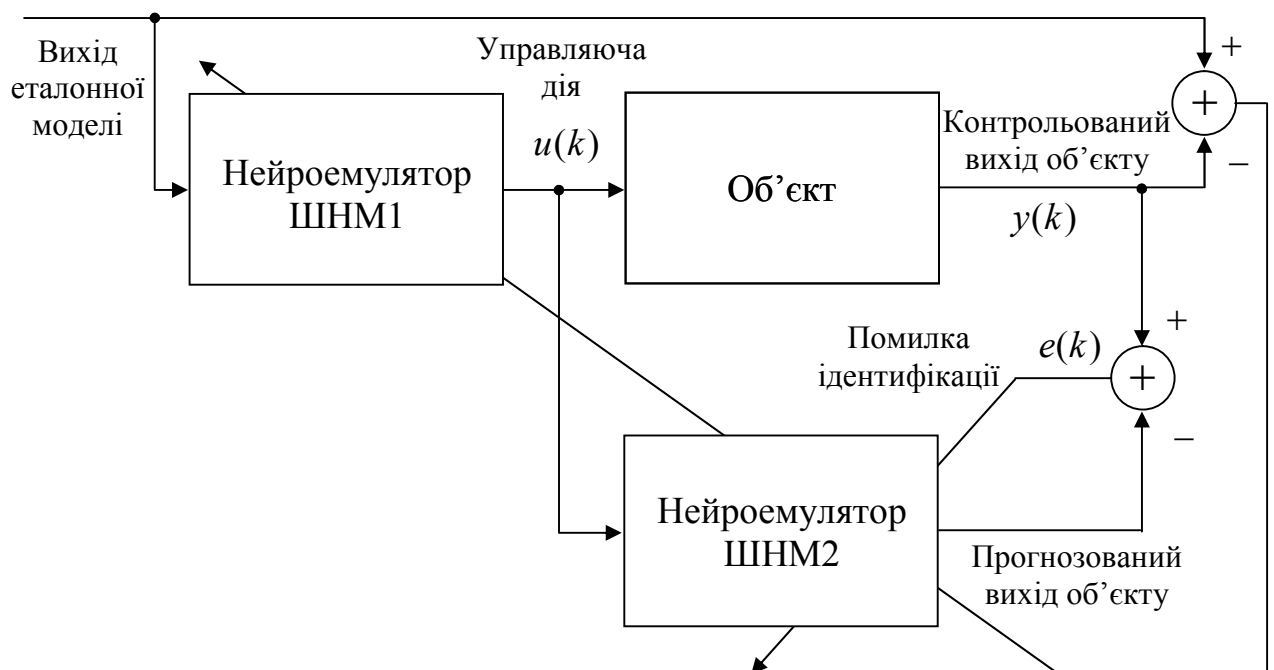


Рисунок 1.4 – Загальна схема інверсно-динамічного нейроуправління

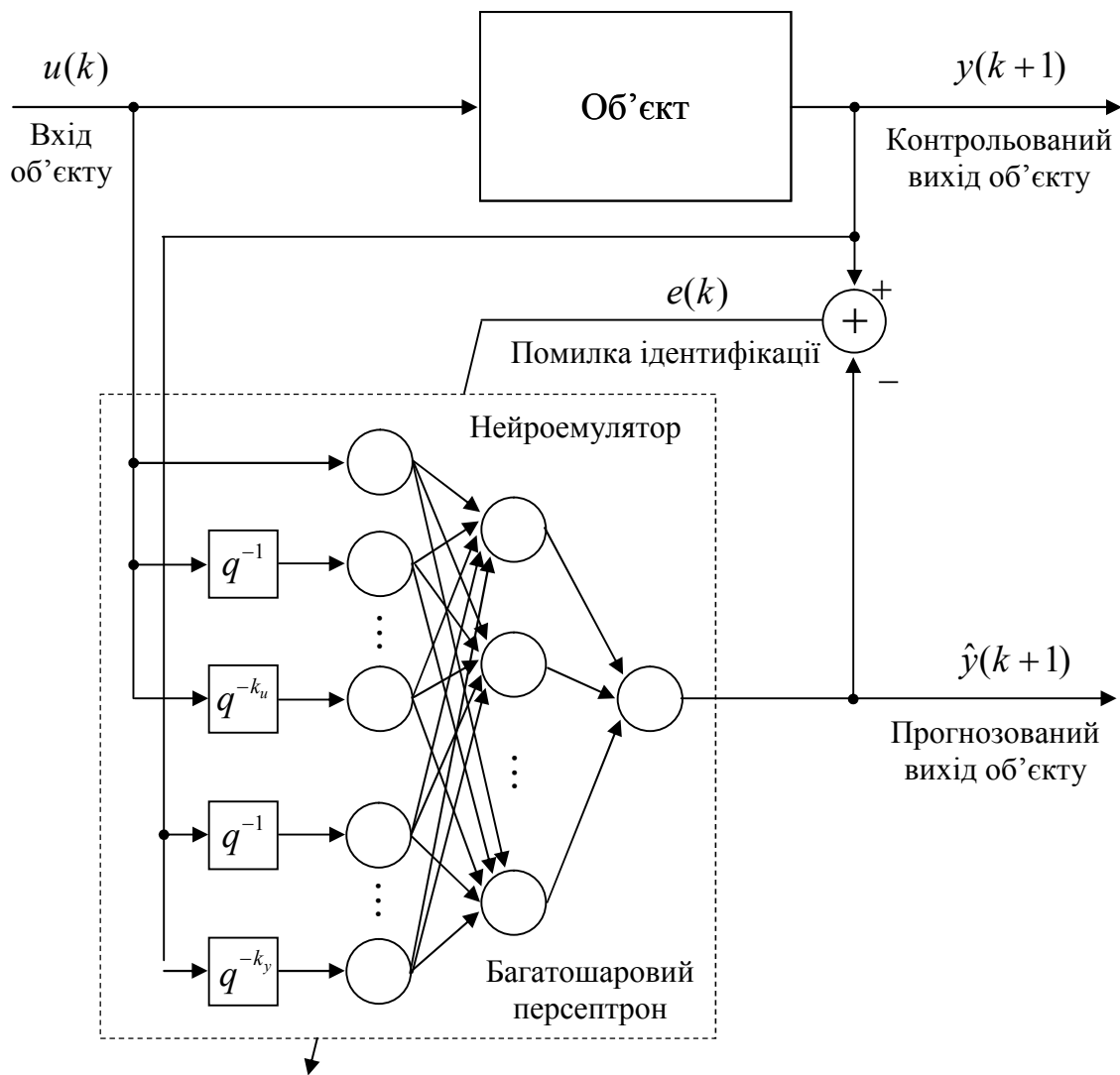


Рисунок 1.5 – Структура нейроемулятора

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

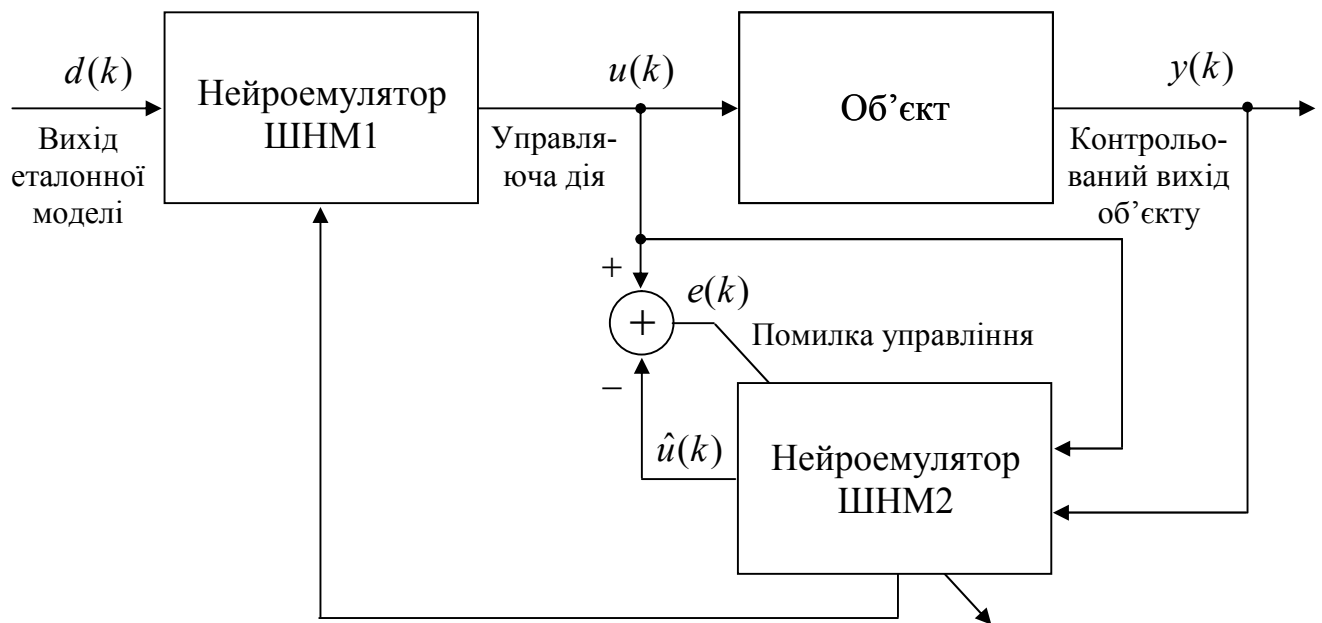


Рисунок 1.6 – Схема зворотного відображення

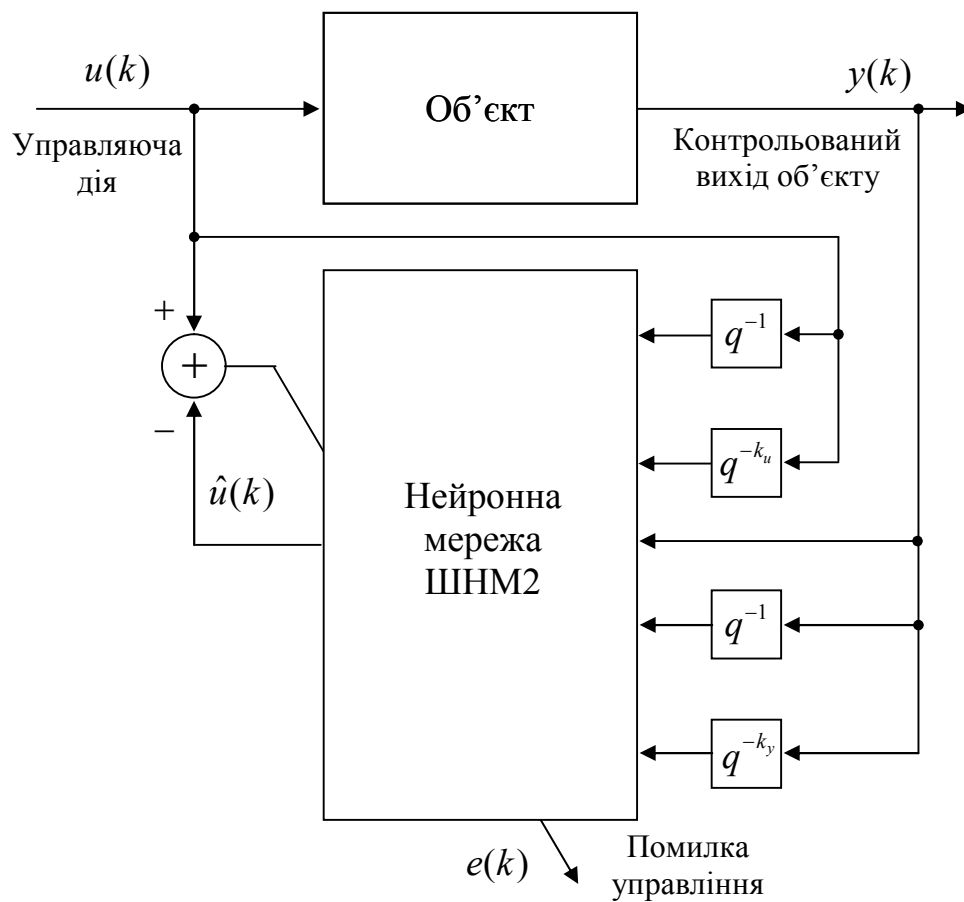


Рисунок 1.7 – Структура нейроемулятора для схеми зворотного відображення

1.2.3 Паралельне нейроуправління

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

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

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

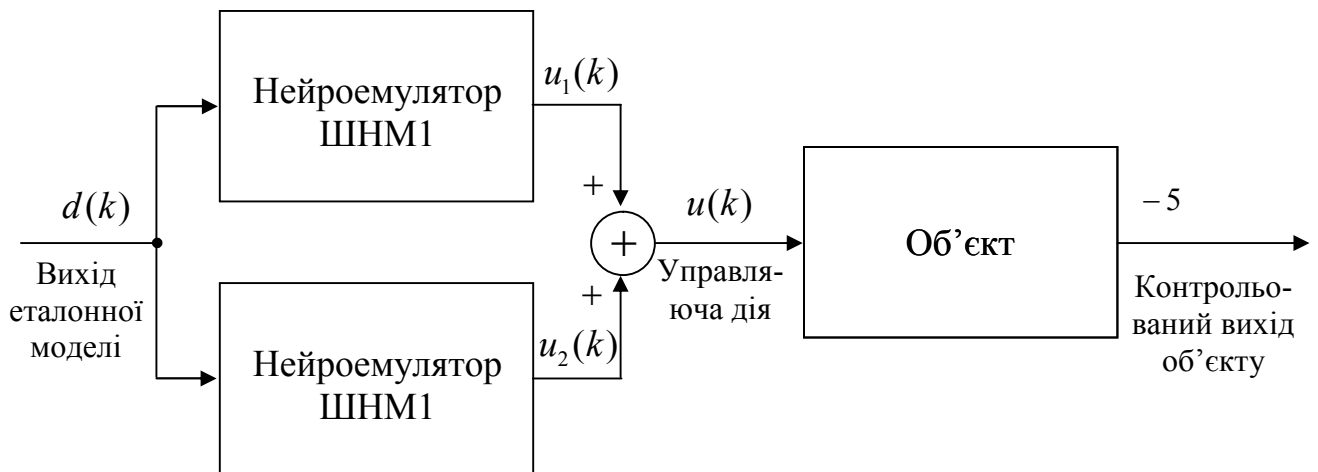


Рисунок 1.8 – Паралельна схема нейроуправління

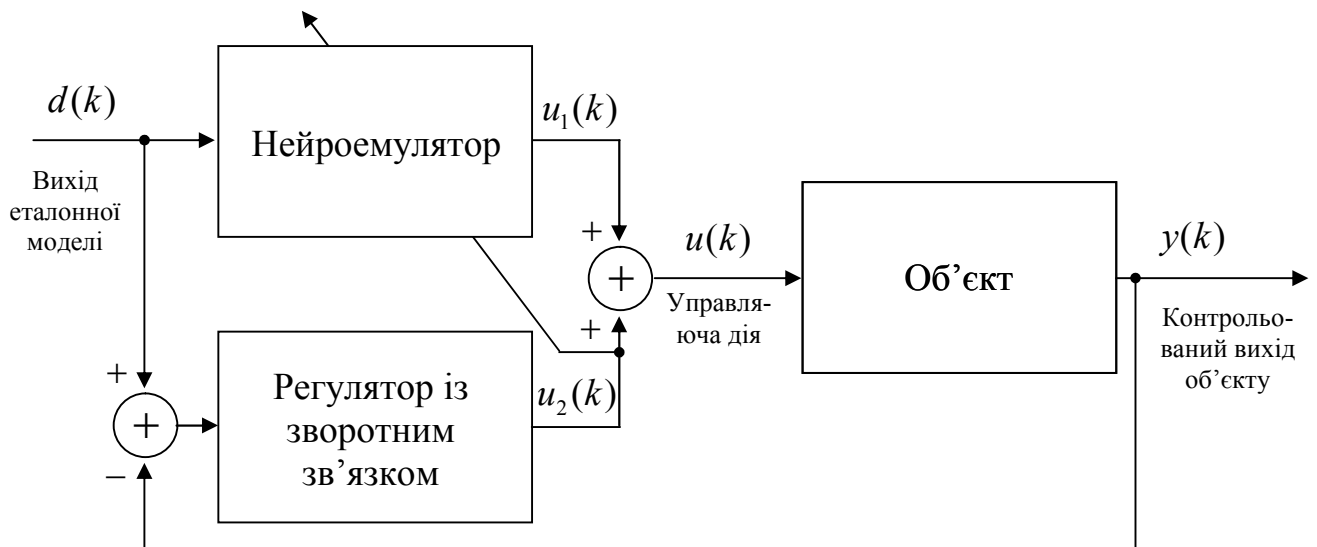


Рисунок 1.9 – Нейроуправління по помилці зворотного зв'язку

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

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

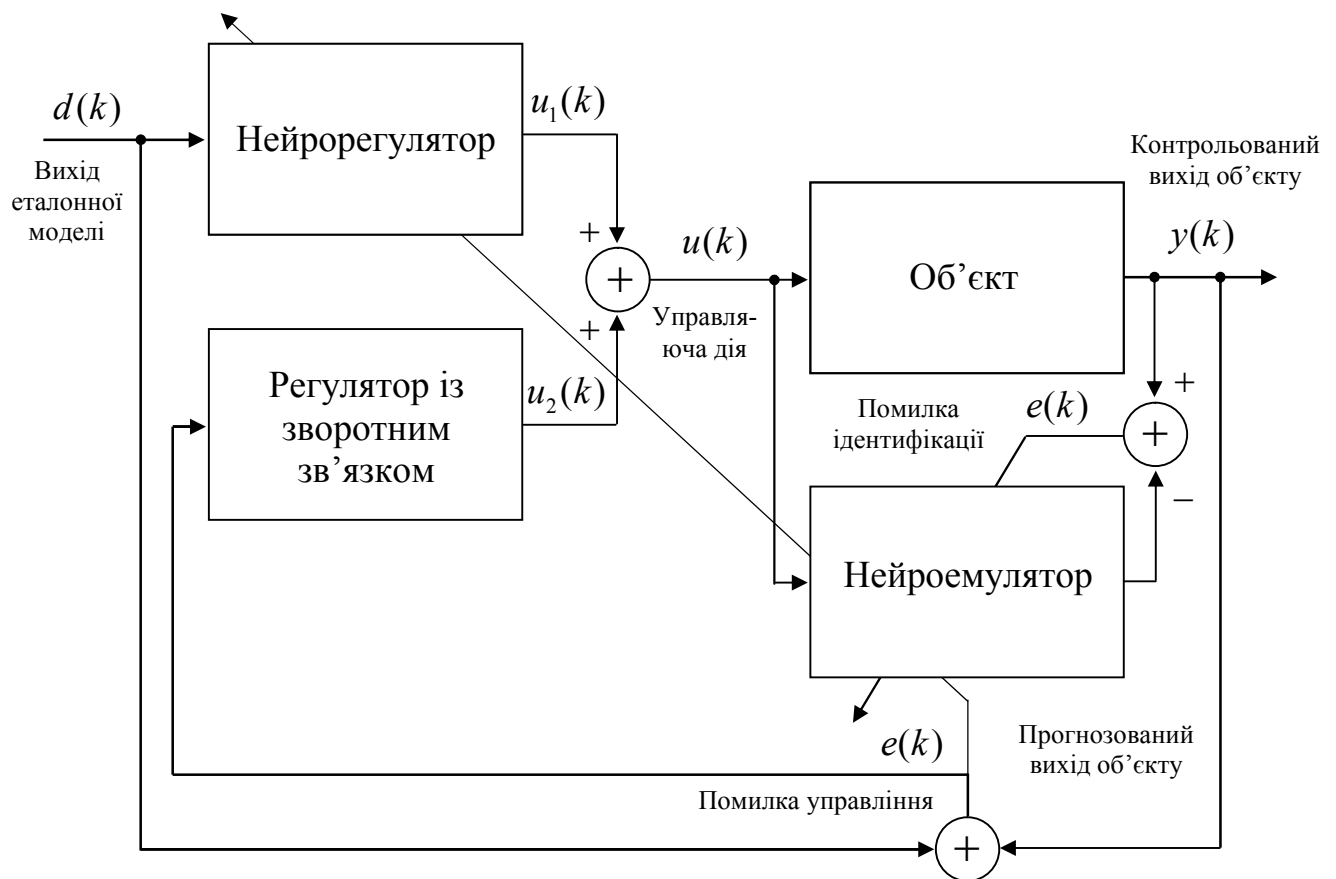


Рисунок 1.10 – Узагальнена схема нейроуправління по помилці зворотного зв'язку

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

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

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

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

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

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

Як критерій якості використовуємо функцію помилки (1.3), а для настройки вагів нейрорегулятора застосуємо метод найшвидшого спуску. Виберемо структуру нейрорегулятора як показано на рис.1.11.

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

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

а корекція вагів нейронів прихованого шару відповідно до правила:

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

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

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

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

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

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

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

де

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

Алгоритм настройки вагів нейрорегулятора для паралельної схеми управління має вигляд:

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

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

1.2.4 Пряме адаптивне нейроуправління

Розглянемо нелінійний об'єкт, описуваний NARX моделлю

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

де $y(k)$ – вихід об'єкту управління;

$u(k)$ – управляюча дія;

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

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

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

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

похідна якої обчислюється за виразом

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

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

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

Нейромережева модель об'єкту (НММ) має вид:

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

де $\hat{y}(k+1)$ – вихід ШНМ

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

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

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

де

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

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

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

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

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

Мінімум J може бути знайдений рекурсивно за допомогою методу найшвидшого спуску

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

де $\eta > 0$ – норма навчання.

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

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

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

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

Підставляючи (1.21) в (1.20), отримаємо

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

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

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

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

Тепер остаточно можна записати:

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

Таким чином, алгоритм прямого адаптивного управління має вигляд:

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

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

1.2.5 Управління з самонастройкою

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

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

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

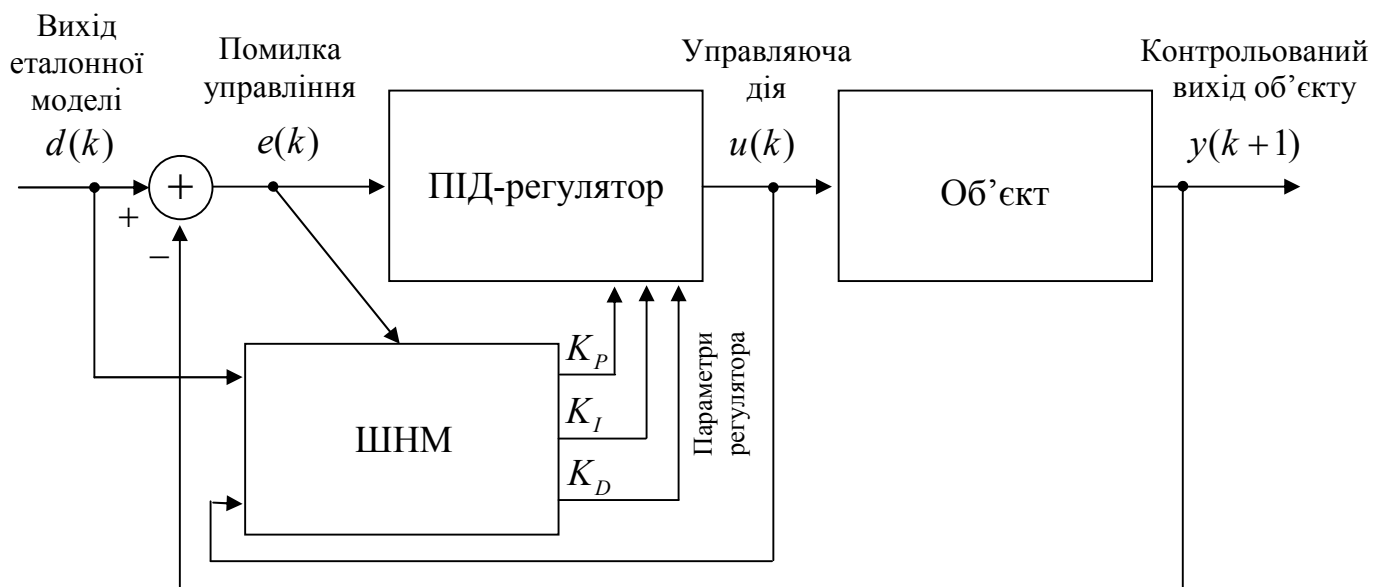


Рисунок 1.11 – Структура навчання ПІД-регулятора з само настройкою

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Оскільки правило функціонування нейронів вихідного шару має вигляд:

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

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

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

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

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

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

Тепер, ввівши позначення для локальної помилки вихідного шару

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

можемо записати правило настройки вагів нейронів вихідного шару

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

Перейшовши до прихованого шару, отримаємо

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

де δ_i^1 – локальна помилка прихованого шару

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

Таким чином, правило настройки вагів нейронів прихованого шару має вигляд

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

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

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

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

1.2.6 Нелінійне предиктивное управління

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

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

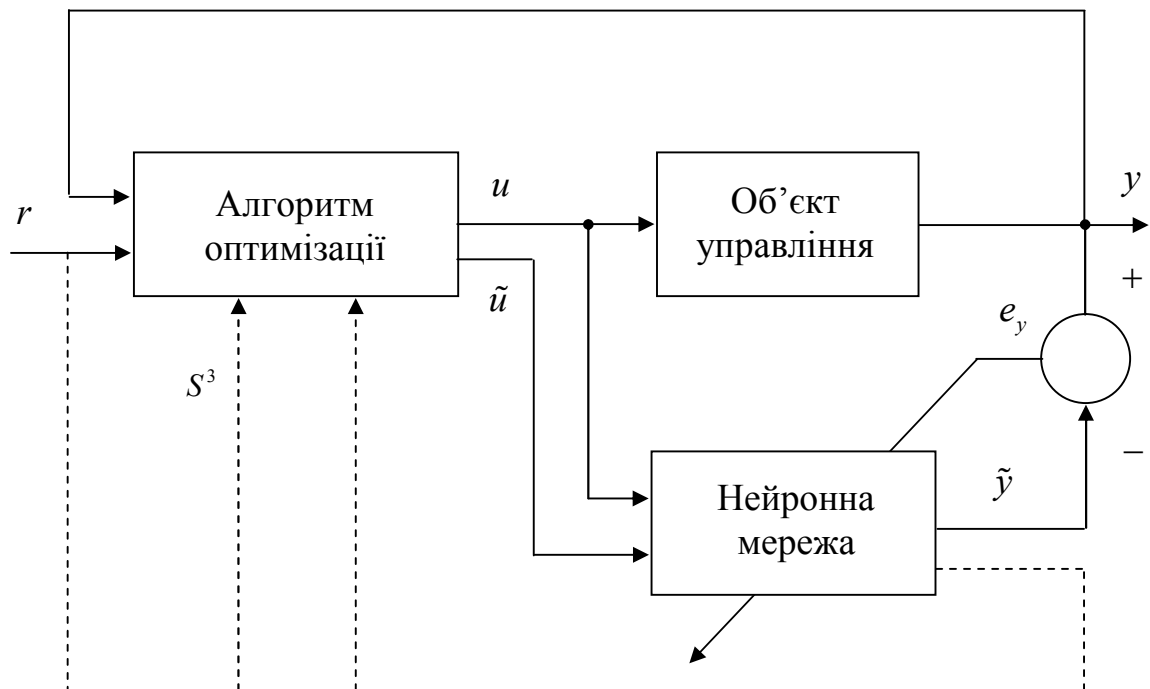


Рисунок 1.12 – Предиктивне регулювання

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

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

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

1.4 Призначення і класифікація рольгангів

Рольганги - це обладнання, призначене для переміщення металевого прокату за допомогою обертових роликів. Залежно від призначення, рольганги поділяються на такі види:

1. Робочі рольганги - розташовані у вузлі стану й слугують для подачі металу до валків та відведення його після прокатки. У великих станах перші ролики робочих рольгангів розташовані безпосередньо в станині кліті. Ці станинні ролики розміщуються близько до робочих валків, що сприяє кращому захопленню металу, особливо коротких злитків.

2. Допоміжні робочі рольганги або подовжувальні - є продовженням основних робочих рольгангів. Їх використовують у випадках, коли довжина прокатуваного металу перевищує довжину основних рольгангів.

3. Транспортні рольганги - забезпечують переміщення металу між механізмами стану або між різними станами. Залежно від напрямку потоку металу, вони можуть бути підвідними або відвідними. Рольганг, розташований на початку стану й призначений для прийому металу, називається приймальним.

Основними характеристиками рольганга є діаметр і довжина бочки ролика, крок рольганга (відстань між осями сусідніх роликів) та кількість роликів. Діаметр роликів визначається мінімально можливим для зниження маси за умови дотримання вимог міцності. Наприклад, у робочих рольгангах для обтискових станів і слябінгів діаметр роликів зазвичай становить 600–400 мм, тоді як для листових і сортових станів цей параметр варіюється в межах 350–200 мм.

Довжина бочки роликів відповідає довжині бочки валків стану або трохи перевищує її. Крок роликів підбирається таким чином, щоб метал одночасно спирався щонайменше на два ролики, що запобігає його прогинанню під власною вагою. У рольгангах обтискових станів крок становить $h_p = 0,7 \div 1 м$; у великосортних станів - $h_p = 1,2 \div 1,6 м$; у середньосортних - $h_p = 0,9 \div 1 м$; у тонколистових - $h_p = 0,5 \div 0,7 м$.

Кількість роликів залежить від довжини металу, що транспортується, і в деяких випадках може досягати кількох десятків. Лінійна швидкість робочих і відвідних рольгангів зазвичай перевищує швидкість прокатки приблизно на $5 \div 10\%$. Швидкість підвідних рольгангів становить $1,5 \div 2,5 \text{ м/с}$, а транспортних - $1,5 \div 12 \text{ м/с}$.

Деякі рольганги потребують регулювання швидкості, яке може бути плавним або ступінчастим. Плавне регулювання часто застосовується в робочих рольгангах нереверсивних регульованих станів. Ступінчасте регулювання зазвичай використовується в робочих рольгангах реверсивних обтискових станів.

1.5 Технічні вимоги до електроприводів рольгангів

Електроприводи робочих і розкочувальних рольгангів функціонують у повторно-короткочасному реверсивному режимі з високою частотою увімкнень.

Частота увімкнень на годину досягає:

- у робочих рольгангів блюмінга та слябінгу - 1000–1200;
- у розкочувальних рольгангів блюмінга та слябінгу - 450;
- у робочих рольгангів товстолистого стану гарячої прокатки - 850.

Висока частота увімкнень вимагає підвищених динамічних характеристик електроприводів під час розгону й гальмування, оскільки час роботи допоміжних механізмів істотно впливає на продуктивність прокатного стану.

Електроприводи повинні забезпечувати широкий діапазон регулювання швидкості - не менше (10–20):1.

Структура електроприводів:

- Реверсивна система із пропорційним регулятором частоти обертання або ЕРС, а також підлеглим регулятором струму якоря.
- Основна вимога до системи регулювання - висока швидкодія.

Час реверсу струму з $+I_n$ на $-I_n$ не повинен перевищувати 25–40 мс. Завдання швидкості може бути ступінчастим або плавно змінним і задається через контактний або сельсинний командоапарат із використанням задатчика інтенсивності, що формує динамічний струм двигуна.

1.6 Постановка задачі синтезу нейромережевої системи управління

Проведений аналіз літератури показав, що існує велика кількість підходів до проблеми синтезу систем управління, при цьому універсального і ідеального регулятора поки не існує. При синтезі систем управління механізмом повороту стріли роторного екскаватора, крім вже згаданих вимог, необхідно враховувати той факт, що опорно-поворотні механізми роторного екскаватора є багатомасовими пружними системами, коливання в яких виникають при змінах як управляючих, так і збурюючих дій. Традиційні регулятори не в змозі забезпечити задовільні показники якості функціонування системи.

Використання нейромережевих технологій управління значно спрощує вирішення математичних проблем, пов'язаних із аналітичним синтезом і аналізом властивостей проєктованих систем. Це зумовлено тим, що якість процесів управління та досягнуті характеристики в нейромережевих системах переважно визначаються фундаментальними властивостями багат шарових нелінійних нейронних мереж, а не аналітично виведеними оптимальними законами, які зазвичай реалізуються у вигляді програмного забезпечення. Налаштовувані багат шарові нейронні мережі мають низку переваг, які роблять їх ефективними для вирішення завдань управління динамічними об'єктами.

Універсальні властивості апроксимацій багат шарових мереж, грають ключову роль у формуванні нелінійних алгоритмів управління. На основі узагальненої теореми апроксимації Стоуна-Вейерштрасса зроблений висновок про те, що за допомогою нелінійних нейронних мереж можна скільки завгодно точно рівномірно наблизити будь-яку безперервну функцію багатьох змінних на будь-якій замкнутій обмеженій множині. Висновок послужив теоретичним обґрунтуванням можливого застосування багат шарових нейронних мереж для формування оптимальних функцій, що управляли, в динамічних системах. Через свою структуру і алгоритми навчання багат шарові мережі дозволяють вирішувати такі нелінійні задачі теорії управління, які традиційними аналітичними методами не завжди можна вирішити в повному об'ємі. Можливість форму-

вати дії практично будь-якої форми, що управляють, робить штучні нейронні мережі засобом, адекватним рівню складності нелінійного завдання. Тому нейромережеві системи управління відносять до так званих «інтелектуальних» систем, тобто до систем, за визначенням, з неповною інформацією, з складною або немодельованою динамікою і неконтрольованою зміною власних властивостей.

Адаптивність нейромережевих структур унаслідок їх навчання в процесі функціонування дозволяє коректувати в реальному часі функцію управління при неконтрольованих змінах динамічних і статичних характеристик об'єкту, використовуючи для цього поточну вимірювальну інформацію в системі.

Здібність нейронних мереж до паралельної обробки сигналів робить природним їх застосування для управління багатовимірними (або багатоканальними) об'єктами. Багатошарові нейронні мережі є однорідним обчислювальним середовищем. Це універсальні паралельні обчислювальні структури, призначені для вирішення найрізноманітніших класів завдань, зокрема для побудови регуляторів в системах управління нелінійними динамічними об'єктами.

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 нейромережевої системи управління електроприводом рольганга перед ножицями стана гарячої прокатки, що має високі динамічні характеристики. При цьому об'єкт управління характеризується наявністю в ній пружних елементів, що призводить до необхідності розглядати його як багатомасову систему. Система повинна забезпечити високоякісне регулювання по задаючій і збурюючій діях при ступінчастому вхідному сигналі з випадковою амплітудою.

Для досягнення поставленої мети необхідно вирішити наступні завдання. задач.

1. Виконати перевірку привідного двигуна електроприводу рольганга перед ножицями стана гарячої прокатки по нагріву і перевантажувальній здатнос-

ті. Виконати синтез системи управління електроприводом і провести моделювання системи на ЕОМ.

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

3. Розробити структурну схему нейромережевої системи управління, що реалізує високоякісне регулювання з врахуванням кінцевої жорсткості елементів механічної частини електроприводу.

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

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

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

ВИСНОВКИ ДО РОЗДІЛУ 1

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

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

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

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

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

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

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

7. В результаті аналізу вимог, що пред'являються до систем управління електроприводами основних і допоміжних механізмів прокатних станів і з урахуванням пружності механічних зв'язків показана доцільність використання нейромережевих технологій для синтезу нейромережевої системи управління електроприводом рольганга перед ножицями стана гарячої прокатки.

8. Проаналізовані нейромережеві підходи до рішення задачі управління. Сформульовано задачу дослідження, що складається в розробці нейромережевої системи управління електроприводом рольганга стана гарячої прокатки з урахуванням пружних механічних зв'язків..

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ ЕЛЕКТРОПРИВОДОМ РОЛЬГАНГА СТАНА ГАРЯЧОЇ ПРОКАТКИ

2.1 Технічні дані механізму і силового електроустаткування. Вимоги до системи управління

Вихідними даними для синтезу системи регулювання є наступні.

Технічні дані рольганга.

Маса злитка $m_m = 9800 \div 13100 \text{ кг}$

Діаметр бочки ролика $d_p = 0,47 \text{ м}$

Діаметр шийки ролика $d_{ш} = 0,22 \text{ м}$;

Довжина бочки ролика $l_p = 1,25 \text{ м}$

Кількість роликів секції $z_p = 3$

Окружна швидкість на бочці ролика $V_p = 0,2 \div 2 \text{ м/с}$

Момент інерції на тихохідному валу редуктора

не завантаженого рольганга $J_{mh} = 340 \text{ кг} \cdot \text{м}^2$

завантаженого рольганга (масою зливку 12 т) $J_{mз} = 970 \text{ кг} \cdot \text{м}^2$

Маса ролика $m_p = 1600 \text{ кг}$

Максимальне число включень в годину $k_{max} = 450$

Технічні дані електроустаткування.

Редуктор.

Передавальне число $i = 5,3$

Двигун постійного струму Д816

Номінальна потужність $P_n = 150 \text{ кВт}$

Номинальна напруга $U_n = 220 \text{ В}$

Номинальний струм якоря $I_n = 745 \text{ А}$

Номинальна частота обертання $n_n = 480 \text{ об/хв}$

Збудження – незалежне

Число активних провідників обмотки якоря $N = 504$

Число паралельних гілок якоря $2a=4$

Число полюсів $2p=4$

Магнітний потік полюса $\Phi_n = 104 \cdot 10^{-3} \text{ Вб}$

Номинальний струм збудження $I_{zn} = 7,44 \text{ А}$

Опір якоря і додаткових полюсів $R_{я} + R_{дп} = 0,043 \text{ Ом}$

Момент інерції $J_{дв} = 16,25 \text{ кг} \cdot \text{м}^2$

Тривалість включення ПВ=100%

Тиристорний електропривод

КТЕУ 1000/220-13222.

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

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

Трансформатор для тиристорного електроприводу

ТСЗП – 400/10У3

Номинальна напруга мережевої обмотки $U_1 = 6 \text{ кВ}$

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

Номинальна випрямлена напруга перетворювача $U_d = 230 \text{ В}$

Номинальна сила випрямленого струму $I_d = 1250 \text{ А}$

Втрати короткого замикання $\Delta P_{xx} = 1500 \text{ Вт}$

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

Згладжуючий реактор

СРОСЗ – 700 МУХЛ4

Номинальний струм реактора $I_{\text{рн}} = 1250 \text{ A}$

Індуктивність $L_p = 8 \text{ мГн}$

Вимоги до показників якості функціонування системи управління.

Перерегулювання швидкості при ступінчастій входній дії не більше 10%,
час регулювання не більше 5 с.

2.2 Перевірка електродвигуна по нагріву і перевантажувальній здатності

Перевірку двигуна за умовою нагріву виконуємо по методу еквівалентного моменту, для чого розраховуємо складові моменту статичного навантаження двигуна.

Розрахунок моментів проводимо для максимальної маси заготовки $m_{\text{max}} = 13100 \text{ кг}$.

Сила тяжіння метала:

$$G_{\text{м}} = m_{\text{max}} g ,$$

$$G_{\text{м}} = 13100 \cdot 9,8 = 128000 \text{ Н}$$

Сила тяжіння одного ролика:

$$G_{\text{р}} = m_{\text{р}} \cdot g ,$$

$$G_{\text{р}} = 16000 \cdot 9,8 = 15680 \text{ Н} ,$$

Статичний момент при буксуванні

$$M_{\text{сб}} = \frac{(z_{\text{р}} G_{\text{р}} + G_{\text{м}}) d_{\text{ш}} \mu_{\text{ш}} + G_{\text{м}} \mu_{\text{б}} d_{\text{р}}}{2 i \eta_{\text{н}}} ,$$

де $z_{\text{р}}$ - число роликів секції рольганга;

$d_{ш}$ - діаметр шийки ролика;

$\mu_{ш}$ - коефіцієнт тертя ковзання в підшипниках ролика, згідно [1]
 $\mu_{ш} = 0,009$;

η_x - ККД редуктора при холостому ході.

ККД зубчатих редукторів рольгангов при номінальному навантаженні, згідно [67] рівне $\eta_n = 0,95$. При холостому ході приймаємо $\eta_x = 0,5\eta_n = 0,475$.

$$M_{сб} = \frac{(3 \cdot 15680 + 128000) \cdot 0,22 \cdot 0,009 + 127400 \cdot 0,3 \cdot 0,47}{2 \cdot 5,3 \cdot 0,92} = 1882 \text{ Нм}$$

Момент холостого ходу рольганга, приведений до валу двигуна:

$$M_x = \frac{z_p G_p d_{ш} \mu_{ш}}{2 i \eta_x},$$

де $\eta_x = 0,5\eta_n$ - КПД редуктора при холостому ході; $\eta_x = 0,46$.

$$M_x = \frac{3 \cdot 15680 \cdot 0,22 \cdot 0,009}{2 \cdot 5,3 \cdot 0,46} = 18,9 \text{ Нм}$$

Повний момент статичного опору при транспортуванні металу:

$$M_c = \frac{(z_p G_p + G_m)(d_{ш} / 2) \mu_{ш} + G_m f}{i \eta_x},$$

де f - коефіцієнт тертя кочення металу по роликах; для гарячого металу
 $f = 0,15 \cdot 10^{-2}$.

Оскільки момент, потрібний для транспортування металу, невеликий то при визначенні M_c береться $\eta_x = 0,5\eta_n$.

$$M_c = \frac{(3 \cdot 15680 + 128000)(0,22 / 2) 0,009 + 127400 \cdot 0,15 \cdot 10^{-2}}{5,3 \cdot 0,46} = 152 \text{ Нм}.$$

Момент двигуна, необхідний для подолання сил інерції металу, роликів і двигуна:

$$M_{\text{дин}} = 2a \left[\frac{z_p J_p + J_m}{\eta_n i d_p} + \frac{i J_d}{d_p} \right],$$

де $a = \frac{dv}{dt}$ - лінійне прискорення роликів і металу; критичне прискорення $a_{\text{кр}}$,

вище за яке відбувається прослизання металу об ролик для гарячого металу:

$$a_{\text{кр}} = 1,47 \div 2,94 \text{ м/с}^2;$$

приймаємо $a = 2 \text{ м/с}^2$;

J_p - момент інерції ролика щодо осі ролика;

J_m - момент інерції рухомого металу, що приводиться до осі ролика.

У технічних умовах даний момент інерції завантаженого рольганга.

$$J_{\text{зм}} = z_p J_p + J_m = 970 \text{ кг} \cdot \text{м}^2.$$

J_d - момент інерції двигуна і редуктора, приймаємо

$$J_d = 1,1 \cdot J_{\text{дв}} = 1,1 \cdot 16,25 = 18 \text{ кг} \cdot \text{м}^2.$$

$$M_{\text{дин}} = 2 \cdot 2 \cdot \left[\frac{970}{0,92 \cdot 5,3 \cdot 0,47} + \frac{5,3 \cdot 18}{0,47} \right] = 2506 \text{ Нм}.$$

Повний момент двигуна, що розвивається при прискоренні

$$M_{\text{п}} = M_{\text{с}} + M_{\text{дин}},$$

$$M_{\text{п}} = 152 + 2506 = 2658 \text{ Нм},$$

при уповільненні

$$M_{\text{г}} = |M_{\text{с}} - M_{\text{дин}}|,$$

$$M_r = |152 - 2506| = 2354 \text{ Нм}.$$

Для перевірки двигуна по тепловому режиму визначимо час пуску t_n , гальмування t_m і час роботи з постійною швидкістю t_p . Розглянемо найбільш несприятливий випадок – розгін електроприводу до максимальної швидкості $V_{\max}$ і відповідне гальмування від $V_{\max}$ до 0.

$$t_n = t_r = \frac{V_{\max}}{a},$$

$$t_n = t_r = \frac{2}{2} = 1 \text{ с}.$$

Час циклу при числі включень в годину $k_{\max} = 450$.

$$t_{\text{ц}} = \frac{3600}{450} = 8 \text{ с}.$$

Час роботи при ПВ=40%

$$t_p = \frac{8}{100} \cdot 40 = 3,2 \text{ с}.$$

Час роботи із сталою швидкістю

$$t_y = t_{\text{ц}} - 2t_n,$$

$$t_y = 1,2 \text{ с}.$$

Еквівалентний момент двигуна

$$M_e = \sqrt{\frac{M_n^2 t_n + M_r^2 t_r + M_c^2 t_y}{t_{\text{ц}}}},$$

$$M_e = \sqrt{\frac{2658^2 \cdot 1 + 2354^2 \cdot 1 + 152^2 \cdot 1,2}{8}} = 2267 \text{ Нм}.$$

Номінальний момент двигуна

$$M_{\text{н}} = \frac{P_{\text{н}}}{\omega_{\text{н}}},$$

$$M_{\text{н}} = \frac{150 \cdot 10^3}{50,2} = 2988 \text{ Нм},$$

де

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

$$\omega_{\text{н}} = \frac{3,14 \cdot 480}{30} = 50,2 \text{ с}^{-1}.$$

Еквівалентний момент менше номінального моменту двигуна, отже по тепловому режиму двигун відповідає вимогам.

Для перевірки двигуна по перевантажувальній здатності визначимо величину заздальгідь-виправдовуваного моменту

$$M_{\text{пв}} = \frac{(z_{\text{р}} G_{\text{р}} + G_{\text{м}}) d_{\text{ш}} \mu_{\text{ш}} \pm G_{\text{м}} \mu_{\text{б}} d_{\text{р}}}{2 i \eta_{\text{н}}} \pm 2 a_{\text{кр}} \left(\frac{z_{\text{р}} J_{\text{р}}}{i \eta_{\text{н}} d_{\text{р}}} + i \frac{J_{\text{д}}}{d_{\text{р}}} \right),$$

де знак плюс відповідає прискоренню, а мінус – уповільненню

$$z_{\text{р}} J_{\text{р}} = J_{\text{мн}}.$$

Прискорення приводу

$$M_{\text{пв}} = \frac{(3 \cdot 15680 + 128000) \cdot 0,22 \cdot 0,009 + 128000 \cdot 0,3 \cdot 0,47}{2 \cdot 5,3 \cdot 0,92} + \\ + 2 \cdot 2,94 \left(\frac{340}{5,3 \cdot 0,92 \cdot 0,47} + \frac{5,3 \cdot 18}{0,47} \right) = 4834 \text{ Нм}$$

Уповільнення приводу:

$$M_{\text{пв}} = \frac{(3 \cdot 15680 + 128000)0,22 \cdot 0,009 - 128000 \cdot 0,3 \cdot 0,47}{2 \cdot 5,3 \cdot 0,92} -$$
$$- 2 \cdot 2,94 \left(\frac{3400}{5,3 \cdot 0,92 \cdot 0,47} + \frac{5,3 \cdot 16,25}{0,47} \right) = -1083 \text{ Нм}$$

Величина гранично-виправдовуваного моменту використовується для перевірки двигуна по перевантажувальній здатності.

Відношення моментів

$$\frac{M_{\text{пв}}}{M_{\text{н}}} = \frac{4834}{2988} = 1,67.$$

Оскільки перевантажувальна здатність двигунів постійного струму складає $(2 \div 2,5)M_{\text{н}}$, то по перевантажувальній здатності двигун вибраний правильно.

2.3 Розрахунок параметрів силового кола електроприводу

Приведена до ланцюга випрямленого струму індуктивність трансформатора

$$L_{\text{т}} = U_{\text{к}} \cdot \frac{1}{2mf} \cdot \frac{E_{\text{д0}}}{I_{\text{дн}}},$$

де m - фазність пульсації випрямленої напруги перетворювача; для трифазної мостової схеми $m = 6$;

f - частота живлячої мережі;

$I_{\text{дн}}$ - сила номінального випрямленого струму (сила номінального струму навантаження перетворювача);

$E_{\text{д0}}$ - ЕДС холостого ходу перетворювача; для трифазної мостової схеми випрямлення

$$E_{\text{д0}} = 2,34U_{2\phi},$$

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

$$E_{d0} = 2,34 \cdot 115 = 269B$$

$$L_{\tau} = 0,064 \cdot \frac{1}{2 \cdot 6 \cdot 50} \cdot \frac{262}{1000} = 0,284 \cdot 10^{-4} \Gamma H$$

Індуктивність якірного ланцюга електродвигуна

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

де γ - постійна, рівна 0,6 для некомпенсованих двигунів;

p - число пар головних полюсів;

$$L_{\text{як}} = 0,6 \frac{1}{50,2 \cdot 2} \cdot \frac{220}{745} = 8,1 \cdot 10^{-4} \Gamma H.$$

Індуктивність головного кола

$$L_{\Sigma} = L_{\text{як}} + L_{\text{м}},$$

$$L_{\Sigma} = 8,1 \cdot 10^{-4} + 0,284 \cdot 10^{-4} = 8,576 \cdot 10^{-4} \Gamma H.$$

Опір якоря електродвигуна з урахуванням додаткових полюсів і компенсаційної обмотки і нагріву машини до робочої температури

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

де $\alpha_{\tau} = 1,36$ - коефіцієнт зміни опору при нагріві, для машин з класами ізоляції В, F і Н

Згідно паспортних даних електродвигуна

$$R_{\text{як}} + R_{\text{дп}} = 0,0091 \text{ Ом}, \quad R_{\text{ко}} = 0,$$

$$R_{\text{як}} = 1,36 \cdot 0,0091 = 0,0124 \text{ Ом}.$$

Номінальна ЕРС електродвигуна

$$E_{\text{н}} = U_{\text{н}} - I_{\text{н}} R_{\text{як}},$$

$$E_{\text{н}} = 220 - 745 \cdot 0,0124 = 210,8 \text{ В}.$$

Постійна проти ЕДС двигуна

$$c = \frac{E_{\text{н}}}{\omega_{\text{н}}},$$

$$c = \frac{210,8}{50,2} = 4,2.$$

Коефіцієнт посилення двигуна

$$k_{\text{д}} = \frac{1}{c},$$

$$k_{\text{д}} = \frac{1}{4,2} = 0,24.$$

Коефіцієнт посилення тиристорного перетворювача

$$k_{\text{тп}} = \frac{E_{d0}}{U_{\text{м}}},$$

де $U_{\text{м}}$ - амплітудне значення напруги порівняння

$$k_{\text{тп}} = \frac{269}{10} = 27.$$

Еквівалентний активний опір тиристорного перетворювача

$$R_{\text{пр}} = R_{\text{х}} + R_{\text{т}},$$

де R_x - еквівалентний опір анодного перекриття

$$R_x = 0,5 U_{\kappa} \frac{E_{d0}}{I_{дн}},$$

U_{κ} - напруга короткого замикання трансформатора;

$$R_x = 0,5 \cdot 0,064 \cdot \frac{269}{1000} = 0,011 \text{ Ом}.$$

R_T - приведенне до ланцюга випрямленого струму активний опір обмоток трансформатора

$$R_T = (0,1 - 0,2) \cdot U_{\kappa} \cdot \frac{\pi}{m} \cdot \frac{E_{d0}}{I_{дн}},$$

$$R_T = 0,15 \cdot 0,064 \cdot \frac{3,14}{6} \cdot \frac{269}{1000} = 0,0013 \text{ Ом}.$$

$$R_{np} = 0,011 + 0,0012 = 0,0122 \text{ Ом}.$$

Активний опір головному ланцюгу

$$R_{\Sigma} = R_{як} + R_{np},$$

$$R_{\Sigma} = 0,0124 + 0,0122 = 0,0246 \text{ Ом}.$$

Електромагнітна постійна часу головного кола

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

$$T_e = \frac{8,576 \cdot 10^{-4}}{0,0246} = 0,036 \text{ с}.$$

Електромагнітна постійна часу якорного кола електродвигуна

$$T_a = \frac{L_{\text{як}}}{R_{\text{як}}},$$

$$T_a = \frac{8,2 \cdot 10^{-4}}{0,0126} = 0,067 \text{ с}.$$

Електромеханічну постійну часу електроприводу визначаємо для режиму, коли рольганг завантажений злитком максимальної ваги

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

де J_m - сумарний момент інерції на валу електродвигуна

$$J_m = J_d + \frac{J_{\text{мз}}}{i^2},$$

$$J_m = 18 + \frac{970}{5,3^2} = 18 + 34,5 = 52,5 \text{ кг} \cdot \text{м}^3.$$

$$T_m = \frac{52,5 \cdot 0,0246}{4,2^2} = 0,073 \text{ с}.$$

2.4 Вибір структури системи автоматичного регулювання координат електроприводу

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

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

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

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

Згідно з рекомендаціями [71], обирається двоконтурна система автоматичного управління з дворазовою інтеграцією, яка включає внутрішній контур струму і зовнішній контур, замкнутий по електрорушійній силі (ЕРС) двигуна. Системи регулювання швидкості з зворотним зв'язком за ЕРС використовуються для електроприводів у випадках, коли вимоги до точності регулювання швидкості не є надто високими. Серед електроприводів загальнопромислових механізмів електроприводи цього класу становлять переважну більшість.

Характерною особливістю вживаних електроприводів з роздільним управлінням ТП є наявність зони нечутливості при роботі груп тиристорів, а також виникнення режиму переривистих струмів при малих навантаженнях. Для усунення негативного впливу вказаних особливостей ТП в контур регулювання струму вводимо підлеглий контур регулювання напруги ТП.

2.5 Синтез послідовного коректуючого пристрою контура струму

Контур струму в більшості систем підлеглого регулювання є внутрішнім контуром. Тому розрахунок параметрів і їх остаточний вибір при експериментальному доведенні багатоконтурної системи починають з контура струму. Оскільки швидкодія системи в цілому визначається внутрішнім контуром, то, отже, правильний розрахунок параметрів контура струму є найбільш відповідальним етапом проектування систем підлеглого регулювання.

Принципова схема контура струму з внутрішнім контуром напруги показана на рис. 2.1. Параметри елементів контура напруги і струму вибираються таким чином, що передавальна функція контура струму при безперервному струмі не відрізнятиметься від передавальної функції контура струму без вбудованого контура напруги.

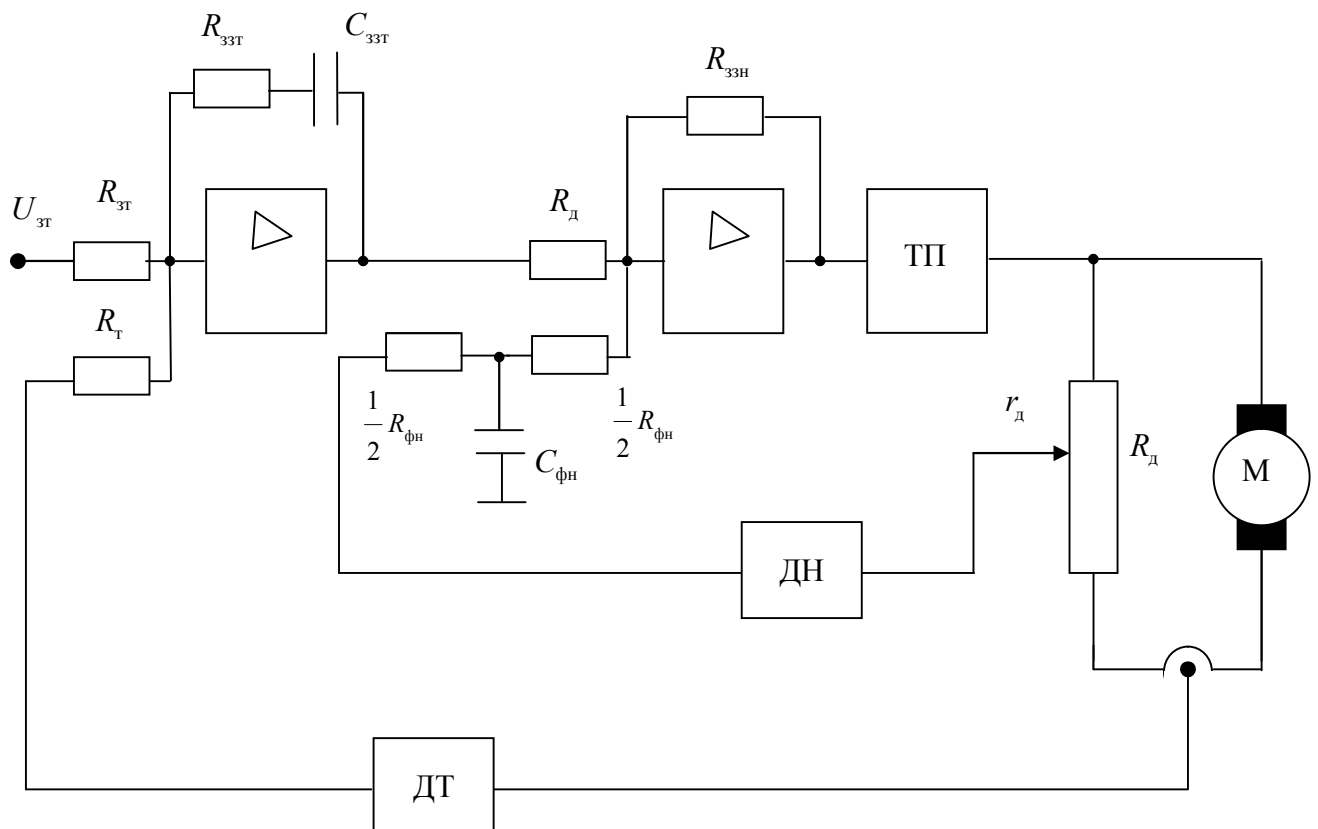


Рисунок 2.1 – Спрощена принципова схема контура струму

У режимі переривистого струму середнє значення ЕРС тиристорного перетворювача рівна ЕРС двигуна, тому, якщо прийняти останню величину постійною, як завжди робиться при розгляді контура струму, то можна вважати, що зворотний зв'язок по напрузі зникає і в цьому режимі коефіцієнт посилення регулятора зростає, що приводить до поліпшення динамічних показників системи в режимі переривистих струмів.

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

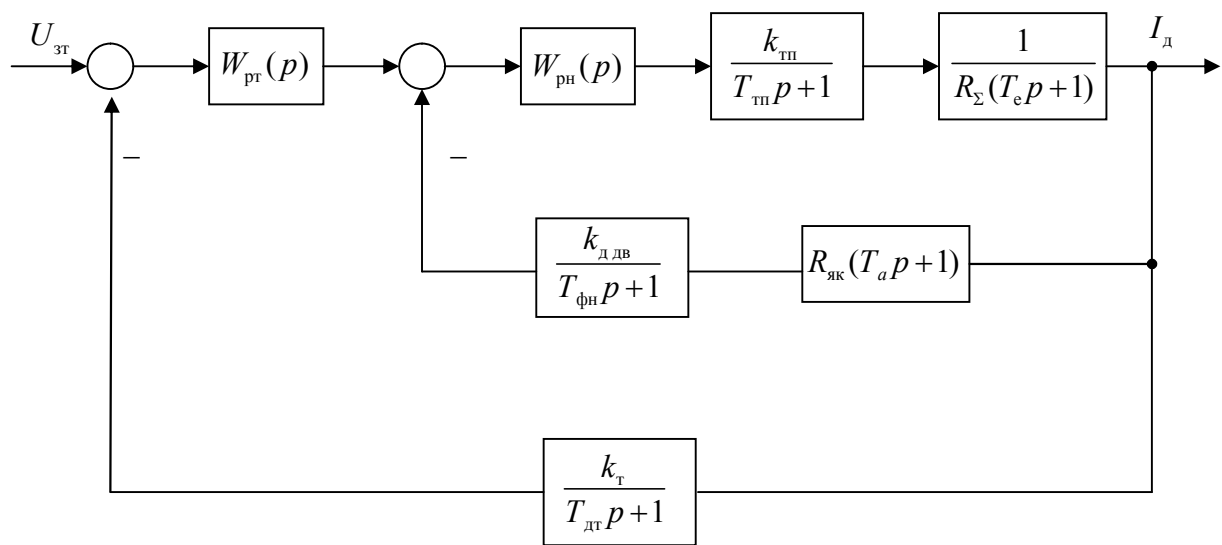


Рисунок 2.2 – Алгоритмічна схема контура струму

Схема складається з наступних елементів.

1. Тиристорний перетворювач, що є інерційною ланкою з коефіцієнтом посилення $k_{тп}$ і малою постійною часу $T_{тп}$, яка складається з середньостатистичного запізнювання перетворювача і постійної часу системи фазового управління.

2. Якірний ланцюг електродвигуна, що є інерційною ланкою з коефіцієнтом посилення $1/R_{\Sigma}$ і великою постійною часу T_e .

3. Датчик струму, що є інерційною ланкою з коефіцієнтом посилення $k_{дт}$ і малою постійною часу $T_{тп}$, яка складається з постійної часу власне датчика струму і постійної часу фільтру на його вході.

Для отримання сигналу зворотного зв'язку по струму використовується приладовий шунт, тобто шунт, встановлюється для амперметра. Приймаємо номінальний струм шунта $I_{нш}$ рівним максимальному значенню струму робочого перевантаження двигуна, $I_{нш} = I_{\max} = 2,5I_n$, падіння напруги на шунті при номінальному струмі $U_{нш} = 75 \cdot 10^{-3} \text{ В}$.

Коефіцієнт посилення шунта

$$k_{ш} = \frac{U_{нш}}{I_{нш}},$$

$$k_{ш} = \frac{75 \cdot 10^{-3}}{2,5 \cdot 350} = 0,0857 \cdot 10^{-3} \text{ В / А}.$$

Коефіцієнт посилення датчика струму визначуваний таким чином

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

$$k_{дт} \leq \frac{24}{0,0857 \cdot 10^{-3} \cdot 2,5 \cdot 350} = 319,7 \text{ В / А}.$$

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

Передавальна функція датчика току спільно з шунтом

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

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

$$k_{т} = k_{дт} \cdot k_{ш} = 320 \cdot 0,0857 \cdot 10^{-3} = 0,0274 \text{ В / А}.$$

4. Дільник напруги. Сигнал, пропорційний напрузі двигуна, знімається з дільника напруги R_d , коефіцієнт посилення якого визначається з умови отримання сигналу 24 В на вході датчика напруги при максимальній напрузі двигуна

$$k_{д дв} = \frac{24}{U_{дв \max}}.$$

Приймаючи $U_{дв \max} = U_n = 220 \text{ В}$, отримуємо

$$k_{д дв} = \frac{24}{440} = 0,109$$

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

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

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

$$W_{рн}(p) = k_{рн} = \frac{R_{зпн}}{R_{зн}}.$$

Визначимо передавальну функцію замкнутого контура напруги

$$W_n(p) = \frac{\frac{k_{рн} \cdot k_{тп}}{(T_{тп}p + 1)R_{\Sigma}(T_{\epsilon}p + 1)}}{1 + \frac{k_{рн} \cdot k_{тп}}{(T_{тп}p + 1)R_{\Sigma}(T_{\epsilon}p + 1)} \cdot \frac{k_{д дв} R_{як} (T_a p + 1)}{(T_{фн}p + 1)}} = \frac{k_{рн} k_{тп} (T_{фн}p + 1)}{[k_{д дв} k_{рн} k_{тп} R_{як} (T_a p + 1) + R_{\Sigma} (T_{фн}p + 1)(T_{\epsilon}p + 1)](T_{тп}p + 1)}.$$

Якщо прийняти $T_{фн} = T_a$, то передавальна функція замкнутого контура напруги прийме вид

$$W_n(p) = \frac{k_{рн} k_{тп}}{[k_{д дв} k_{рн} k_{тп} R_{як} + R_{\Sigma} (T_{\epsilon}p + 1)](T_{тп}p + 1)}.$$

Введемо позначення для коефіцієнта посилення контура напруги

$$k_{\text{н}} = k_{\text{рп}} \cdot k_{\text{тп}} \cdot k_{\text{д дв}} \frac{R_{\text{як}}}{R_{\Sigma}}.$$

Після нескладних перетворень отримуємо передатну функцію контура напруги у вигляді:

$$W_{\text{н}}(p) = \frac{k_{\text{рп}} k_{\text{тп}}}{R_{\Sigma} (1 + k_{\text{н}})} \cdot \frac{1}{\left(\frac{T_{\text{е}}}{1 + k_{\text{н}}} p + 1 \right) \cdot (T_{\text{тп}} p + 1)}. \quad (2.1)$$

Алгоритмічна схема контура струму з урахуванням (2.1) показана на рис.2.3.

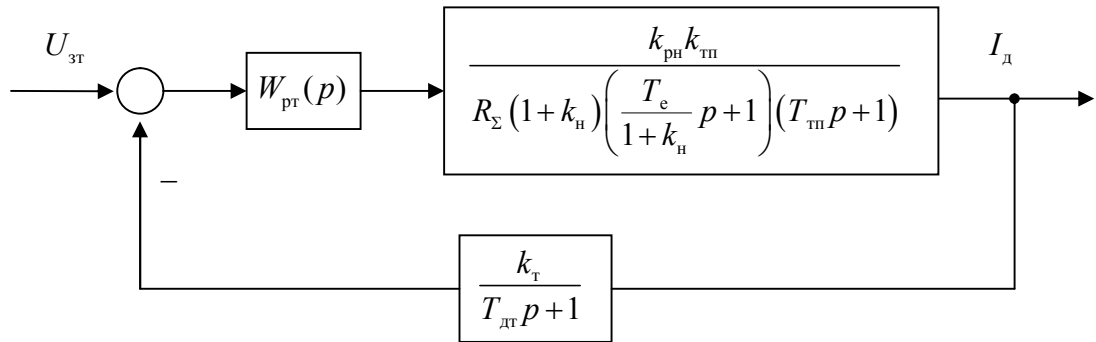


Рисунок 2.3 – алгоритмічна схема контура струму з перетвореним контуром напруги.

На рис.2.4 показана перетворена алгоритмічна схема з одиничним зворотним зв'язком. Форсуюча ланка $(T_{\text{дт}} p + 1)$ на вході контура не показана, оскільки її вплив на динамічні характеристики контура струму у край не значні зважаючи на мале значення постійної часу $T_{\text{дт}}$.

Мала некомпенсована постійна часу контура струму рівна сумі малих постійних часу

$$T_{\mu\Gamma} = T_{\text{тп}} + T_{\text{дт}}.$$

Згідно рекомендаціям [5] приймаємо $T_{\mu\Gamma} = 0,005 \text{ с}$.

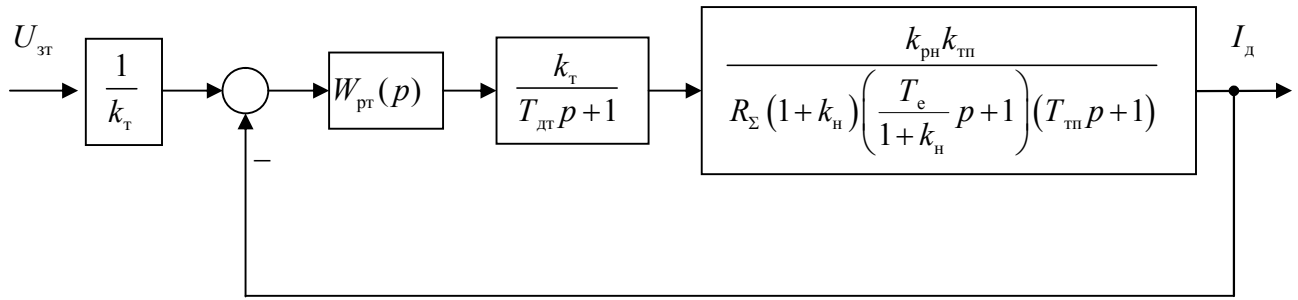


Рисунок 2.4 – Алгоритмічна схема контура струму з одиничним зворотним зв'язком

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

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

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

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

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

$$W_{\text{т}}(p) = \frac{k_{\text{рн}}k_{\text{тп}}k_{\text{т}}}{T'_{\text{пт}}pR_{\Sigma}(1 + k_{\text{н}})(T_{\mu\Gamma}p + 1)}.$$

При рівності

$$\frac{T'_{\text{пт}}R_{\Sigma}(1 + k_{\text{н}})}{k_{\text{рн}}k_{\text{тп}}k_{\text{т}}} = 2T_{\mu\Gamma}$$

отримуємо бажану передавальну функцію розімкненого контура струму.

Передавальна функція оптимізованого замкнутого контура струму згідно модульного критерію має вид:

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

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

Визначаємо параметри регуляторів контурів напруги і струму.

Для визначення коефіцієнта підсилювача П-регулятора необхідно визначити критичний коефіцієнт посилення контура напруги $k_{н\text{кр}}$, відповідний межі стійкої роботи тиристорного перетворювача

$$k_{н\text{кр}} = \frac{\pi^2 T_e T_{\tau}}{2T_0^2 \left(1 - \frac{T_{\tau}}{T_e}\right)},$$

де $T_{\tau} = 2T_{\mu\tau} = 2 \cdot 0,005 = 0,01 \text{ с}$ при настройці контура струму по модульному критерію;

$$T_0 \equiv \frac{2\pi}{m\omega_0},$$

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

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

$$k_{н\text{кр}} = \frac{3,14^2 \cdot 0,036 \cdot 2 \cdot 0,005}{2 \cdot \left(\frac{1}{300}\right)^2 \cdot \left(1 - \frac{1}{0,036}\right)} = 209$$

Прийmemo $k_{н} = 120$

Коефіцієнт посилення П-регулятора визначуваний із співвідношення

$$k_{\text{рн}} = \frac{k_{н} R_{\Sigma}}{k_{\text{тп}} k_{\text{д дв}} R_{\text{як}}},$$

$$k_{\text{рн}} = \frac{50 \cdot 0,0246}{27 \cdot 0,109 \cdot 0,0124} = 33,7.$$

Приймаємо $R_{\text{зн}} = 2 \text{ кОм}$, тоді $R_{\text{зн}} = R_{\text{зн}} \cdot k_{\text{рн}} = 2 \cdot 33,7 = 67,4 \text{ кОм}$

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

$$\frac{R_{\text{фн}}}{4} \cdot C_{\text{фн}} = T_{\text{фн}} = T_a.$$

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

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

$$C_{\text{фн}} = \frac{4 \cdot 0,067}{2 \cdot 10^3} = 13,4 \cdot 10^{-6} \text{ Ф}$$

Параметри регулятора струму визначаємо із співвідношень

$$T'_{\text{рт}} = C_{\text{ззт}} R_{\text{ззт}} = \frac{T_{\text{е}}}{1 + k_{\text{н}}}; \quad (2.2)$$

$$T'_{\text{рт}} = C_{\text{ззт}} R_{\text{зт}} = \frac{2T_{\mu\text{т}} \cdot k_{\text{рн}} \cdot k_{\text{тп}} \cdot k_{\text{т}}}{R_{\Sigma}(1 + k_{\text{н}})}. \quad (2.3)$$

З виразу (2.2) визначаємо величину $R_{\text{ззт}}$,

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

а з (2.3) – величину $R_{\text{зт}}$

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

Приймаємо $C_{\text{ззт}} = 8 \cdot 10^{-6} \text{ Ф}$, тоді

$$R_{\text{зт}} = \frac{0,031}{(1+120) \cdot 8 \cdot 10^{-6}} = 32 \cdot 10^3 \text{ Ом}$$

$$R_{\text{зт}} = \frac{2 \cdot 0,005 \cdot 33,7 \cdot 27 \cdot 0,0274}{0,0246 \cdot (1+120) \cdot 8 \cdot 10^{-6}} = 10,5 \cdot 10^3 \text{ Ом}$$

$$R_{\text{т}} = R_{\text{зт}} = 30,8 \cdot 10^3 \text{ Ом}$$

2.6 Синтез послідовного коректуючого пристрою контура ЕРС

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

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

На рис.2.5 показана принципова схема системи регулювання швидкістю із зворотним зв'язком по ЕРС двигуна.

Сигнал, пропорційний напрузі двигуна, знімається з дільника $R_{\text{д}}$ і подається на один з входів датчика ЕРС ДЕ через фільтр $R_{\text{н}}C_{\text{н}}$, що виконує функцію аперіодичної ланки з постійною часу $T_{\text{а}}$ в ланцюзі зворотного зв'язку по напрузі. На другий вхід датчика ЕРС подається сигнал струмової компенсації через опір $R_{\text{тк}}$.

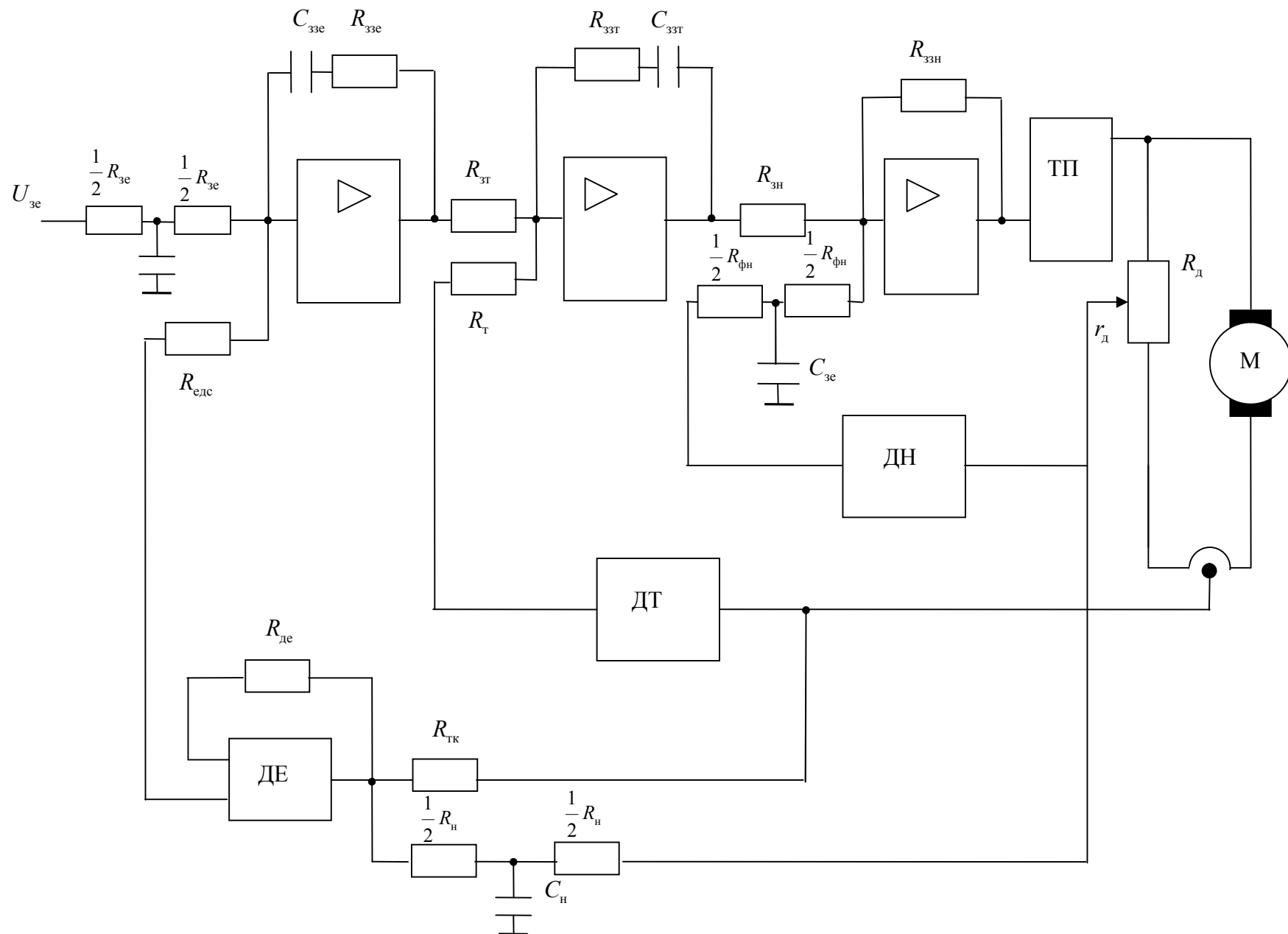


Рисунок 2.5 – Принципова схема системи управління із зворотним зв'язком по ЕРС

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

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

На рис.2.6 показана алгоритмічна схема системи.

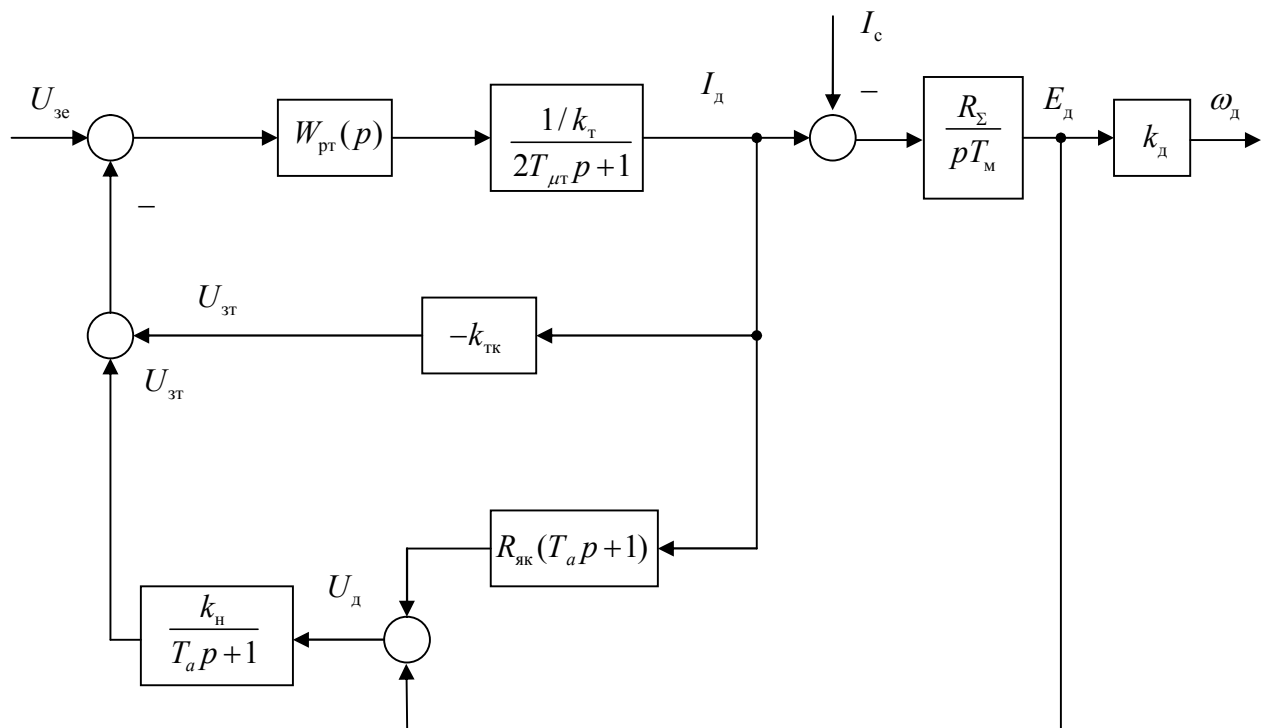


Рисунок 2.6 – Алгоритмічна схема системи регулювання швидкості із зворотним зв'язком по ЕРС

Оптимізований по модульному критерію контур струму представлений аперіодичною ланкою

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

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

Коефіцієнт посилення ланцюга струмової компенсації вибирається таким чином:

$$k_{\text{TK}} = k_{\text{H}} R_{\text{ЯК}},$$

де k_{H} – коефіцієнт посилення ланцюга зворотного зв'язку по напрузі.

При вибраному вище коефіцієнті k_{TK} на вхід регулятора ЕРС поступає сигнал, пропорційний ЕРС:

$$U_{\text{H}} - U_{\text{T}} = \frac{k_{\text{H}}}{T_a p + 1} E_{\text{д}}.$$

Алгоритмічна схема рис.2.6 шляхом еквівалентних перетворень набуває виду, показаному на рис.2.7.

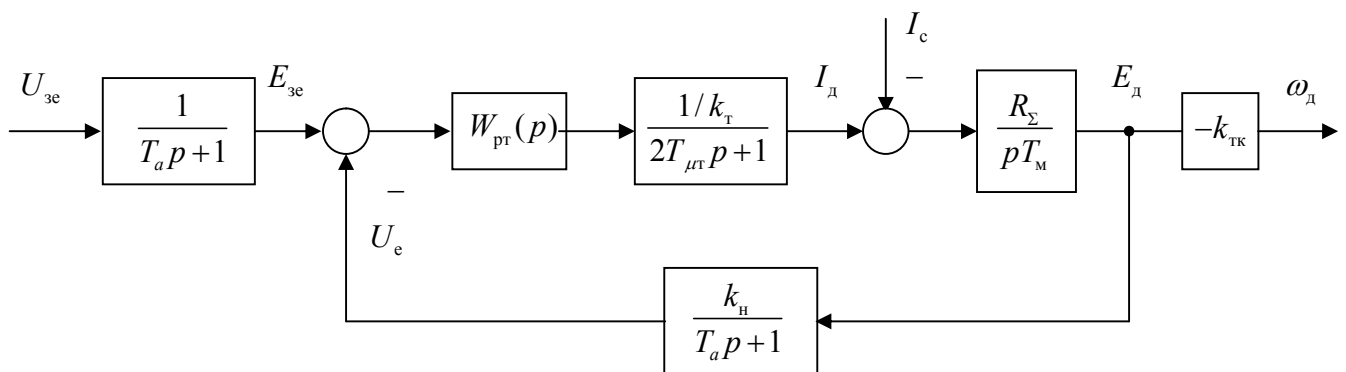


Рисунок 2.7 – Перетворена алгоритмічна схема система

На рис.2.8 показана розрахункова схема з одиничним зворотним зв'язком.

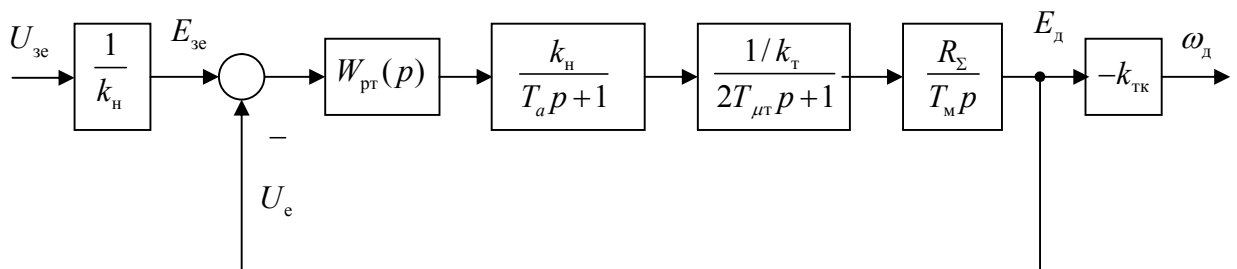


Рисунок 2.8 – Розрахункова алгоритмічна схема системи з одиничним зворотним зв'язком

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

$$W_{\text{pe}}(p) = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\epsilon} k_{\text{н}} R_{\Sigma}} \cdot \frac{1 + 4T_{\mu\epsilon} p}{4T_{\mu\epsilon} p} \quad (2.4)$$

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

$$T_{\mu\epsilon} = 2T_{\mu\Gamma} + T_a,$$

$$T_{\mu\epsilon} = 2 \cdot 0,005 + 0,067 = 0,077 \text{ с}.$$

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

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

$$k_{\text{н}} = \frac{U_{\text{зе max}}}{E_{\text{д max}}},$$

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 двигуна в режимі ідеального холостого ходу, тобто $E_{\text{д max}} = U_{\text{н}} = 220 \text{ В}$, тоді

$$k_{\text{н}} = \frac{24}{220} = 0,109.$$

Зіставляючи передавальну функцію (2.4) з передавальною функцією типового ПІ-регулятора, отримуємо наступні рівняння

$$R_{\text{зе}} C_{\text{зе}} = 4T_{\mu\epsilon}, \quad (2.5)$$

$$\frac{R_{\text{зе}}}{R_{\text{зе}}} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\epsilon} k_{\text{н}} R_{\Sigma}}. \quad (2.6)$$

Задаємося величиною ємності $C_{3\epsilon} = 2 \cdot 10^{-6} \Phi$, тоді з (2.5) отримуємо

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

$$R_{3\epsilon} = \frac{4 \cdot 0,077}{2 \cdot 10^{-6}} = 154 \cdot 10^3 \text{ Ом}.$$

Із співвідношення (2.6) знаходимо

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

$$R_{3\epsilon} = \frac{2 \cdot 0,077 \cdot 0,109 \cdot 0,0246 \cdot 154 \cdot 10^3}{0,073 \cdot 0,0274} = 31,8 \cdot 10^3 \text{ Ом}.$$

Відповідно до рис. 2.5:

$$k_H = k_{\text{д.дв}} \cdot \frac{R_{3\epsilon}}{R_H}$$

звідки

$$R_H = \frac{k_{\text{д.дв}} \cdot R_{3\epsilon}}{k_H}$$

$k_{\text{д.дв}}$ – коефіцієнт посилення дільника напруги двигуна, визначений при розрахунку контура струму

$$R_H = \frac{0,109 \cdot 31,8 \cdot 10^3}{0,109} = 31,8 \cdot 10^3 \text{ Ом}$$

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

$$\frac{R_{3\epsilon}}{4} C_{3\epsilon} = \frac{R_H}{4} C_H = T_a,$$

$$C_{\text{зе}} = \frac{4T_a}{R_{\text{зе}}} = \frac{4 \cdot 0,067}{31,8 \cdot 10^3} = 8,4 \cdot 10^{-6} \Phi.$$

$$C_{\text{н}} = \frac{4T_a}{R_{\text{н}}} = \frac{4 \cdot 0,067}{31,8 \cdot 10^3} = 8,4 \cdot 10^{-6} \Phi.$$

Передавальні функції системи оптимізованою по симетричному критерію, мають вид:

$$W_{\omega_{\text{д}}, U_{\text{зе}}}(p) = \frac{\omega_{\text{д}}(p)}{U_{\text{зе}}(p)} = \frac{k_{\text{д}}}{k_{\text{н}}} \cdot \frac{B(p)}{A(p)};$$

$$W_{\omega_{\text{д}}, I_{\text{ст}}}(p) = \frac{\omega_{\text{д}}(p)}{I_{\text{ст}}(p)} = -\frac{8T_{\text{мє}}^2 k_{\text{д}} R_{\Sigma}}{T_{\text{м}}} \cdot \frac{p(1 + T_{\text{мє}} p)}{A(p)};$$

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

$$W_{I_{\text{д}}, I_{\text{ст}}}(p) = \frac{I_{\text{д}}(p)}{I_{\text{ст}}(p)} = \frac{B(p)}{A(p)},$$

де поліноми $A(p)$ і $B(p)$ визначаються як:

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

$$B(p) = 4T_{\text{мє}} p + 1.$$

Алгоритмічна схема системи показана на рис.2.9.

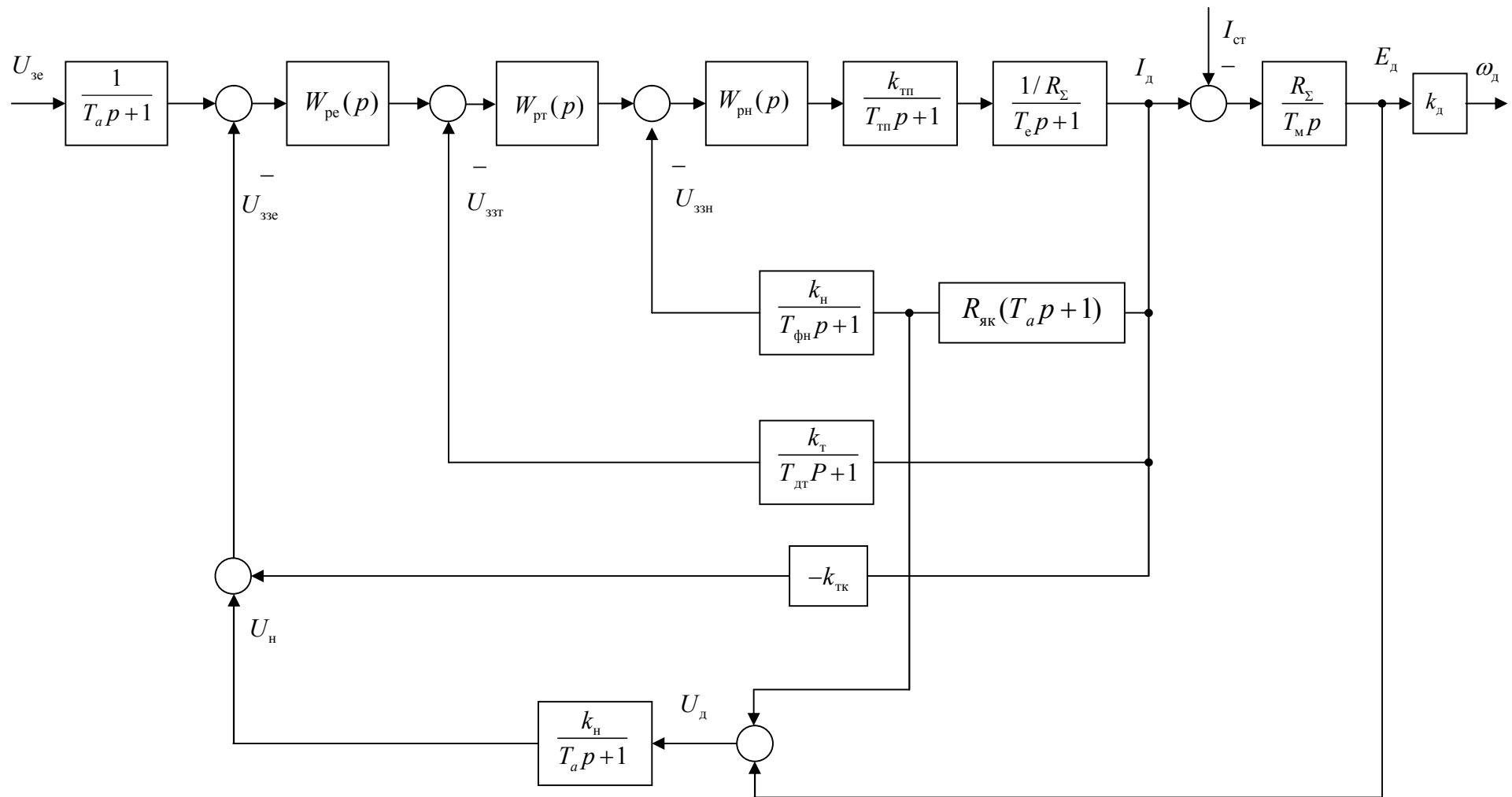


Рисунок 2.9 – Алгоритмічна схема системи

2.7 Дослідження динамічних характеристик системи

Останнім етапом розробки системи управління є вивчення її динамічних характеристик за допомогою обчислювальної техніки. Для цих досліджень використовувався пакет прикладних програм MATLAB, який дозволяє обчислити перехідні процеси через передавальні функції або розв'язати рівняння стану системи. Також можна виконати моделювання з використанням розробленої системи у режимі SIMULINK..

На рис. 2.9 показана схема системи, яка використовувалася для моделювання. На рис. 2.10-2.11 приведені графіки перехідних процесів швидкості ω_d і струму I_d по задаючій і збурюючій діях.

При настройці контура ЕРС по симетричному критерію є значне перерегулювання швидкості, рівне 43%. Для зменшення перерегулювання на вхід системи доцільно поставити фільтр з постійною часу $4T_{\mu e}$. Перерегулювання системи з фільтром на вході знижується до 8,1%. Криві, відповідні схемі з фільтром і без фільтру, позначені на рис. 2.10 цифрами 1 і 2 відповідно.

Аналогічні графіки були отримані по передавальних функціях і за допомогою рівнянь стану. Файл початкових даних, який використовувався для розрахунку, приведений в додатку Д.

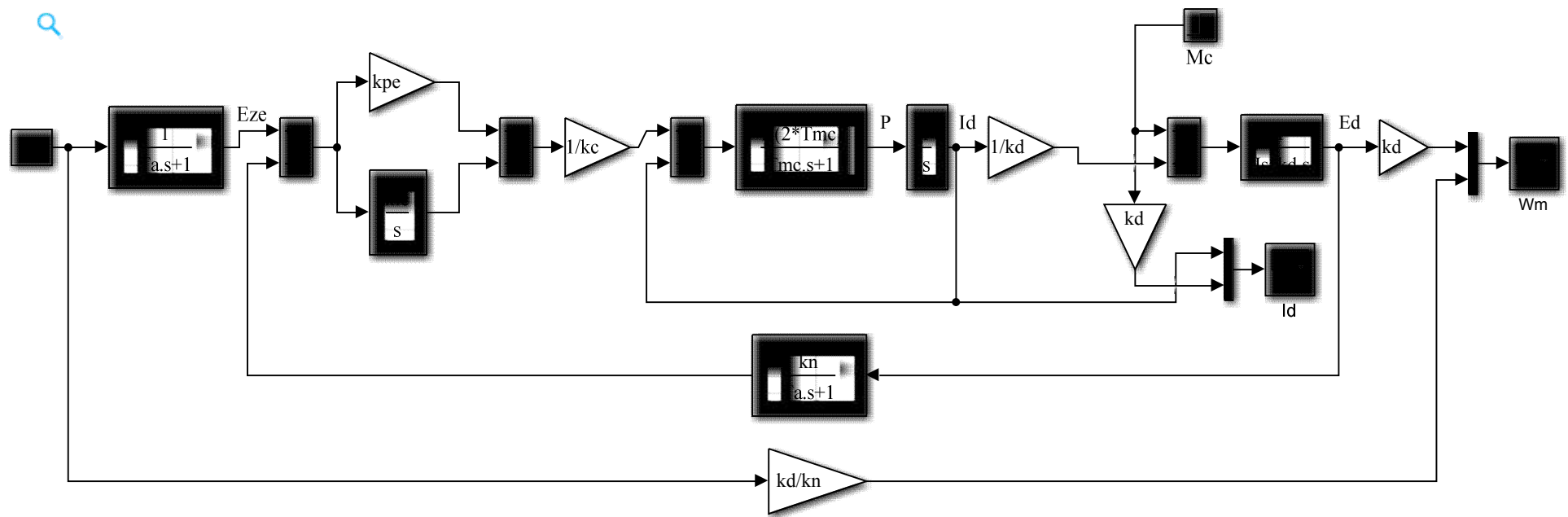


Рисунок 2.9 – Схема системи, розроблена на ЕОМ

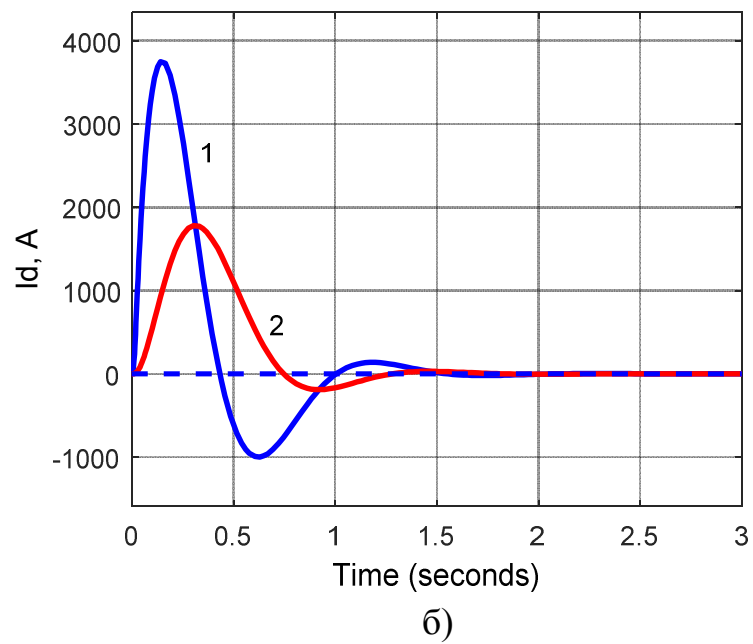
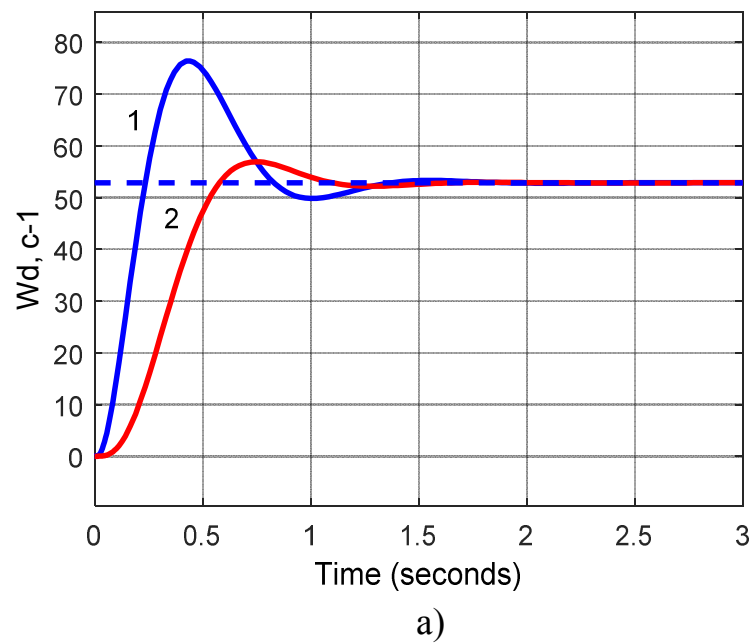
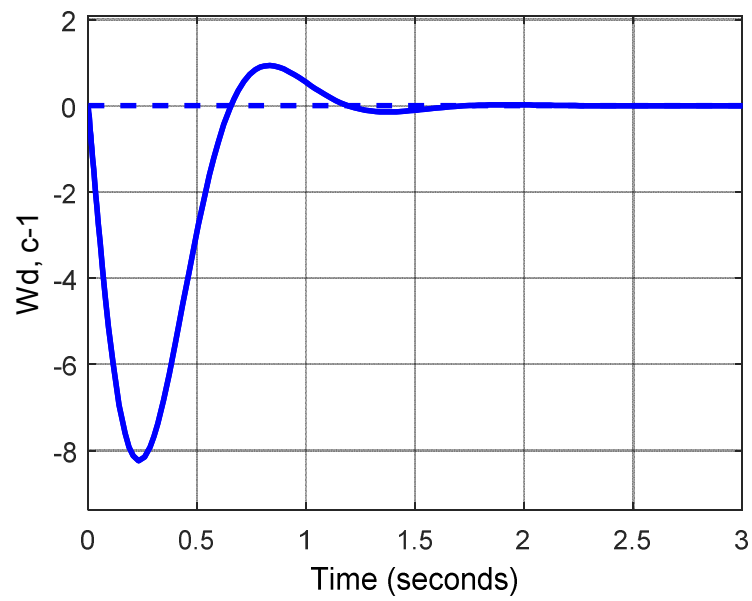
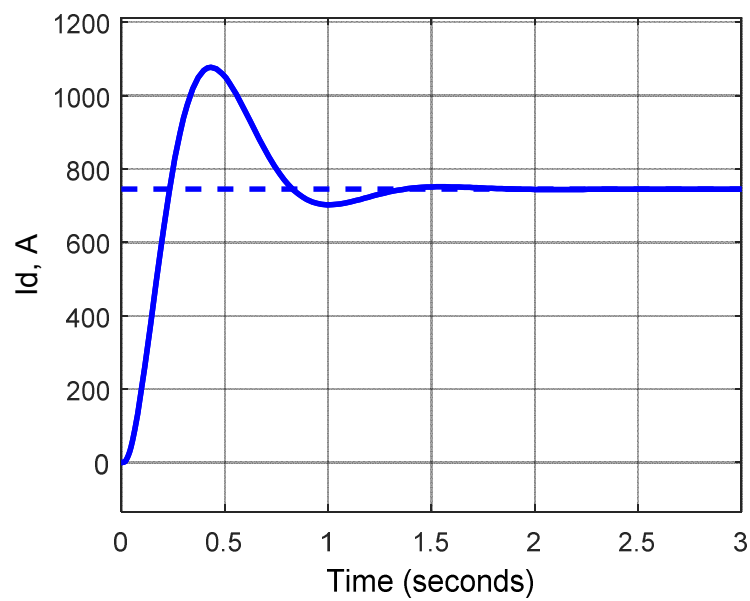


Рисунок 2.10– Графіки перехідних процесів швидкості $\omega_d = f(t)$ а)
і струму $I_d = f(t)$ б) по задаючій дії



a)



б)

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

ВИСНОВКИ ДО РОЗДІЛУ 2

1. На підставі аналізу технічних характеристик допоміжних механізмів прокатних станів і умов експлуатації електроприводів рольгангів вибраний електропривод постійного струму, виконаний за системою ТП–Д. Для управління координатами електроприводу застосована система підлеглого регулювання.

2. Виконана перевірка двигуна за умовами нагріву і перевантажувальній здатності при роботі електроприводу в різних режимах, передбачених технологічним процесом. Отримане при розрахунку значення еквівалентного моменту показує, що розраховані значення пускового і гальмівного моментів забезпечують інтенсивне протікання перехідних процесів при одночасному допустимому навантаженні електродвигуна по нагріву. Оскільки електродвигун допускає короткочасне перевантаження $(2 \div 2,5)M_n$, то по перевантажувальній здатності двигун також задовольняє вимогам.

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

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

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

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

РОЗДІЛ 3

ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

Дослідження динамічних характеристик системи управління електроприводом рольганга проведено в припущенні нескінченної жорсткості всіх зв'язків між елементами механічної частини системи. 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 дії моменту двигуна або збурюючої дії(моменту статичного навантаження), що управляє, викликає коливання зв'язаних мас, що збільшують максимальні навантаження зв'язків і ускладнюють точність обробки необхідних рухів і переміщень.

Податливість механізму обертання рольганга пов'язана з кінцевою жорсткістю зв'язку між двигуном і рольгангом. Така система може розглядатися як двомасова (рис. 3.1).

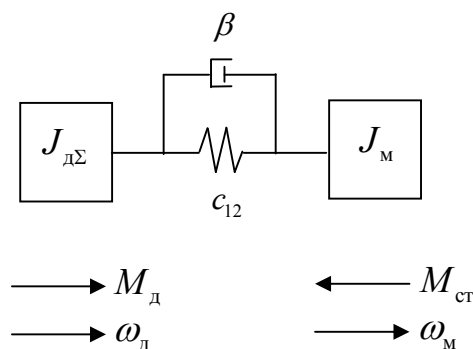


Рисунок 3.1 – Двомасова розрахункова схема

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

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

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

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

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

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

β – коефіцієнт внутрішнього в'язкого тертя 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 контуро ЕРС, настроєним по симетричному критерію. Алгоритмічна схема двомасової системи показана на рис. 3.2.

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

$$\begin{cases} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}. \end{cases} \quad (3.3)$$

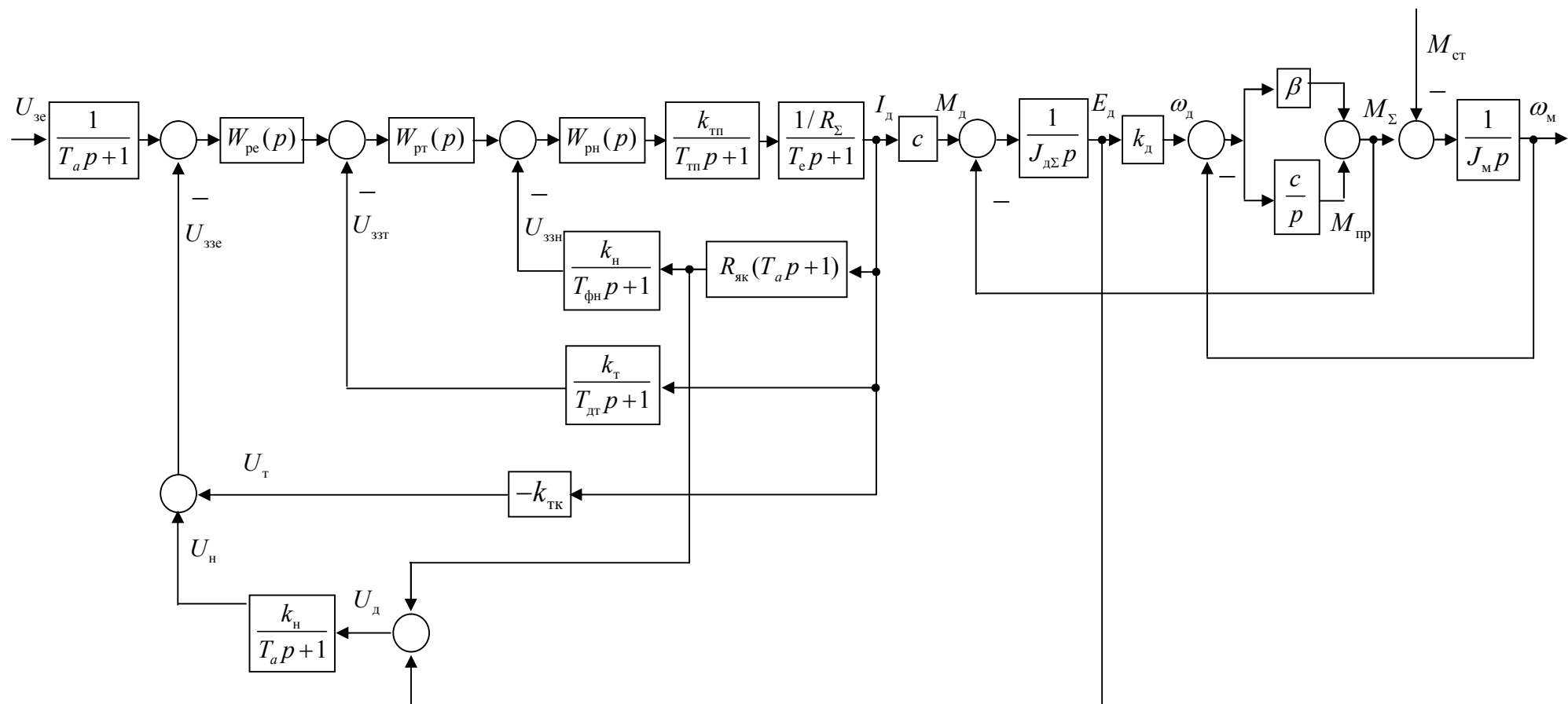


Рисунок 3.2 – Алгоритмічна схема двомасової системи

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:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{ззе}} - \frac{k_{\text{д}}}{2T_{\mu\Gamma}^2} M_{\text{д}} - \frac{1}{T_{\mu\Gamma}} P + \frac{1}{T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{іе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}. \end{array} \right. \quad (3.4)$$

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:

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

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

При записі системи (3.4) замість T_m записано відповідний вираз

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

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

$$\left\{ \begin{array}{l} \frac{d\omega_m}{dt} = \frac{1}{J_m} M_{\text{пр}} + \frac{\beta}{J_m} \omega_d - \frac{\beta}{J_m} \omega_m - \frac{1}{J_m} M_{\text{ст}}; \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_d - c_{12} \omega_m; \\ \frac{d\omega_d}{dt} = \frac{1}{J_{d\Sigma}} M_d - \frac{1}{J_{d\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{d\Sigma}} \omega_d - \frac{\beta}{J_{d\Sigma}} \omega_m; \\ \frac{dM_d}{dt} = \frac{1}{k_d} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_T} E_{3e} - \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_T} U_{3e} - \frac{k_d}{2T_{\mu T}^2} M_d - \frac{1}{T_{\mu T}} P + \frac{1}{T_{\mu T}^2 k_T} U_{ie}; \\ \frac{dE_{3e}}{dt} = \frac{1}{T_a} U_{3e} - \frac{1}{T_a} E_{3e}; \\ \frac{dU_{3e}}{dt} = \frac{k_n}{T_a k_d} \omega_d - \frac{1}{T_a} U_{3e}; \\ \frac{dU_{ie}}{dt} = k_{ie} E_{3e} - k_{ie} U_{3e}. \end{array} \right. \quad (3.5)$$

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

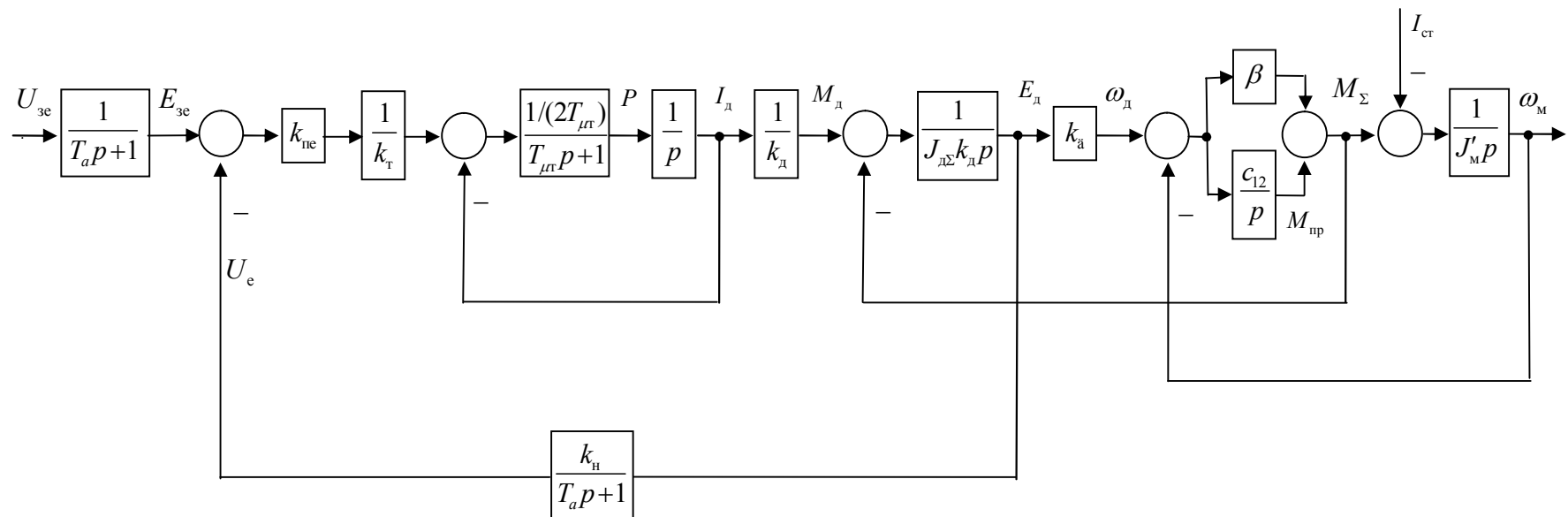


Рисунок 3.3 – Спрощена алгоритмічна схема двомасової системи

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

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

$$\vec{X}(t) = \left\{ \omega_{\text{м}}(t), M_{\text{пр}}(t), \omega_{\text{д}}(t), M_{\text{д}}(t), P(t), E_{\text{зе}}(t), U_{\text{зе}}(t), U_{\text{іє}}(t) \right\}^T,$$

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

де T – знак транспонування.

Рівняння стану системи і рівняння виходу має вид

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

де $\mathbf{A}$ – матриця стану;

$\mathbf{B}$ – матриця управління;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ – вектор виходу.

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

$$\mathbf{A} = \begin{array}{c} \begin{array}{cccccccc} \omega_{\text{м}} & M_{\text{пр}} & \omega_{\text{д}} & M_{\text{д}} & P & E_{\text{зe}} & U_{\text{ззe}} & U_{\text{ie}} \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{д}}} & 0 & 0 & 0 & 0 & 0 \\ \hline -c & 0 & c & 0 & 0 & 0 & 0 & 0 \\ \hline \frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{-k_{\text{д}}}{2T_{\text{мт}}^2} & -\frac{1}{T_{\text{мт}}} & \frac{k_{\text{pe}}}{2T_{\text{мт}}^2 k_{\text{т}}} & -\frac{k_{\text{pe}}}{2T_{\text{мт}}^2 k_{\text{т}}} & \frac{1}{2T_{\text{мт}}^2 k_{\text{т}}} \\ \hline 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & 0 & 0 \\ \hline 0 & 0 & \frac{k_{\text{н}}}{T_a k_{\text{д}}} & 0 & 0 & 0 & -\frac{1}{T_a} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & k_{\text{ie}} & -k_{\text{ie}} & 0 \\ \hline \end{array} \end{array}$$

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

3.2 Розрахунок перехідних процесів двомасової системи

Визначимо параметри, необхідні для розрахунку перехідних процесів двомасової системи,

Момент інерції двигуна і жорстко пов'язаних з ним елементів системи визначений раніше $J_{д\Sigma} = 1,1 \cdot J_{дв} = 1,1 \cdot 7 = 7,7 \text{ кг} \cdot \text{м}^2$

Приведений момент інерції механізму (завантаженого рольганга):

$$J_{\text{м}} = \frac{J_{\text{м3}}}{i^2};$$

$$J_{\text{м}} = \frac{531}{5,95^2} = 15 \text{ кг} \cdot \text{м}^2.$$

Решта параметрів визначені у попередніх розділах.

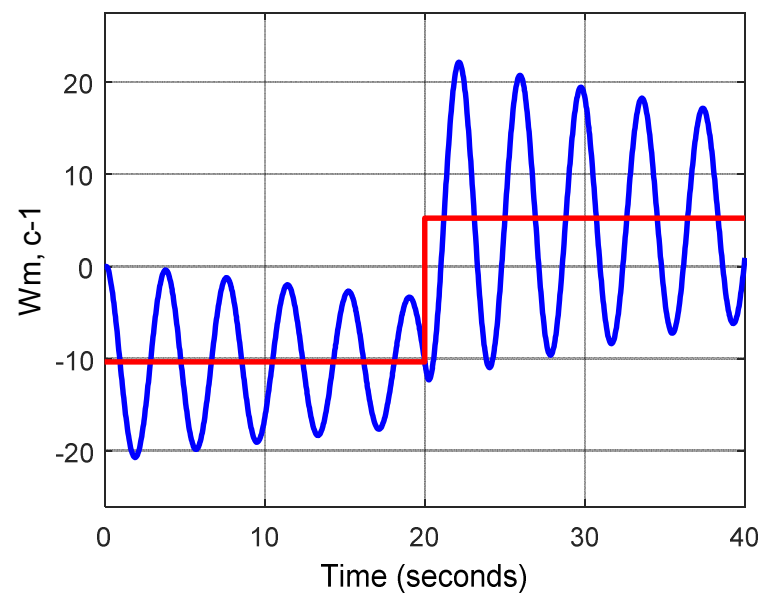
Розрахунок перехідних процесів проводився на ЕОМ з використанням пакету прикладних програм MATLAB. Для дослідження була використана модель системи, приведена на рис. 3.4, а також векторно-матричне рівняння (3.6).

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

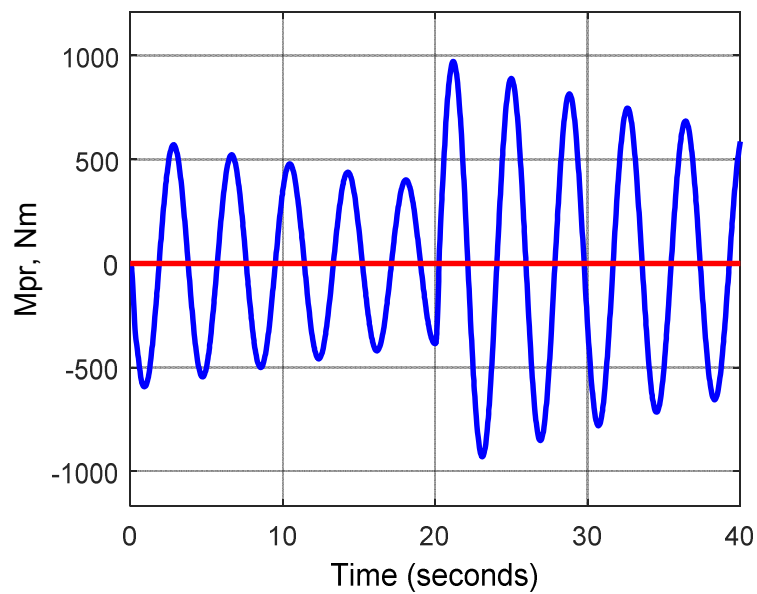
Графіки перехідних процесів двомасової системи показані на рис. 3.5-3.8. Для зменшення інформації, що виводиться, на ЕОМ були виведені на екран монітора, і відповідно роздруковані перші чотири змінні стану. Як видно з графіків, мають місце значні коливання основних координат системи, що несприятливо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



Рисунок 3.4 - Схема моделі двомасової системи, розроблена на ЕОМ

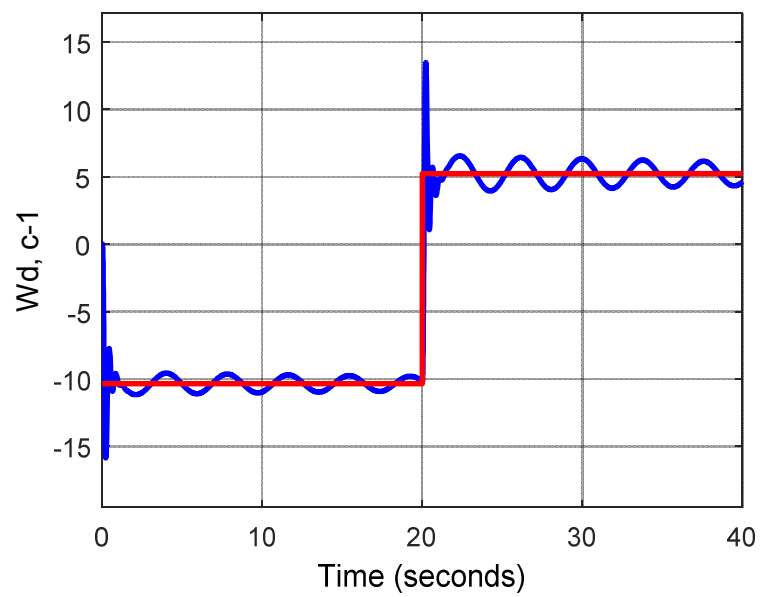


a)

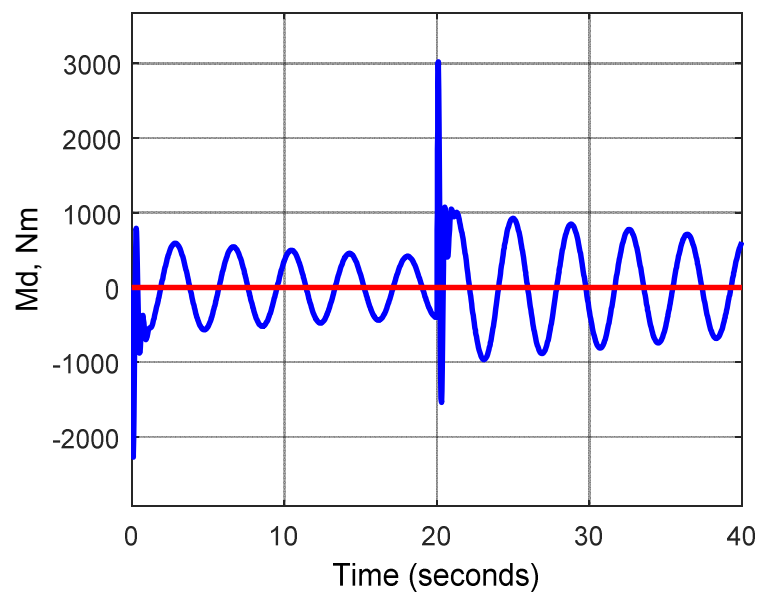


б)

Рисунок 3.5 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії



a)



б)

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

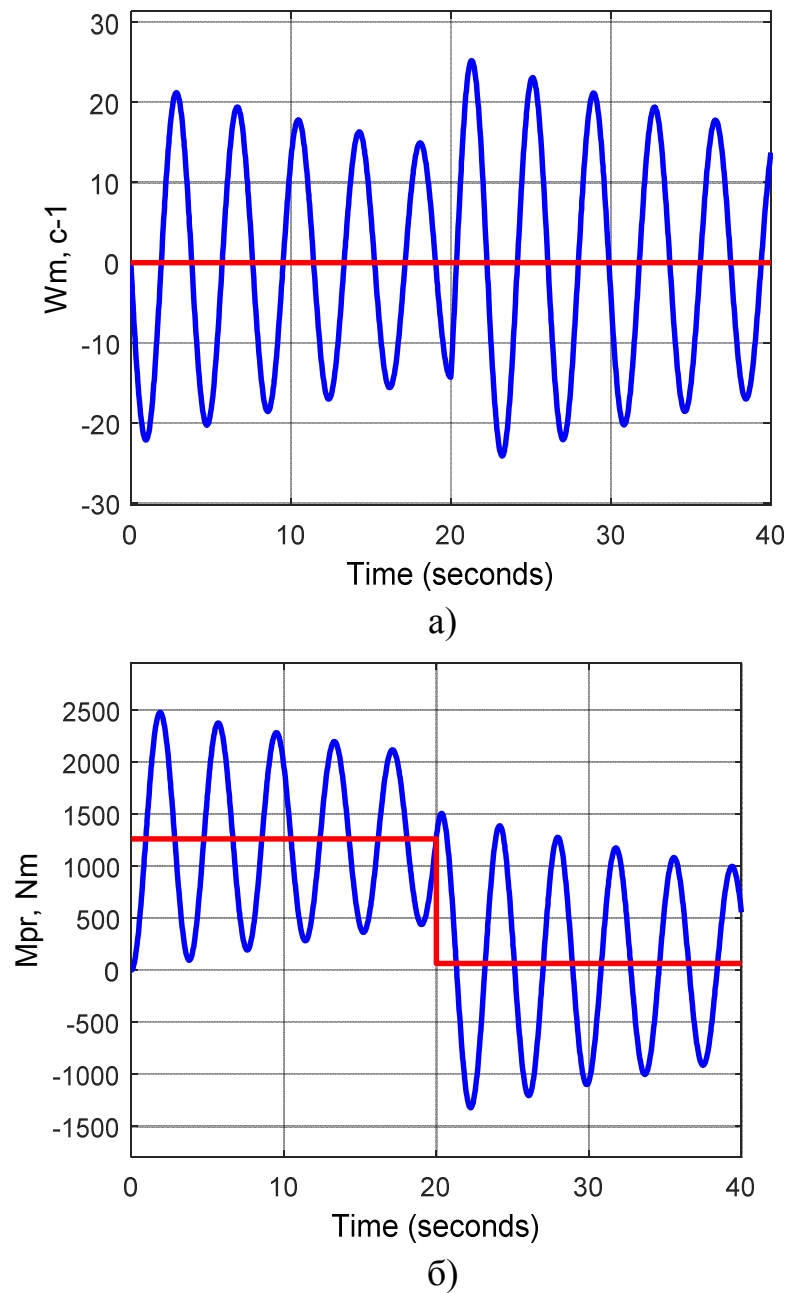
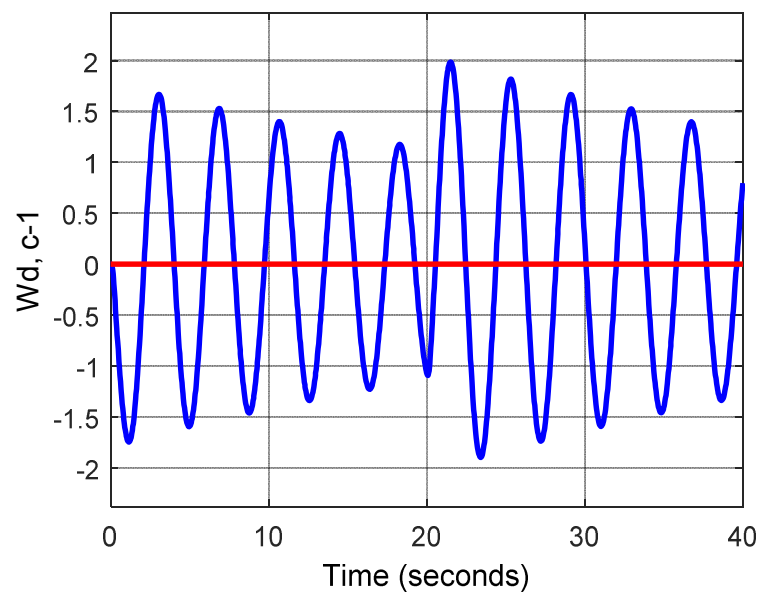
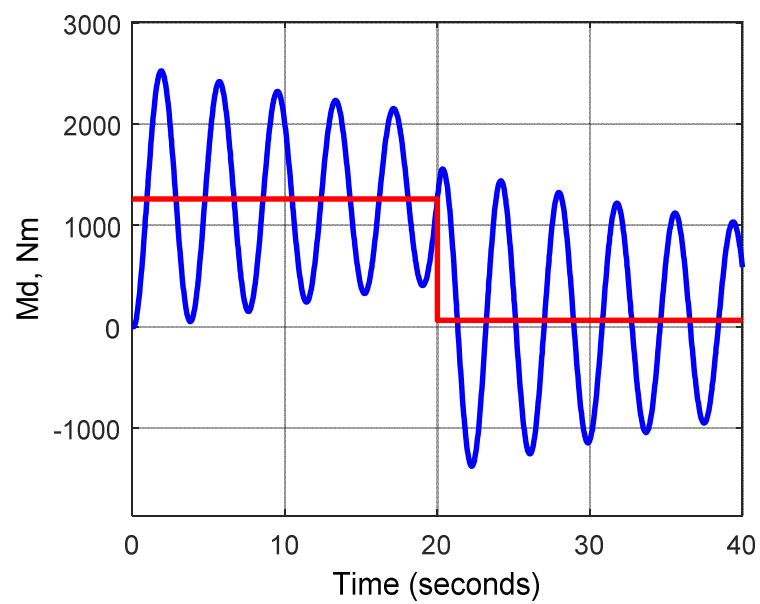


Рисунок 3.7 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по збурюючій дії



a)



б)

Рисунок 3.8 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму рольганга з урахуванням кінцевої жорсткістю зв'язку між двигуном і рольгангом. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер рухів елементів тільки в середньому, оскільки не враховує вплив пружності реальних механічних зв'язків. Внаслідок кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, додаток до якої управляючої дії або 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 механізмом рольганга з урахуванням податливості елементів механічної частини електроприводу може розглядатися як двомасова.

2. Виконано розробку математичної моделі. Отримано рівняння стану двохмасової 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 +24В до -24В, що відповідає зміні швидкості в межах номінальних значень. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи мають незадовільний (коливальний) характер.

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

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

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

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

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

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

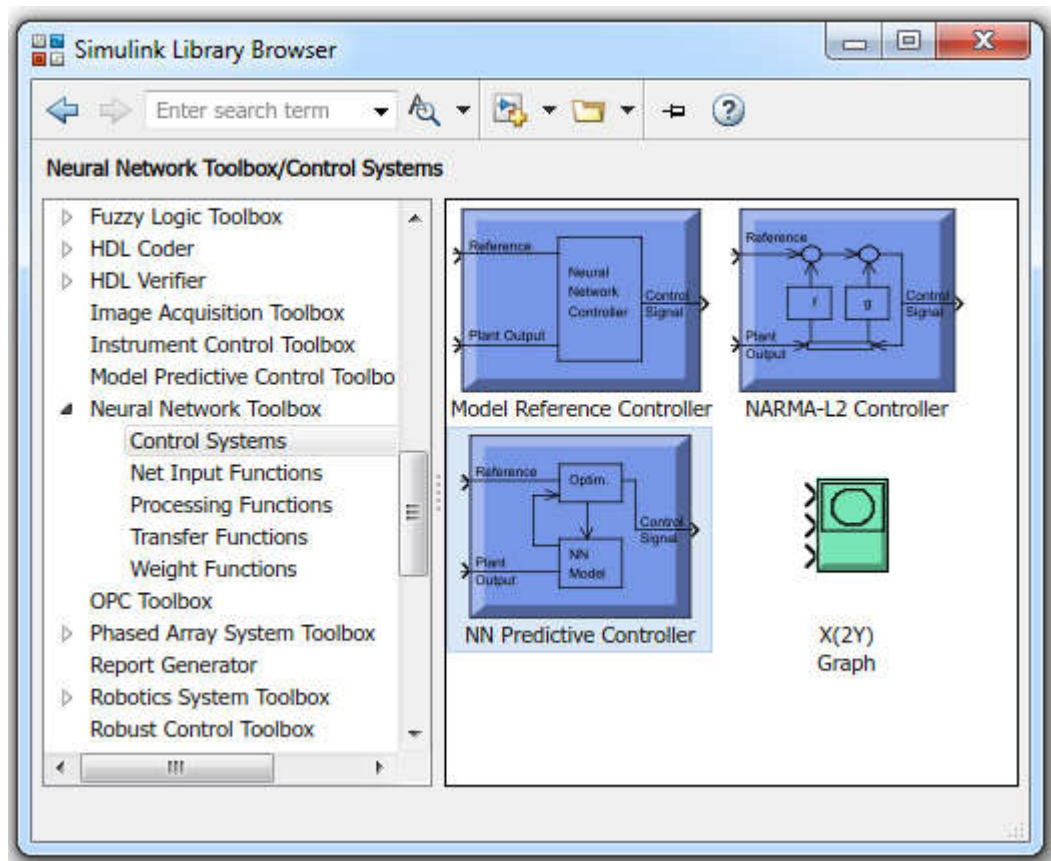


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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

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

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

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

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

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

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

4.2.1 Ідентифікація об'єкту управління.

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

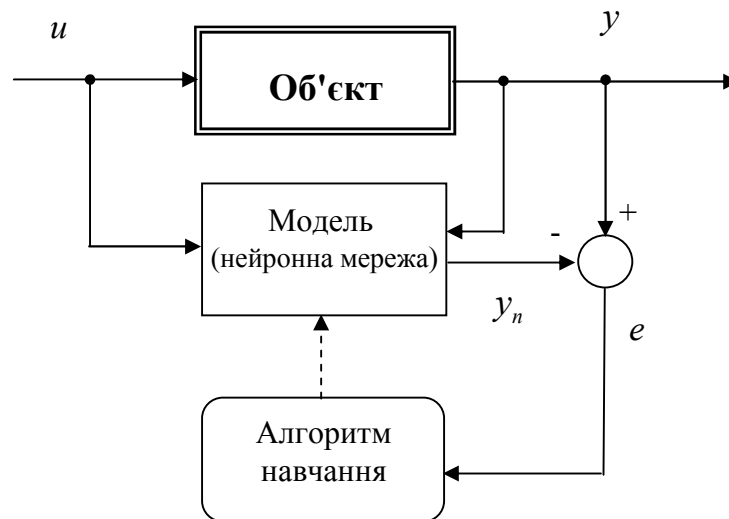


Рисунок 4.2 – Схема підсистеми ідентифікації.

Нейронна мережа, що представлена на рисунку 4.3 і використовується у регуляторі об'єкту управління, має два шари нейронів та використовує лінії 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**.

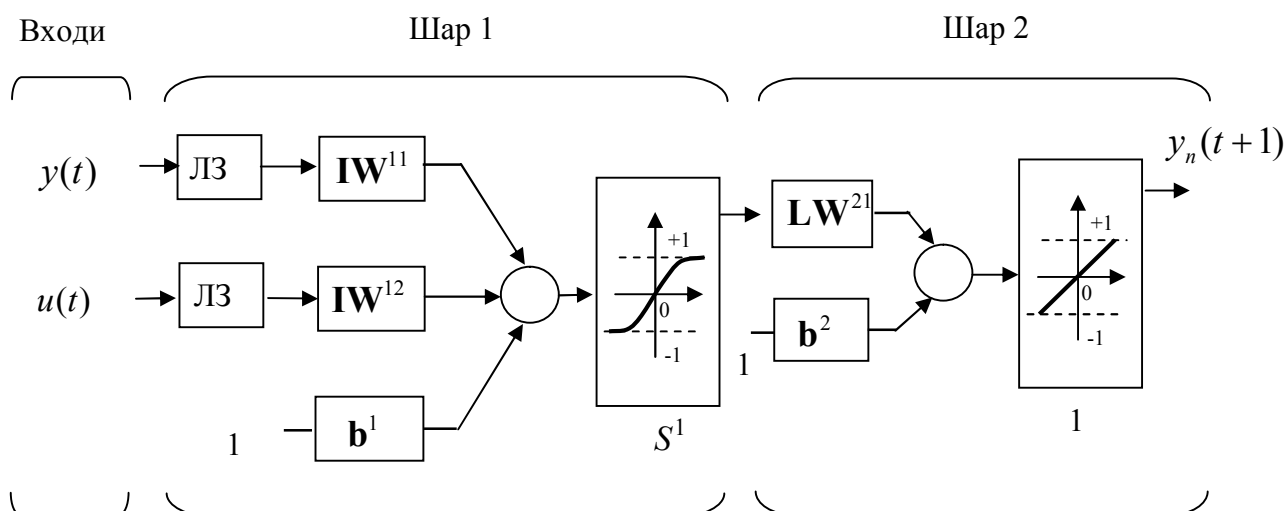


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

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 не нейромережеву модель, y_r – очікувана, а y_m – фактична реакція моделі системи. Змінна ρ визначає внесок потужності управління у критерій якості. На рисунку 4.4 показано структурну схему процесу управління з прогнозом, де регулятор складається з нейромережевої моделі керованого об'єкту і блоку оптимізації. Блок оптимізації визначає значення управління, яке мінімізує критерій якості управління, що управляє процесом.

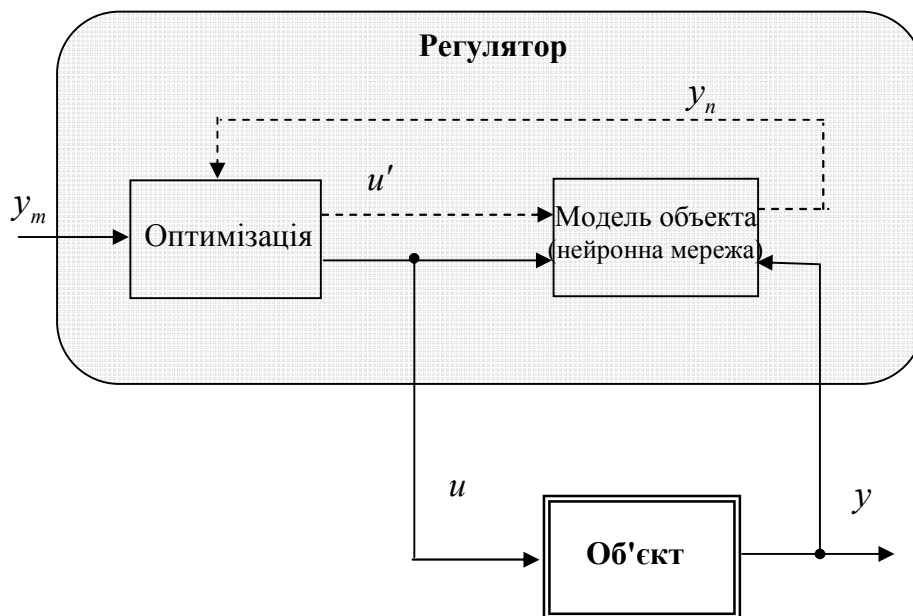


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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

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

Функції одновимірної оптимізації:

Csrchbac – пошук із зворотним прогоном;

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

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjjdjj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

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

Допоміжні функції:

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

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

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

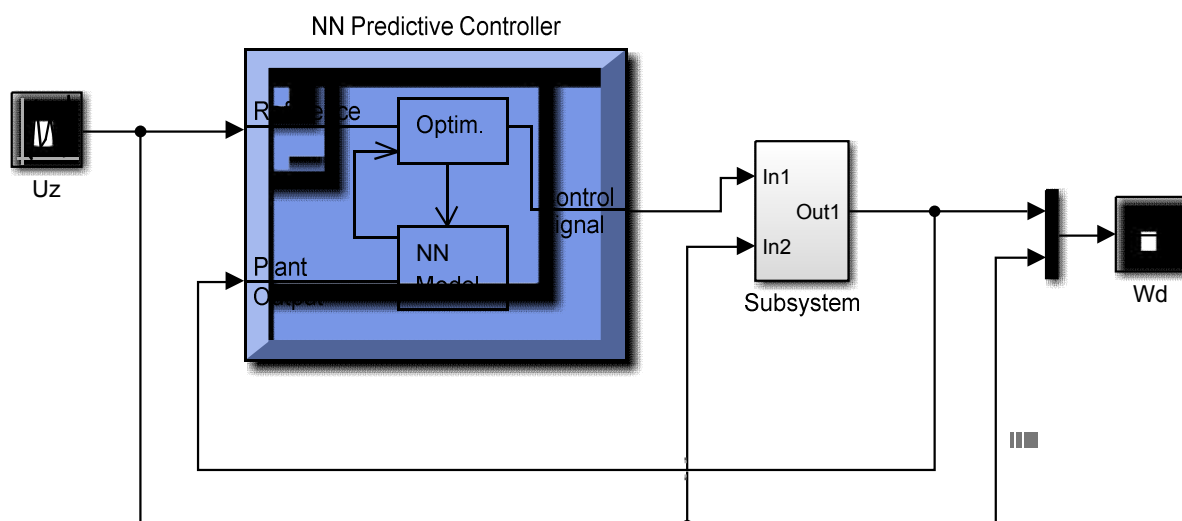


Рисунок 4.5 – Схема системи управління з нейрорегулятором
NN Predictive Controller

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

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

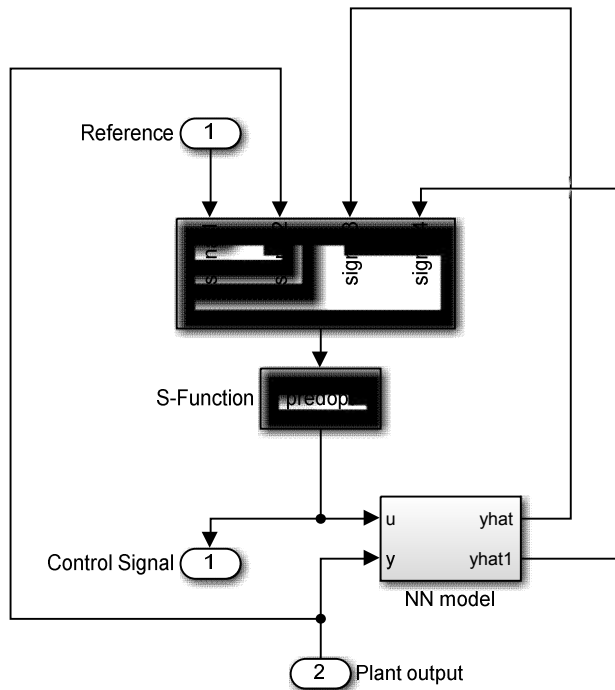


Рисунок 4.6 – Структурна схема нейрорегулятора

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

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

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

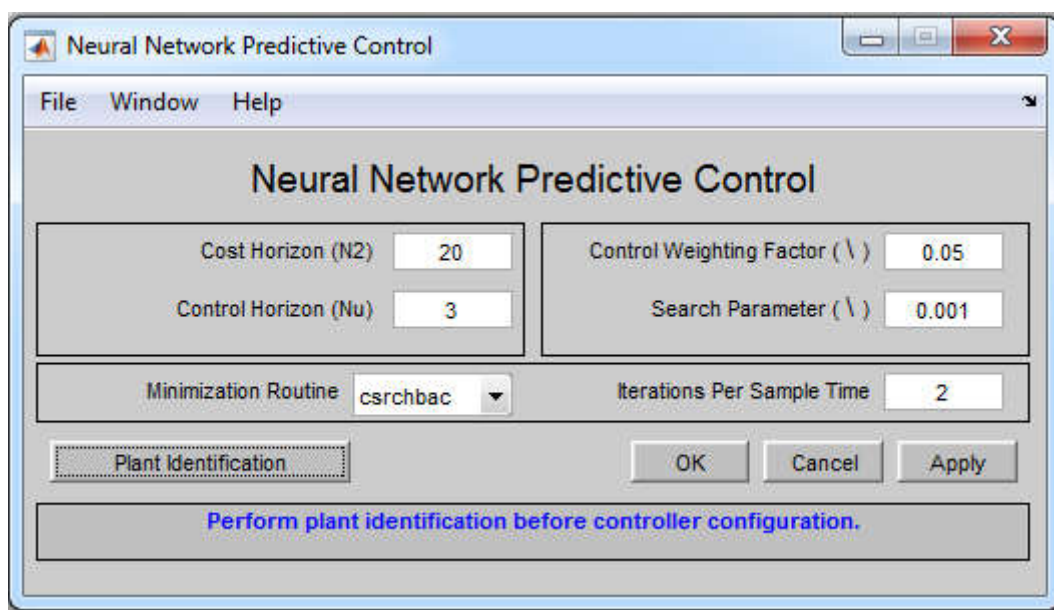


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

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

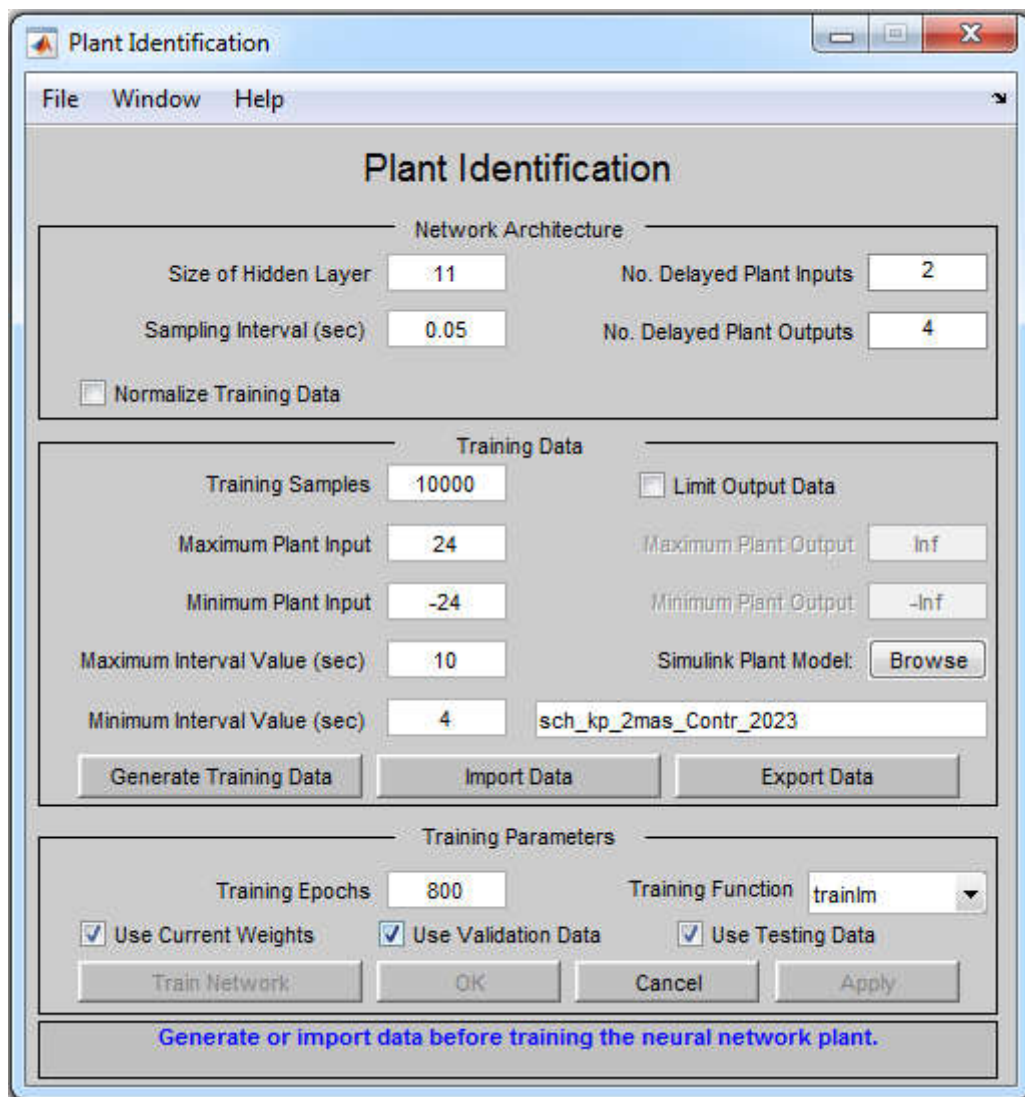


Рисунок 4.8 – Вікно ідентифікації керованої системи

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

Опис вікна Plant Identification

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Maximum Plant Output. Максимальне значення вихідного сигналу.

Minimum Plant Output. Мінімальне значення вихідного сигналу.

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

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

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

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

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

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

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

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

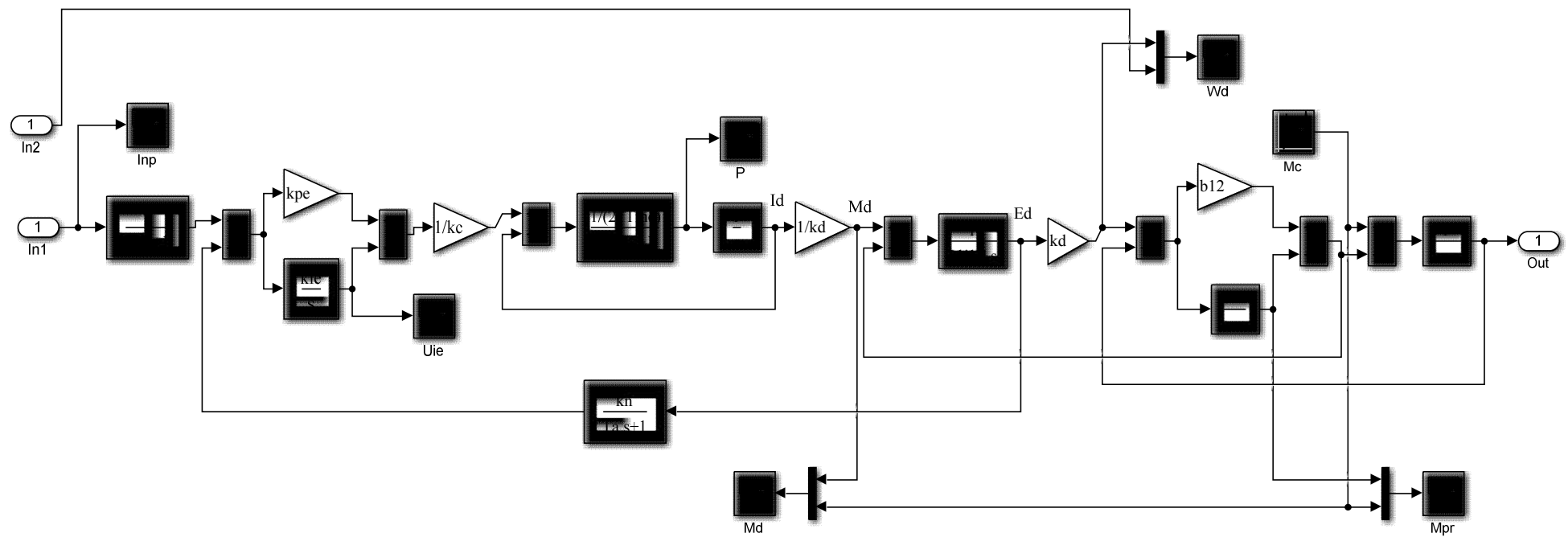


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

Секція Training Parameters (параметри навчання). Training Parameters

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

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

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

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

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

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

менших квадратів Левенберга-Марквардта; trainbr - байєсівський регулятор.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

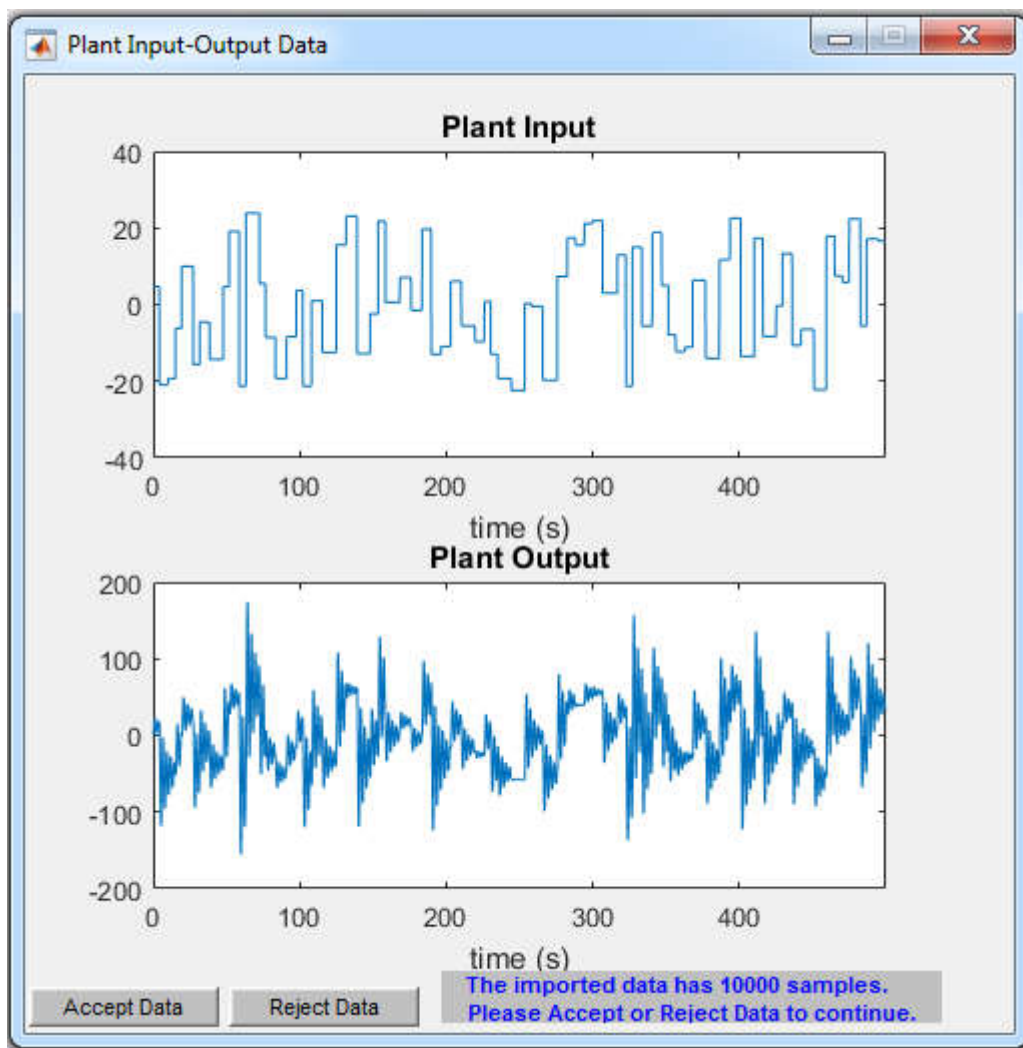


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

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

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

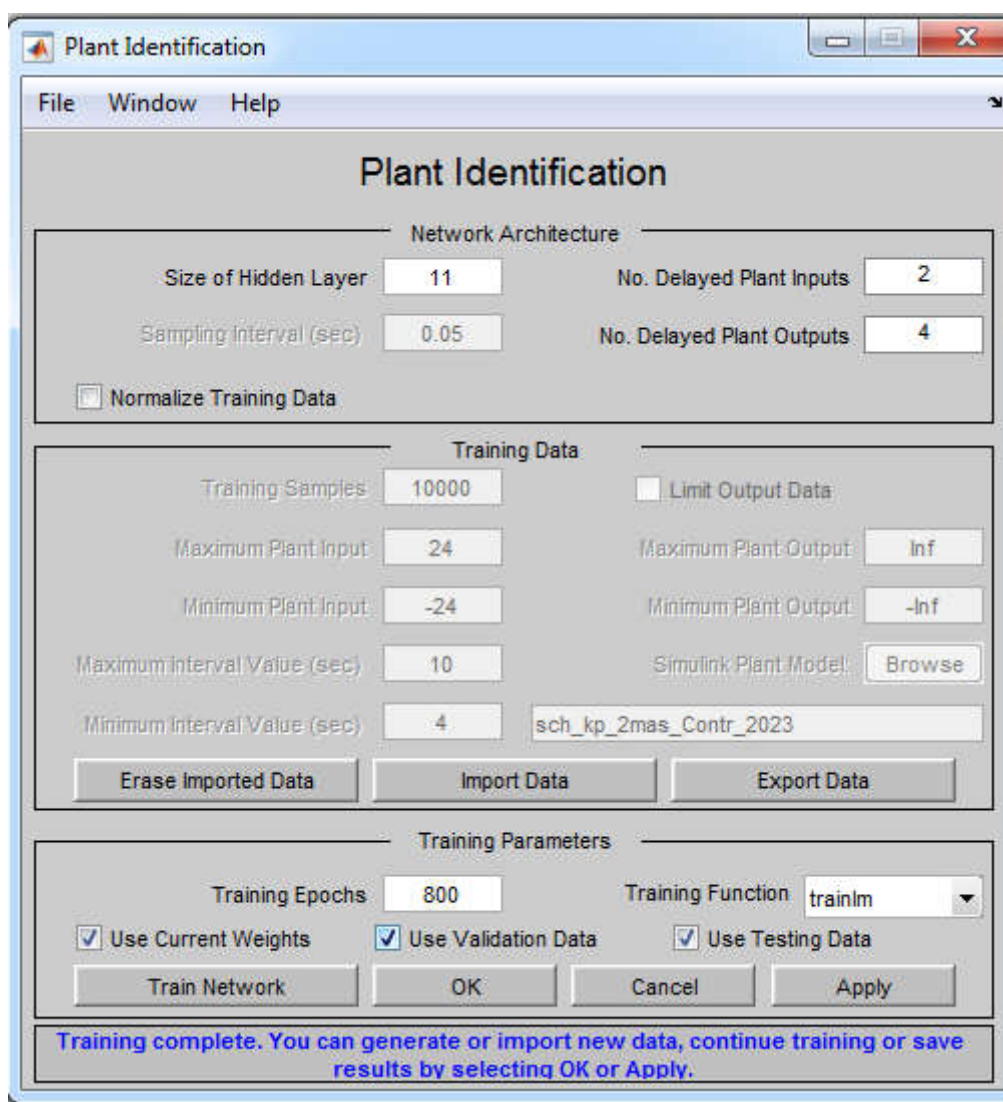


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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

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

та один нейрон у другому (вихідному) шарі. Активаційні функції: гіперболічний тангенс (tansig) у першому шарі та лінійна функція (purelin) у другому шарі.

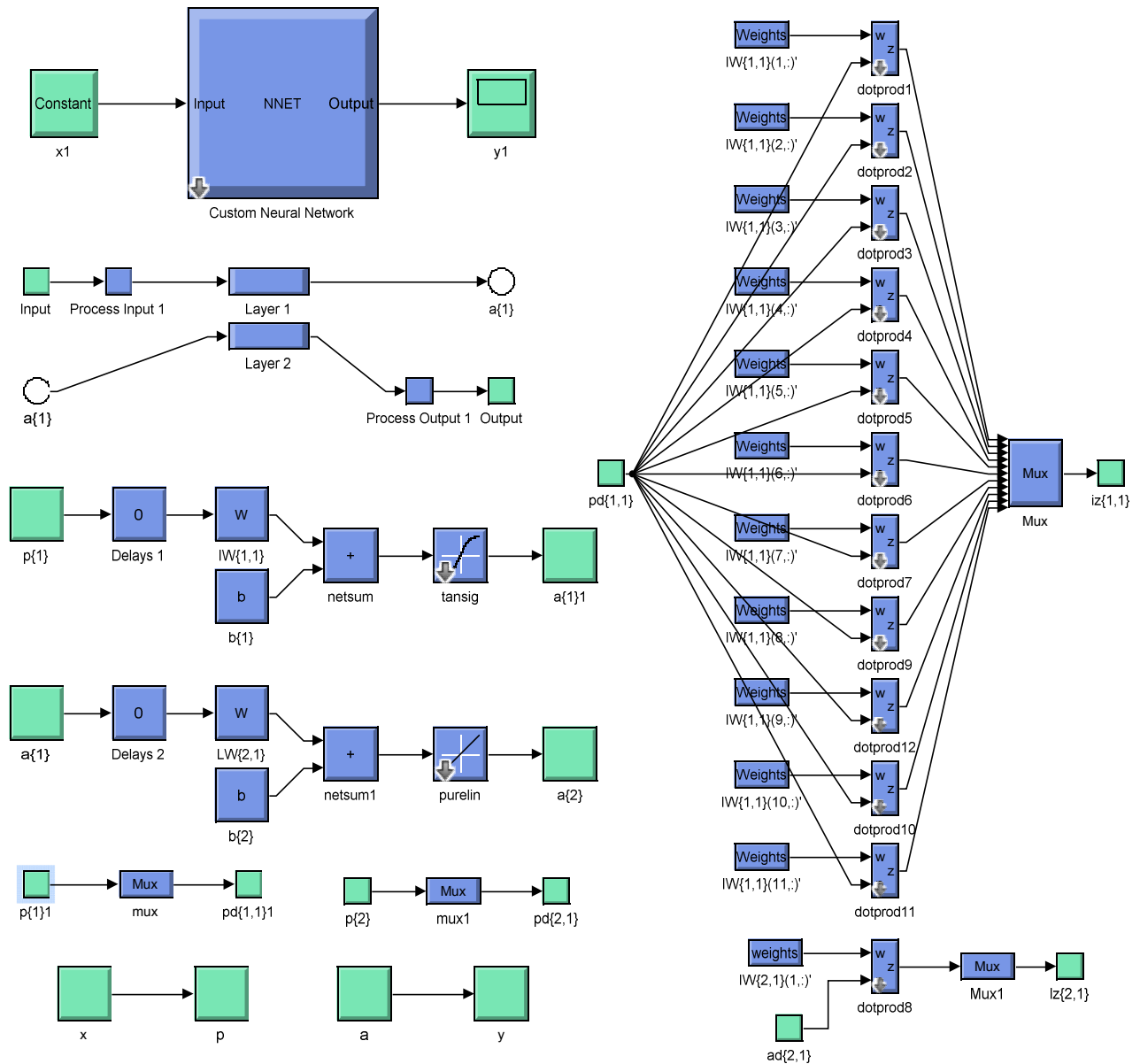


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

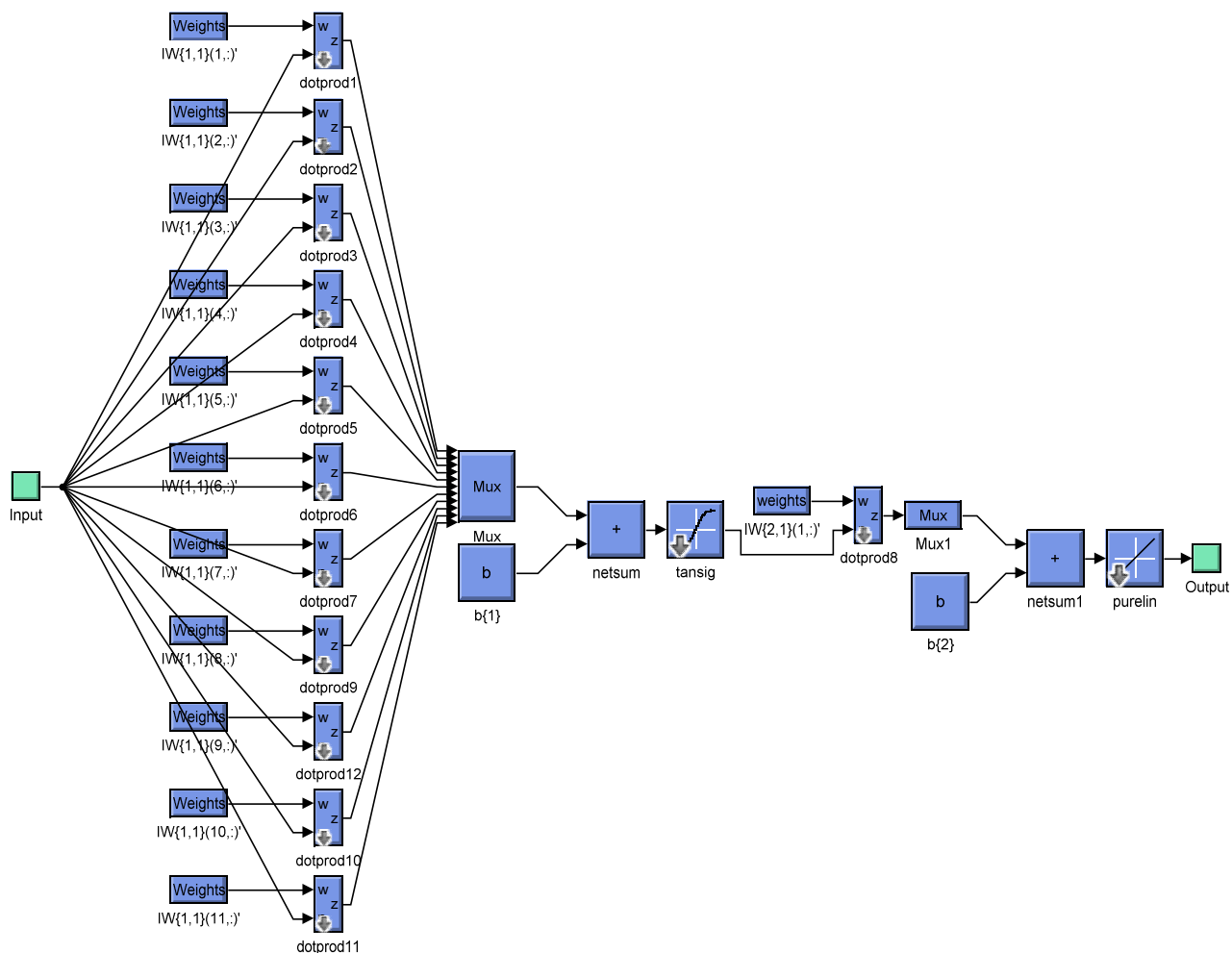


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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

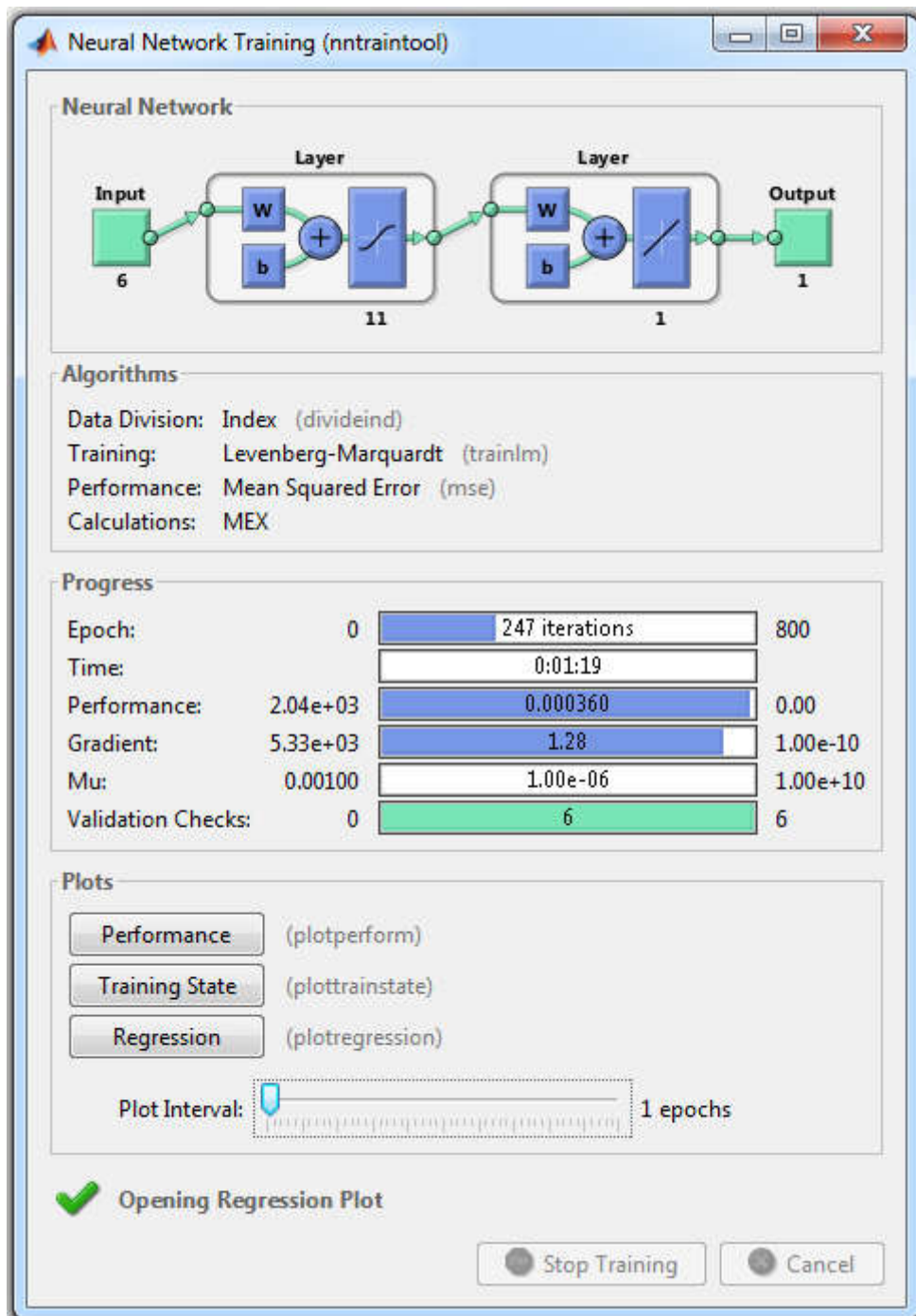


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

Вікно Neural Network Training (nntraintool) має наступні секції.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Mu. Відноситься до тау-параметра, який використовується в алгоритмах оптимізації для регулювання швидкості навчання нейронної мережі. Тау-параметр (μ) 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 μ обчислюється алгоритмом оптимізації під час навчання і може змінюватися на кожній ітерації. Цей параметр є однією з ключових частин процесу оптимізації нейронної мережі та допомагає покращити збіжність та стабільність навчання.

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

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

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

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

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

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

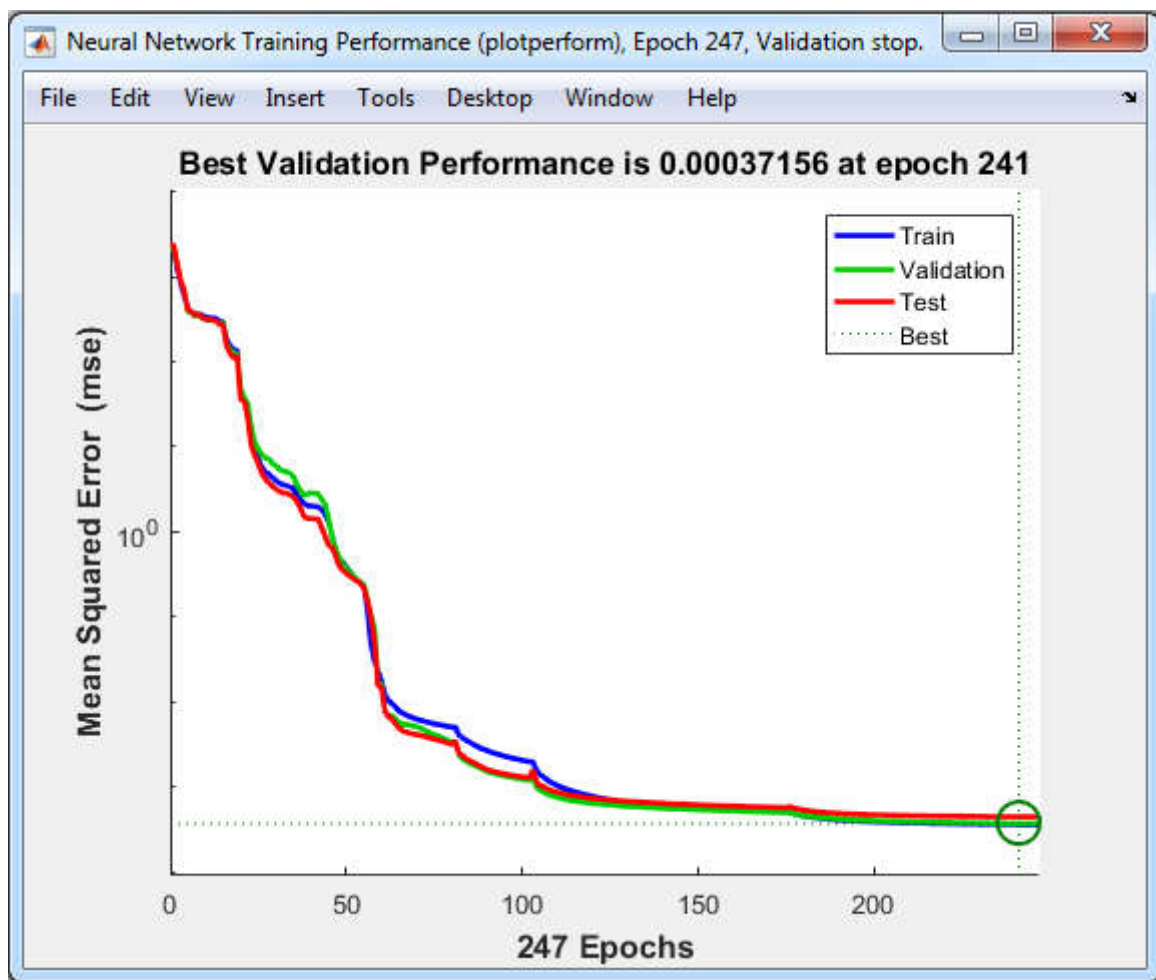


Рисунок 4.15 – Вікно контролю процесу навчання

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

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

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

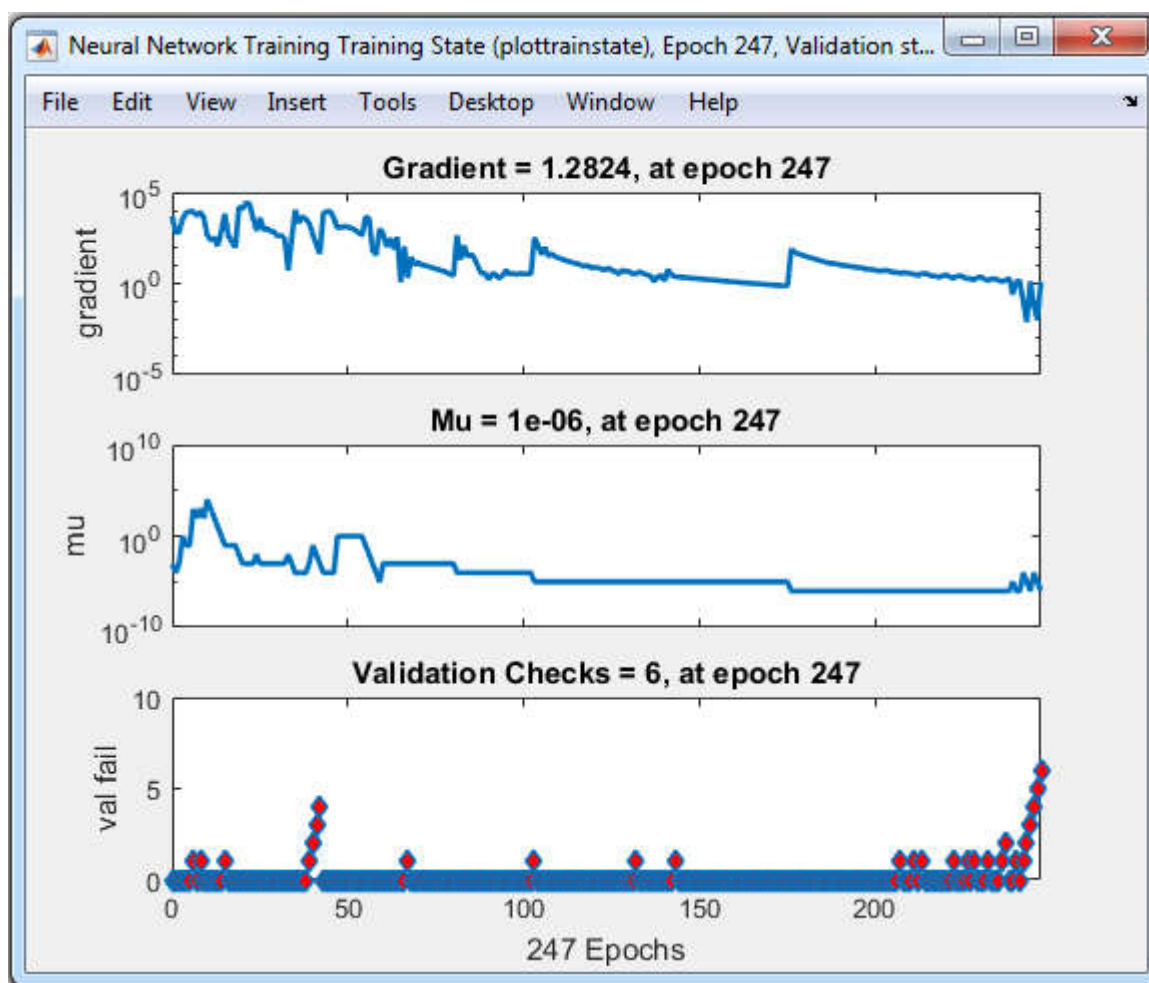


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні Neural Network Training State відображаються наступні графіки.

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

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

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

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

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

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

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

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

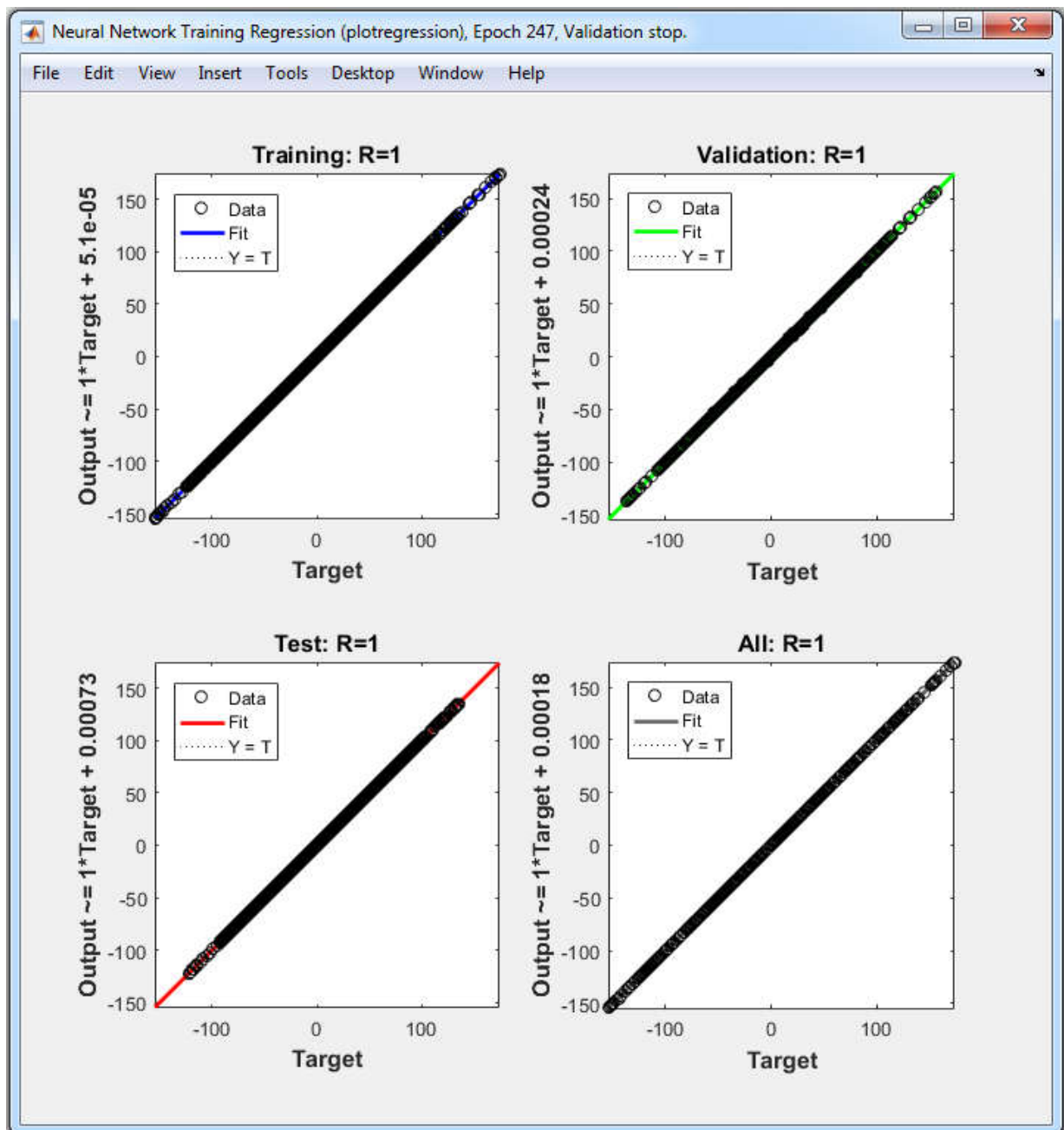


Рисунок 4.17 - Вікно Neural Network Training Regression

Опис вікна Neural Network Training Regression

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

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

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

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

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

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

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

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

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

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

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

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

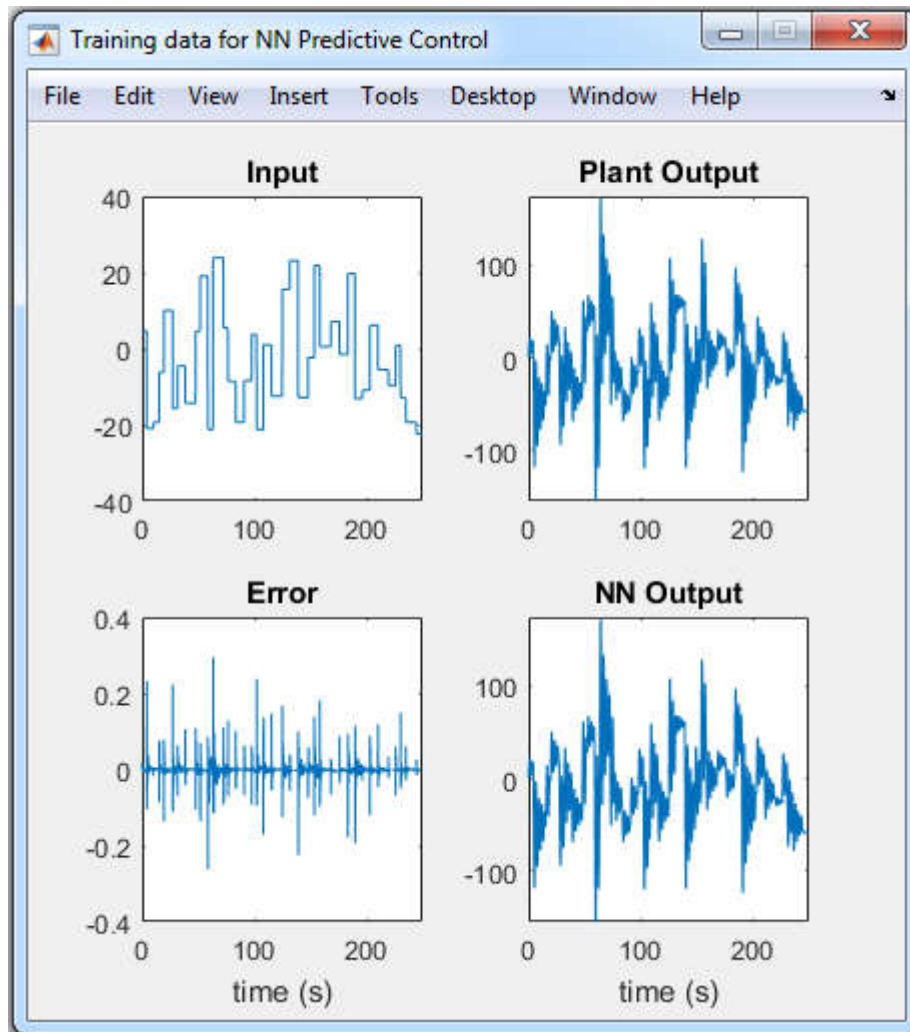


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

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

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

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

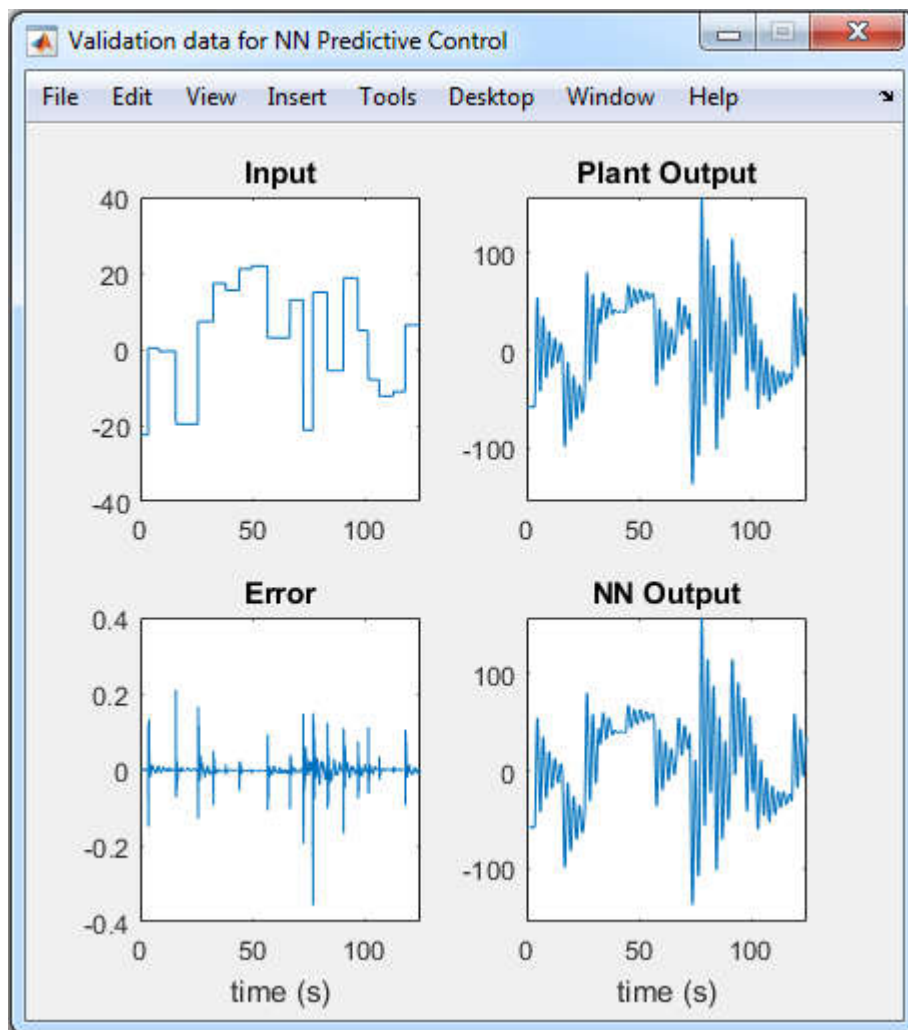


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

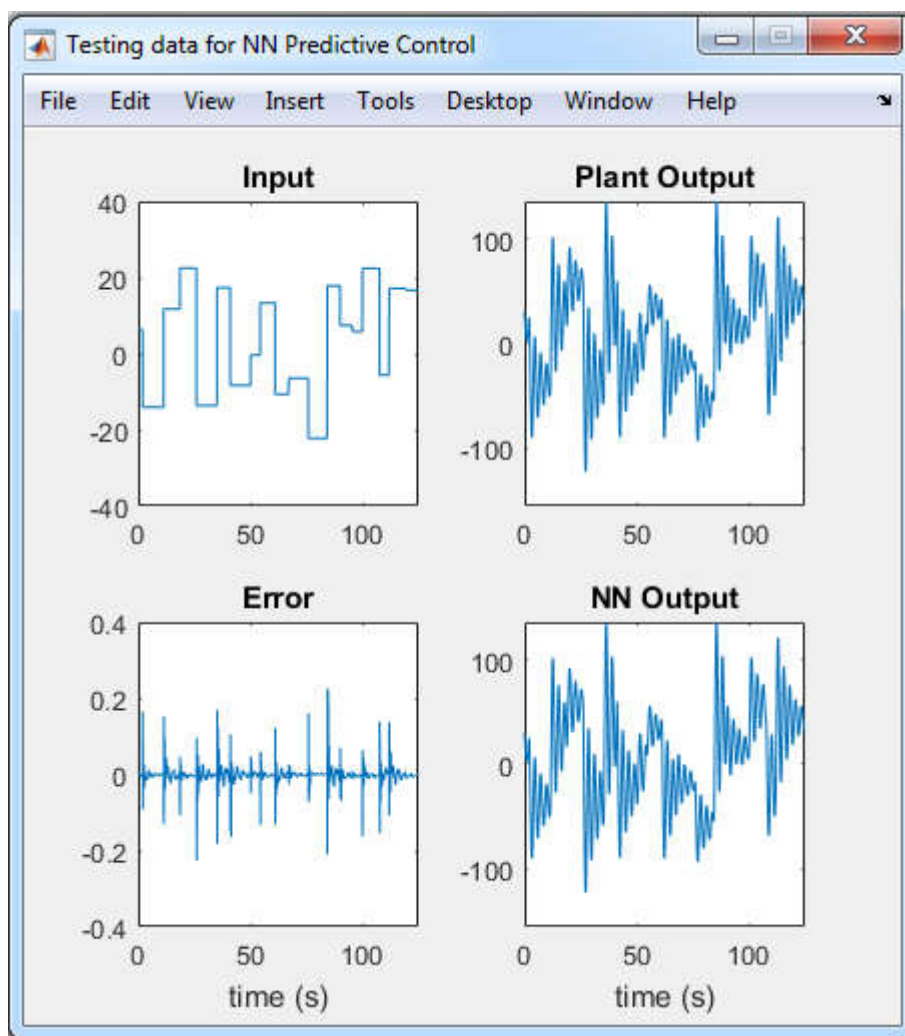


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

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.

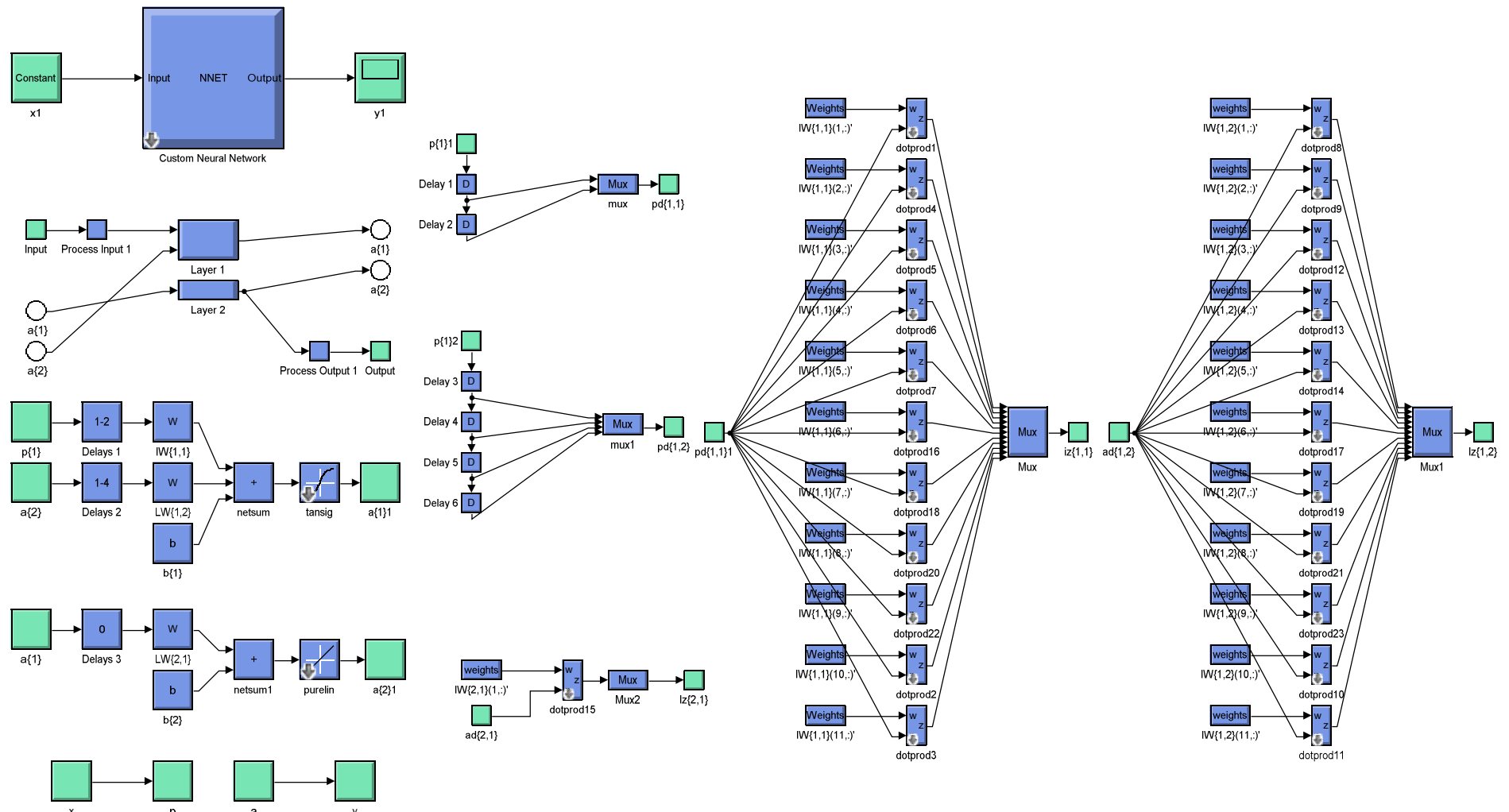


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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.

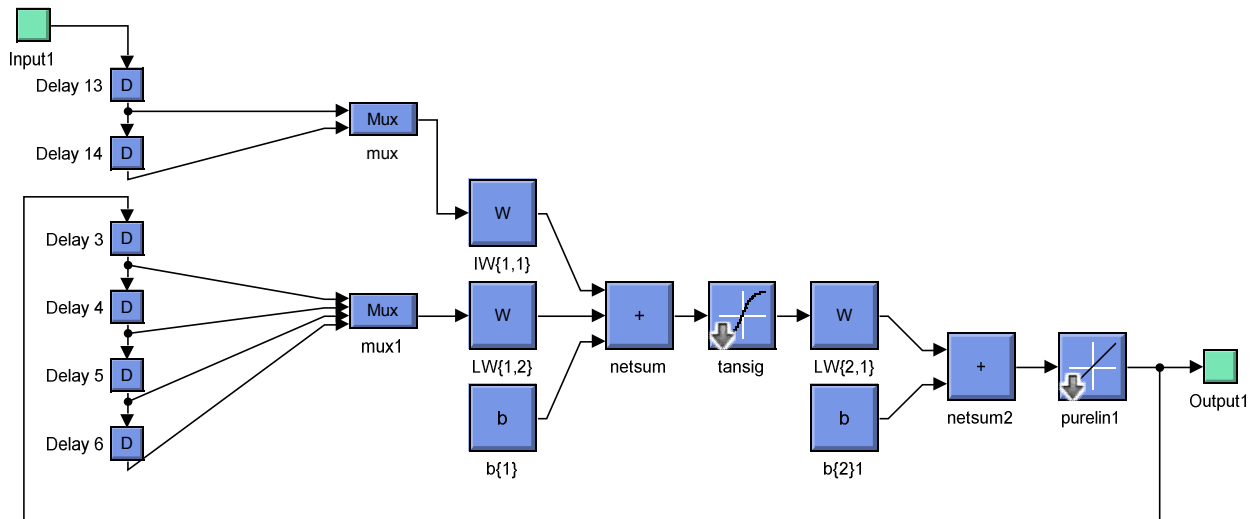


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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

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

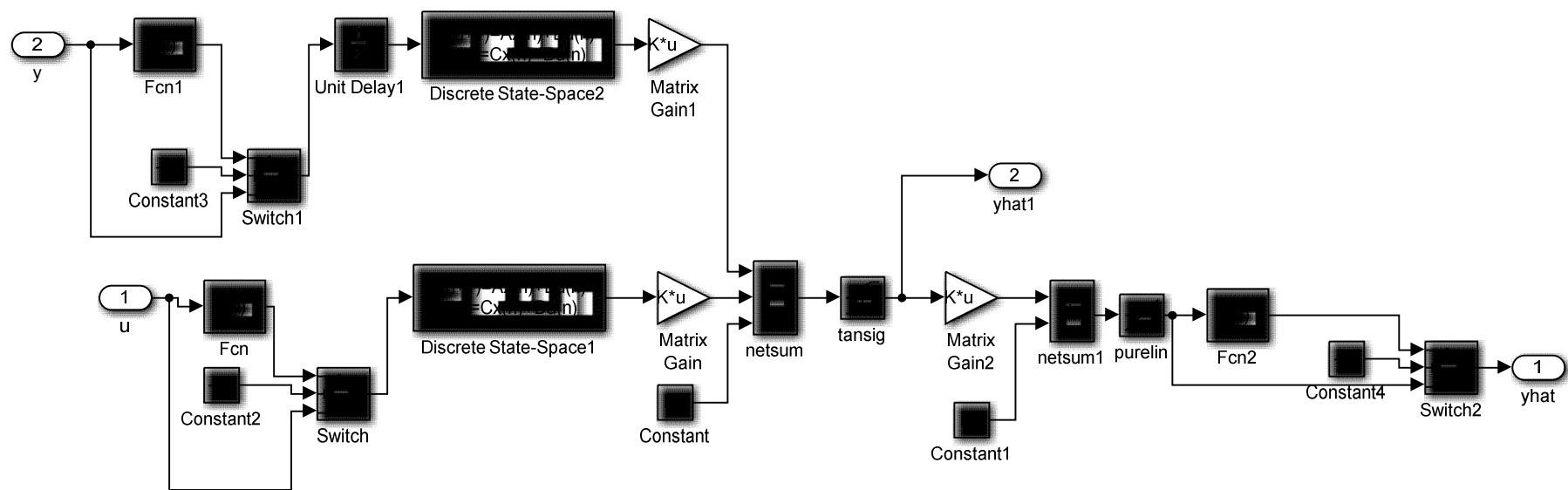


Рисунок 4.23 – Структурна схема нейронмережевої моделі об'єкту регулювання

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

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

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

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

Discrete State Space 1

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

Discrete State Space 2

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

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

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

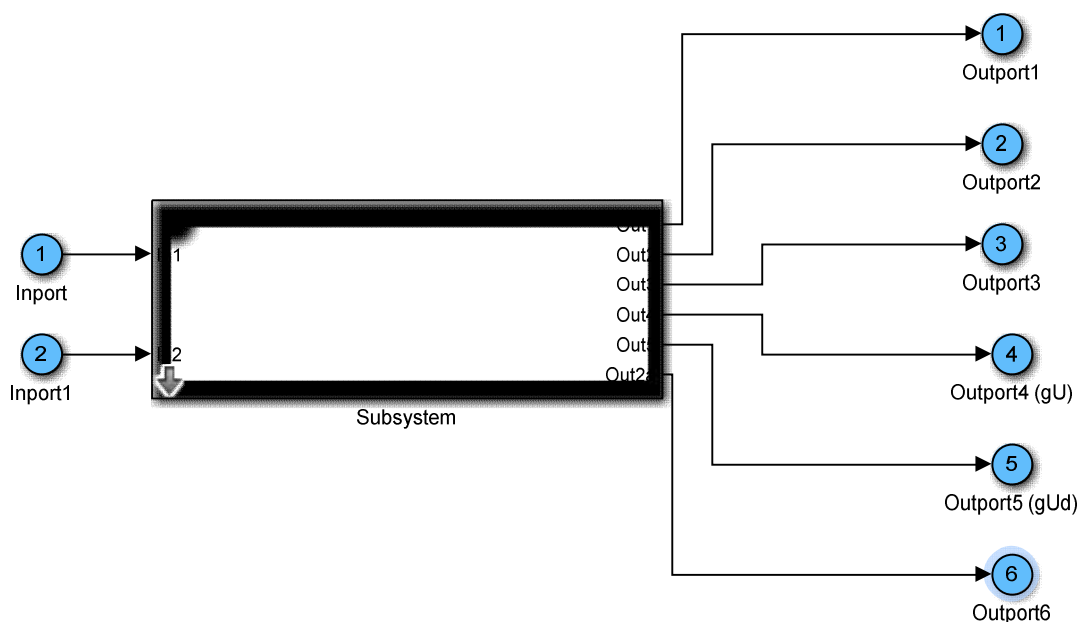


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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

Продовження опису вікна Neural Network Predictive Controller

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

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

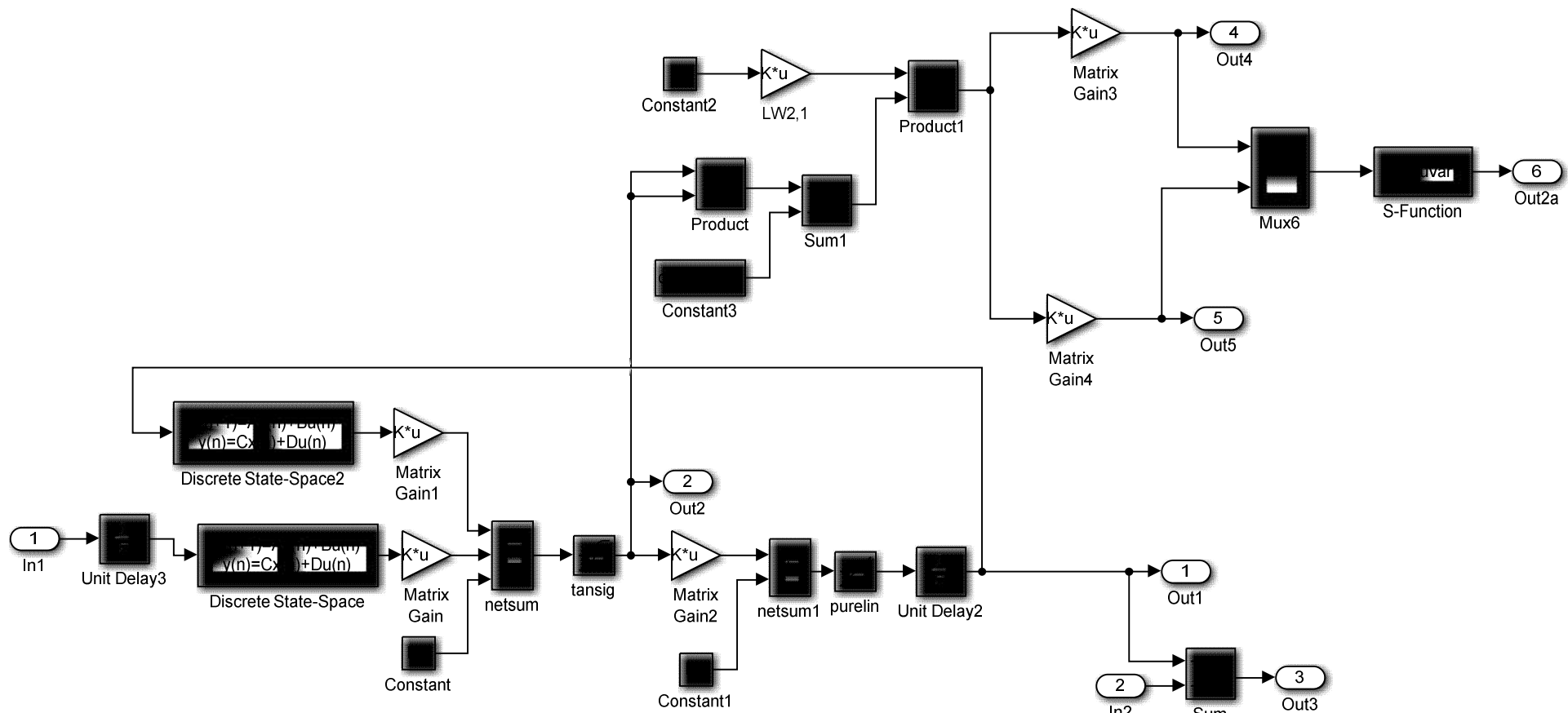


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

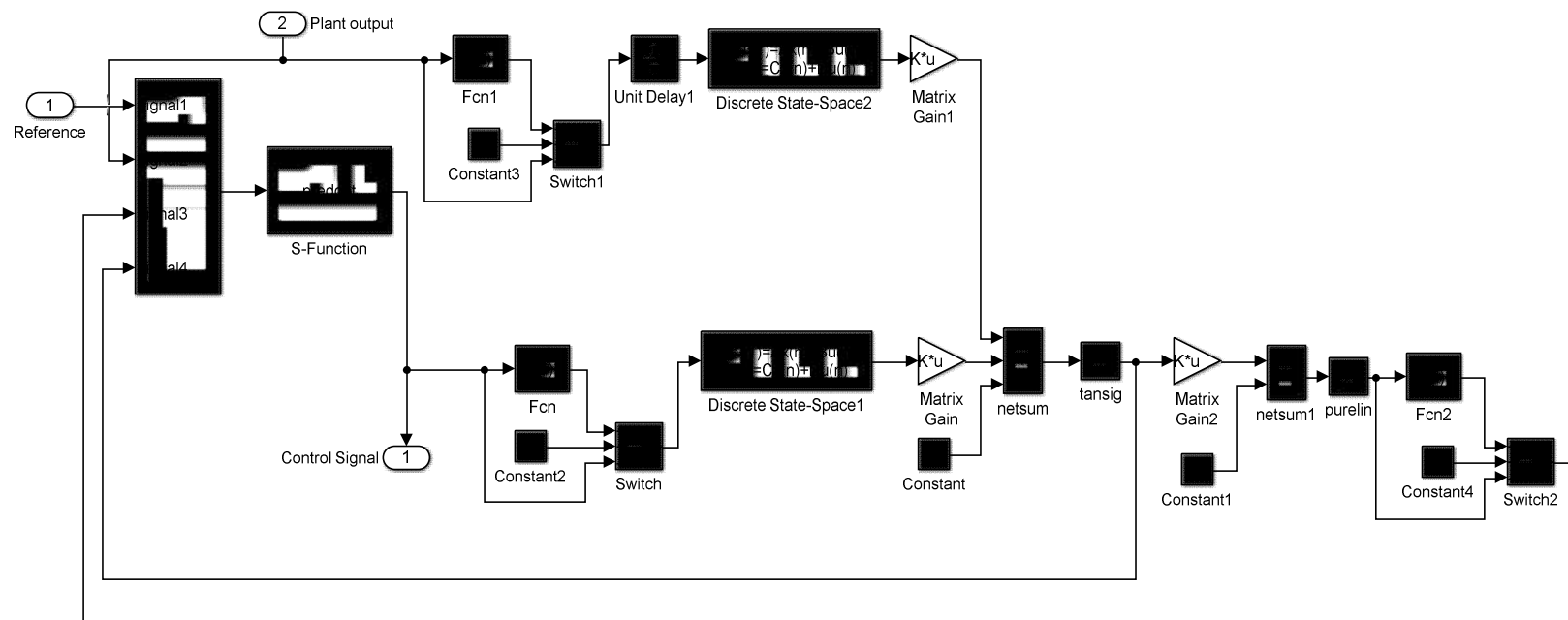


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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

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

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

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

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

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

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

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

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

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

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

Minimization Routine. Вибір процедури одновимірного пошуку.

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

Iterations Per Sample Time. Параметр означає кількість ітерацій оптимізації, які виконуються протягом кожного відрізка часу (Sample Time) при розрахунку оптимального керування системою. Цей параметр дозволяє встановити, скільки ітерацій оптимізації виконується в межах кожного відрізка часу, і впливає на швидкість та точність оптимізації контролера. Зазвичай, більша кі-

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

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

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

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

4.4 Вибір параметрів нейрорегулятора NN PREDICTIVE CONTROLLER

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

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

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

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

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

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

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

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

Як навчальну функцію використано **trainlm**.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller

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

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

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

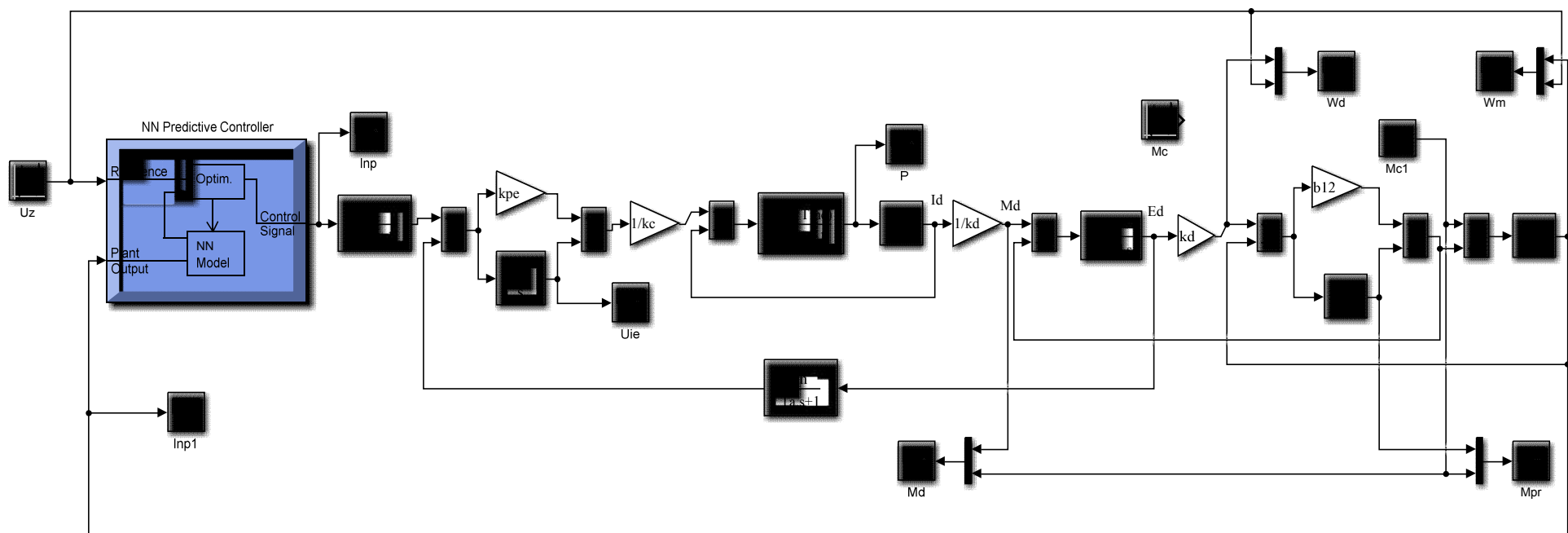
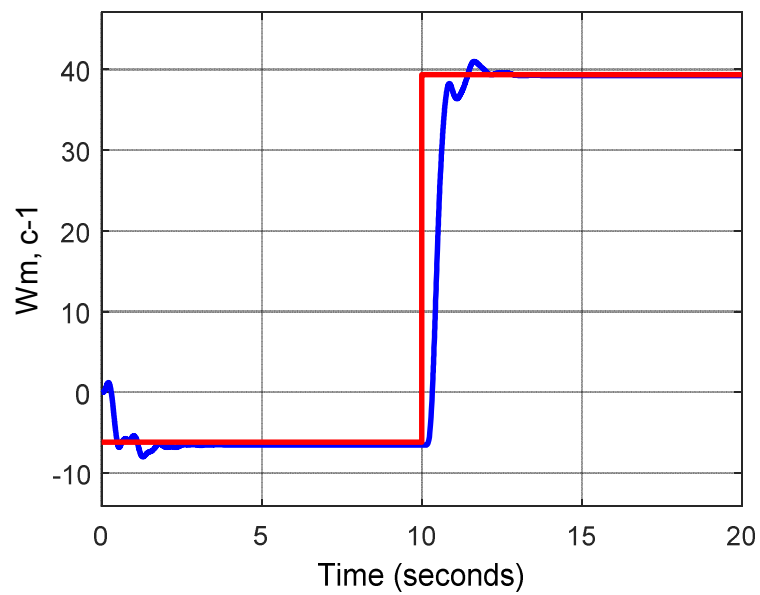
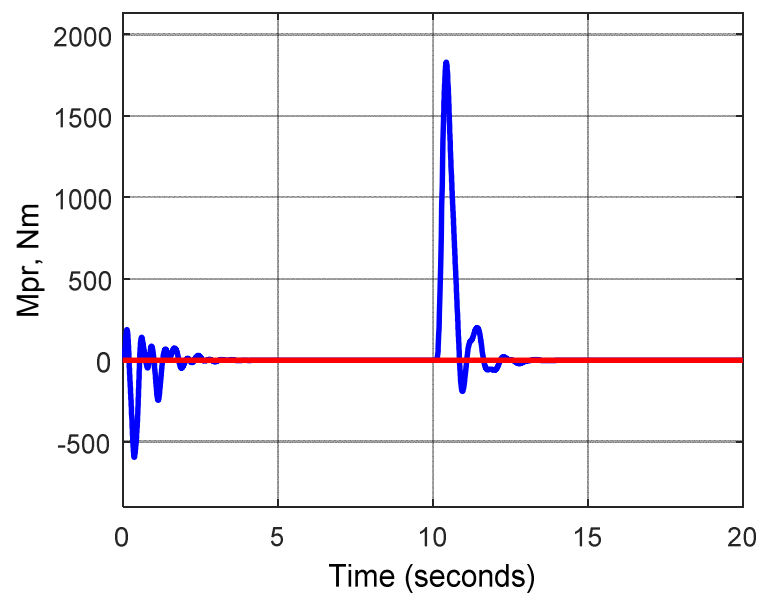


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

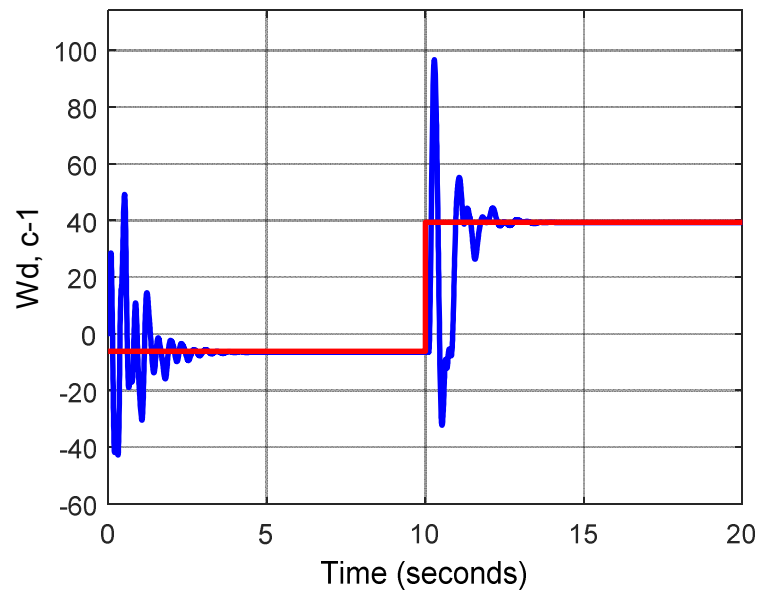


a)

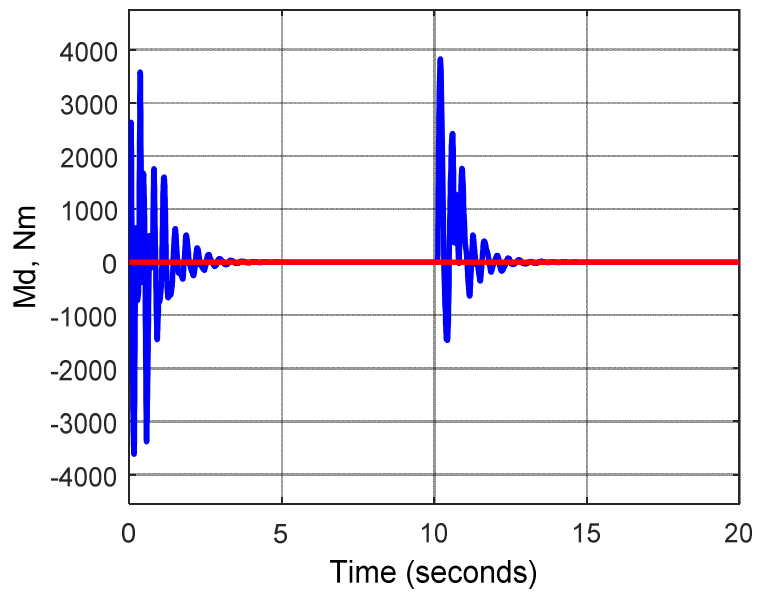


б)

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

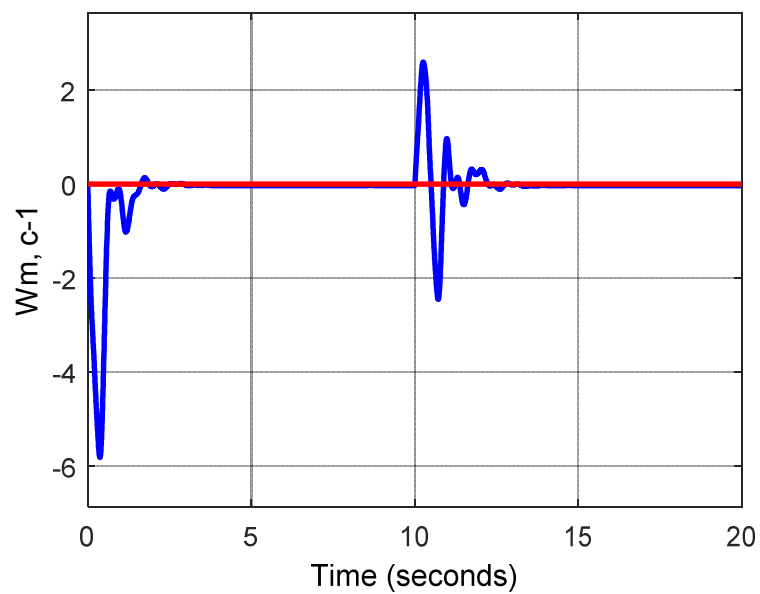


a)

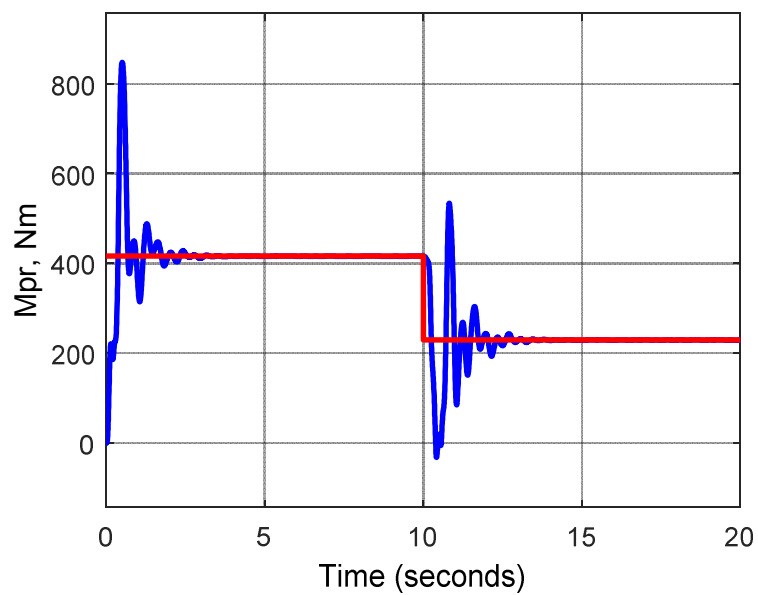


б)

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



a)



б)

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

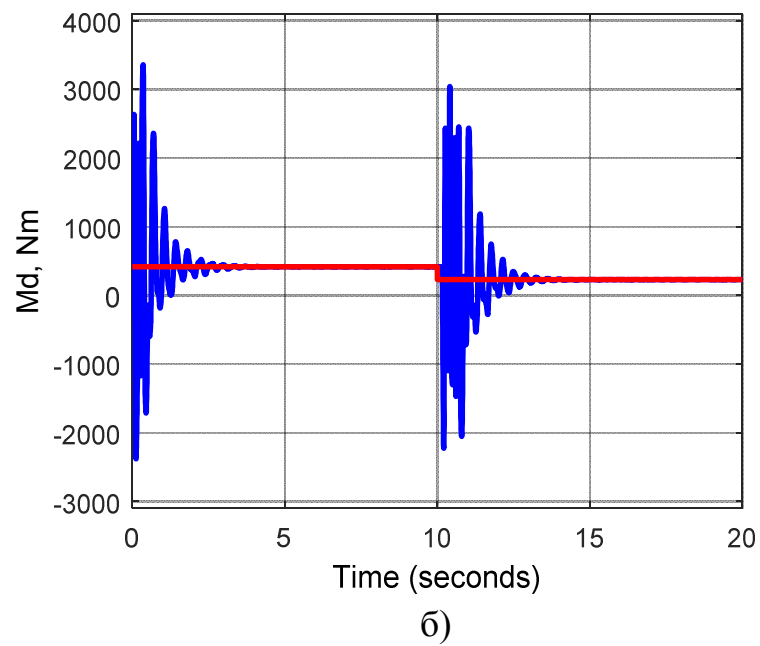
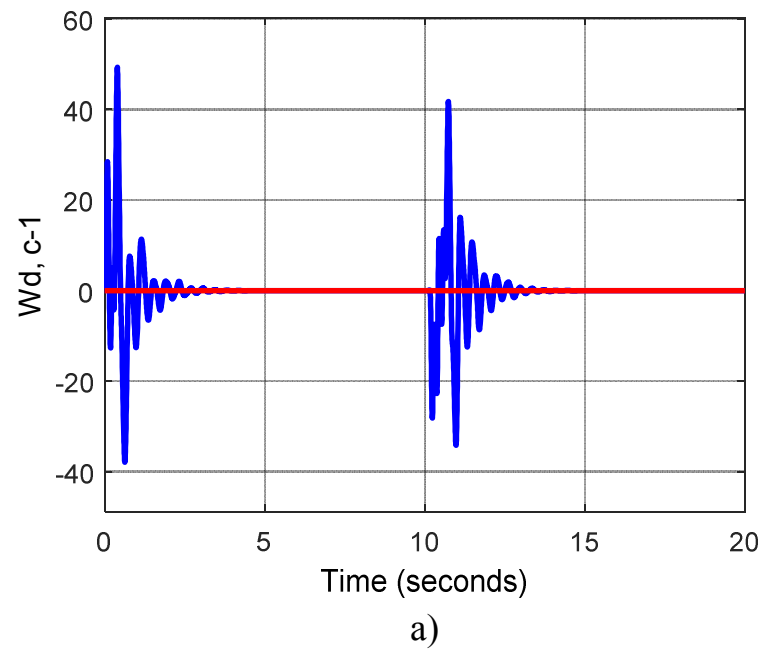


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проведено аналіз методів створення нейрорегуляторів, що реалізовані у пакеті Neural Network Toolbox системи MATLAB, включаючи регулятор з прогнозом (NN Predictive Controller), регулятор, побудований на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) і регулятор, що базується на еталонній моделі (Model Reference Controller). У даній роботі для системи стабілізації було вибрано використання регулятора з прогнозом NN Predictive Controller. Також розглядалися структура та послідовність синтезу цих нейрорегуляторів.

2. У режимі SIMULINK системи MATLAB розроблена системи управління з нейрорегулятора NN Predictive Controller, яка включає блок керованого об'єкта, в якості якого використовується схема моделі двомасової системи управління рольгангом перед ножицями стана гарячої прокатки і блок регулятора NN Predictive Controller.

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

4. Створено модель двомасової системи у вигляді двошарової нейронної мережі з прямою передачею сигналу і виконано навчання мережі з використанням алгоритму Левенберга-Марквардта. Визначено оптимальну кількість нейронів прихованого шару. Помилка навчання мережі має порядок 10^{-4} , миттєві помилки на навчальній, контрольній і тестовій множині не перевищують 0,04 .

5. Виконано синтез нейрорегулятора з використанням предиктивного контролера на основі нейронних мереж (NN Predictive Controller). Завдяки варіації параметрів нейрорегулятора в широкому діапазоні вдалося визначити парамет-

ри, які суттєво впливають на якість регулювання, та знайти оптимальні значення цих параметрів. Використання нейромережевої моделі об'єкта управління з високою точністю ідентифікації дозволило створити нейрорегулятор, що забезпечує відмінні показники якості системи.

6. Проведено моделювання нейромережевої системи з синтезованим нейрорегулятором. Аналіз результатів моделювання показав, що перехідні процеси змінних станів у системі з нейрорегулятором NN Predictive Controller під час запуску та при навантаженні мають задовільний характер. Система є астатичною.. Перерегулювання не перевищує 7%, час регулювання складає 3 с, що відповідає завданню.

ВИСНОВКИ

В магістерській кваліфікаційній роботі розв'язана актуальна науково-практична задача розробки нейромережевої системи управління багатомасовою електромеханічною системою механізму рольганга за ножицями стана гарячої прокатки, що забезпечує високу якість регулювання.

Основні результати роботи полягають у наступному.

1. В результаті аналізу систем управління механізмом рольганга перед ножицями стана гарячої прокатки, 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 рольганга стана гарячої прокатки.

2. На підставі аналізу технічних характеристик допоміжних механізмів прокатних станів і умов експлуатації електроприводів рольгангів вибраний електропривод постійного струму, виконаний за системою ТП-Д. Виконана перевірка двигуна за умовами нагріву і перевантажувальної здатності при роботі електроприводу в різних режимах, передбачених технологічним процесом.

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

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

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

5. Проведено синтез нейрорегулятора з передбаченням NN Predictive Controller. Шляхом варіювання параметрів нейрорегулятора в широкому діапазоні було визначено, які з них значуще впливають на якість регулювання, і встановлені оптимальні значення цих параметрів. Використання нейромережевої моделі об'єкта управління, що забезпечує високу ідентифікацію, дозволило синтезувати нейрорегулятор, що забезпечує високу якість функціонування системи.

6. Проведено моделювання нейромережевої системи з синтезованим нейрорегулятором. Після аналізу результатів моделювання виявлено, що перехідні процеси зміни стану системи з нейрорегулятором NN Predictive Controller під час пуску і навантаження відповідають поставленим завданням: перерегулювання швидкості механізму не перевищує 7%, а час регулювання складає 3 секунди.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

13. McCulloch W. S., Pitts W. A logical calculus of ideas imminent in nervous activity // Bulletin Mathematical Biophysics. – 1943. – № 5. – P.115-133 (Имеется перевод: Дж. Маккаллох, У. Питтс. Логическое исчисление идей, относящихся к нервной деятельности. В кн.: Автоматы. – М.: ИЛ, 1956).
14. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. – 1958. – № 65. – P.386-407 (Имеется перевод: Розенблатт Ф. Принципы нейродинамики. – М.: Мир, 1966. – 480 с.).
15. Рябинин А. Д., Шквар А. М. Некоторые принципы функционального построения инвариантных бионических систем управления // Тр. 4-го Всесоюз. совещания. "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26-30 апреля 1971. – Киев, 1971. – С 54-64.
16. Шквар А. М. Функциональные и структурные аспекты построения некоторых нейронных структур управления // Тр. 4-го Всесоюз. совещания "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26-30 апреля 1971. – Киев, 1971. – С. 78-89.
17. Hecht-Nielsen R. Kolmogorov's Mapping Neural Network Existence Theorem / Proc. of IEEE First Ann. Int. Conf. on Neural Networks. – San Diego, 1987. – 3. – P. 11-13.
18. Hecht-Nielsen R. Theory of the Backpropagation Neural Network / Proc. of Int. Joint. Conf. on Neural Networks. – Washington, D.C., 1989. – 1. – P. 593-606.
19. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных суперпозициями функций меньшего числа переменных // Докл. АН СССР. – 1956. – 108. – № 2. – С. 179-182
20. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного и сложения // Докл. АН СССР, 1957. – 114. – С. 953-956..
21. Kolmogorov, A.N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition // American Math. Soc. Transl. – 1963. – №28. – P. 55-63.

22. Cybenko G. Approximation by Superposition of a Sigmoidal Function // Math. Contr. Sign. Syst. – 1989. – 2. – P. 303-314.
23. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // Neural Networks. – 1989. – 2. – P. 359-366
24. Hornik K. Approximation Capabilities of Multilayer Feedforward Networks / Neural Networks. – 1991. – 4. – P. 251-257.
25. Hopfield J.J., Tank D. W. Neural Computation of Decisions in Optimization Problems // Biol. Cybernetics. – 1985. – 52. – P. 141-152.
26. Narendra Narendra K. Gallman P.O. An iterative method for the identification of nonlinear system using a Hammerstein model // IEEE Trans. Aut. Control. - 1966. - Vol. 11. -N3. - P.546-550.
27. Psaltis, D., Sideris, A., Yamamura, A. A Multilayered neural network controller // IEEE Control Systems Magazine. - 1988. - Vol.8. - N4. -P. 17-21.
28. Joardan M.A. Generic constraints on underspecified target trajectories // Proc. Of Int. Joint Conf. On Neural Networks (IJCNN), Washington. - 1989.- Vol.1. -N3. - P.217-225.
29. Joardan M.A., Rumelhart D.E. Forward models: Supervised learning with a distal teacher // Cognitive Science.- 1990. -Vol.16. -N5.-P.313-355.
30. Saerens M, Soquet A. Neural controller based on backpropagation algorithm // IEEE Proceedings-F. - 1991. -Vol.138. -N.I. -P.55-62.
31. Miller III, W.T., Glanz, F.H., Kraft III, L.G. Application of a general learning algorithm to control of robotic manipulators // Int. J. of Robotics Research. - 1987. - Vol.6. - N2. - P.84-98.
32. Kawato, M., Furukawa, K, Suzuki, R. A hierarchical neural network model for control and learning of voluntary movement//Biological Cybernetics. - 1987. - Vol.57. - N2. -P. 169-185.
33. Pham D.T., Liu X. Neural Network for Identification, Prediction and Control. - London: Springer - Verlag, 1997. - 238p.
34. Milito R., Padilla C.S., Padilla R.A., Cadorin D. An innovations approach to dual control // IEEE Trans. Automat. Control. - 1982. -Vol.27. -N1. -P.133-137.

35. Cichocki A., Unbehauen R. Neural networks for optimization and signal processing. – New York: John Wiley & Sons, 1998. – 52lp.
36. Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // Automatica. – 1989. – Vol.25. – P.335-348.
37. Narendra K.S., Parthasarathy K. Identification and control of dynamical system using neural networks // IEEE Trans. Neural Networks.–1990. – Vol.1.–№1. – P.4-27.
38. Leontaritis I.J., Billings S.A. Input-output parametric model for nonlinear systems. Part 1: Deterministic non-linear systems. Part 2: Stochastic non-linear systems // Int. J. Control. – 1985. – Vol.41. – №2. –P. 303-344.
39. Bunke J. Kunstliche neuronale Netze zur Systemidentifikation aus gestorten Mefiwerten. Forsch. Ber. VDI, Reihe 8, № 667. – Dusseldorf: VDI Verlag. – 1997. – P. 37-51.
40. Бычков В. П. Электропривод и автоматизация металлургического производства. Учеб. Пособие для вузов. Изд.2-е. перераб. и доп. – М.: Высшая школа, 1977. – 391с.
41. Тиристорные электроприводы прокатных станов. Перельмутер В. М. и др. – М., Металлургия, 1978. – 152 с.
42. Бешта О.С., Балахонцев О.В., Бородай В.А. Автоматизований електропривод у прокатному виробництві. – Національний гірничий університет, Дніпропетровськ, 2010. – 224 с.
43. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
44. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
45. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
46. 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.

47. 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.
48. 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.
49. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.
50. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.
51. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
52. 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
53. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г.Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч1.
54. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г. Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч2.
55. 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.
56. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрьоменко. Харків, ХНАДУ, 2012. – 284 с.
57. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
58. 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.
59. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 с.
60. 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.

ДОДАТОК А

СТРУКТУРА ШТУЧНОГО НЕЙРОНА

Нервова система і мозок людини складаються з нейронів, сполучених між собою нервовими волокнами. Нервові волокна здатні передавати електричні імпульси між нейронами. Всі процеси передачі роздратувань від шкіри, вух і очей до мозку, процеси мислення і управління діями - все це реалізовано в живому організмі як передача електричних імпульсів між нейронами.

Нейрон (нервова клітина) є особливою біологічною кліткою, яка обробляє інформацію (рис. А.1). Він складається з тіла і відростків нервових волокон двох типів - дендритов, по яким приймаються імпульси, і єдиного аксона, по якому нейрон може передавати імпульс. Тіло нейрона включає ядро, яке містить інформацію про спадкові властивості, і плазму, що володіє молекулярними засобами для виробництва необхідних нейрону матеріалів. Нейрон отримує сигнали (імпульси) від аксонів інших нейронів через дендрити (приймачі) і передає сигнали, які згенеровані тілом клітки, уздовж свого аксона (передавач), який в кінці розгалужується на волокна. На закінченнях цих волокон знаходяться спеціальні утворення - синапси, які впливають на силу імпульсу.

Синапс є елементарною структурою і функціональним вузлом між двома нейронами (волокно аксона одного нейрона і дендрит іншого). Коли імпульс досягає синаптичного закінчення, вивільняються певні хімічні речовини, звані нейротрансмітерами. Нейротрансмітери дифундують через синаптичну щілину, збуджуючи або загальмовуючи, в залежності від типу синапсу, здатність нейрона-приймача генерувати електричні імпульси. Результативність синапсу може налаштовуватися сигналами, що проходять через нього, так що синапси можуть навчатися залежно від активності процесів, в яких вони беруть участь. Ця залежність від передісторії діє як пам'ять, яка, можливо, відповідальна за пам'ять людини. Важливо відзначити, що ваги синапсів можуть змінюватися з часом, що змінює і поведінку відповідного нейрона.

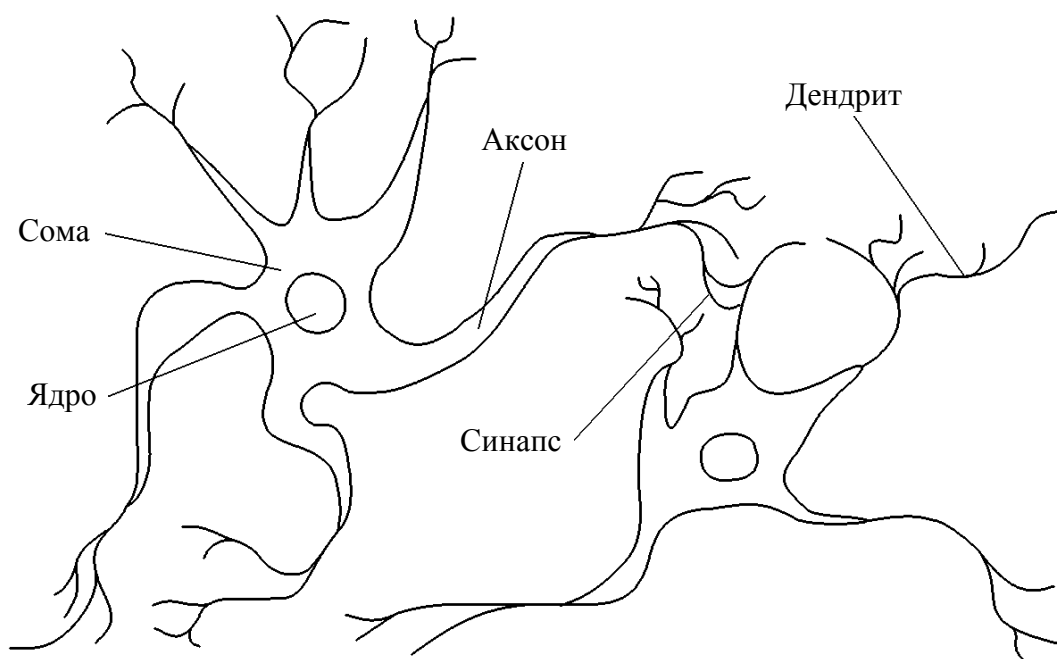


Рисунок А.1 – Взаємозв'язок біологічних нейронів

Структура штучного нейрона показана на рис. А.2.

До складу нейрона входять помножувачі (синапси), суматор і нелінійний перетворювач. Синапси здійснюють зв'язок між нейронами і умножають вхідний сигнал на число, що характеризує силу зв'язку, – вага синапсу. Суматор виконує складання сигналів, що поступають по синаптичним зв'язкам від інших нейронів, і зовнішніх вхідних сигналів. Нелінійний перетворювач реалізує нелінійну функцію одного аргументу – виходу суматора. Ця функція називається функцією активації або передавальною функцією нейрона. Нейрон в цілому реалізує скалярну функцію векторного аргументу. Математична модель нейрона описується співвідношеннями:

$$s = \sum_{i=1}^n w_i \cdot x_i + b, \quad y = f(s)$$

де w_i – вага синапсу ($i = 1 \dots n$);

b – значення зсуву;

s – результат підсумовування;

x_i – компонент вхідного вектора (вхідний сигнал) ($i = 1 \dots n$);

y – вихідний сигнал нейрона;
 n – число входів нейрона;
 f – нелінійне перетворення (функція активації або передавальна функція).

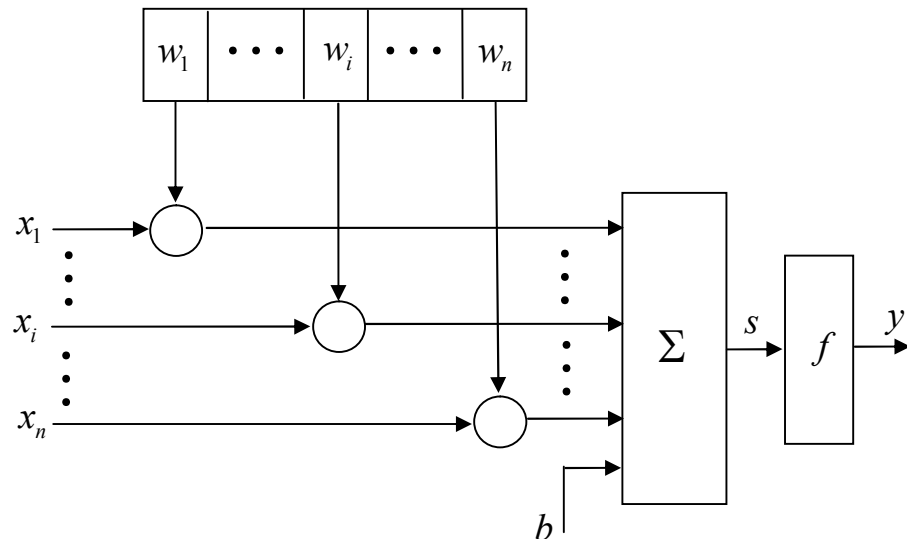


Рисунок А.2 – Структура штучного нейрона

У загальному випадку вхідний сигнал, вагові коефіцієнти і значення зсуву можуть приймати речовинні значення. Вихід (y) визначається видом функції активації і може бути як дійсним, так і цілим. У багатьох практичних завданнях входи, ваги і зсуви можуть приймати лише деякі фіксовані значення.

Синаптичні зв'язки з позитивними вагами називають збудливими, з негативними вагами – гальмуючими.

Таким чином, нейрон повністю описується своїми вагами w_i і передавальною функцією $f(s)$. Отримавши набір чисел (вектор) x_i як входи, нейрон видає деяке число y на виході.

Описаний обчислювальний елемент можна вважати спрощеною математичною моделлю біологічних нейронів – кліток, з яких складається нервова система людини і тварин.

Щоб підкреслити відмінність нейронів біологічних і штучних (математичних), другі іноді називають нейроноподібними елементами або формальними (штучними) нейронами.

На вхідний сигнал s нелінійний перетворювач відповідає вихідним сигналом $f(s)$, який є виходом нейрона y . Приклади активаційних функцій представлені на рис. А.3 і в табл. А.1.

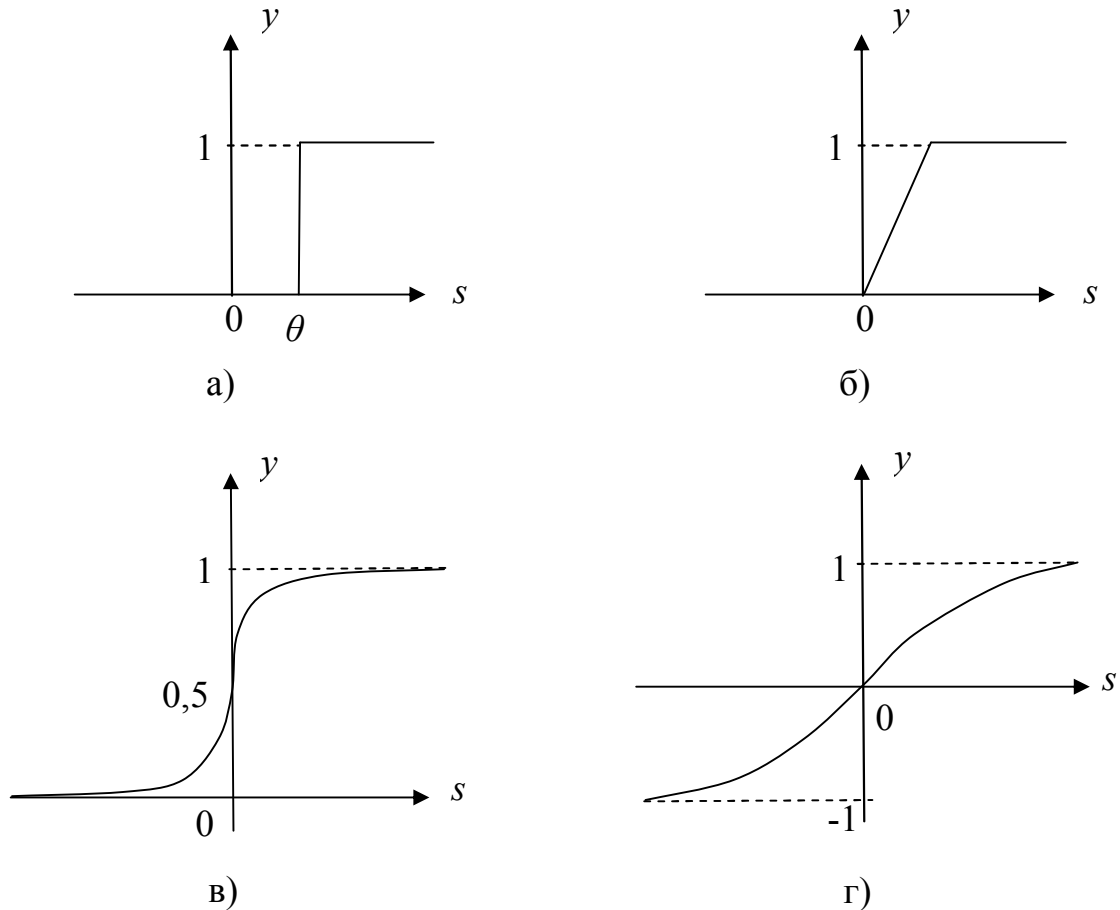


Рисунок А.3 – Приклади активаційних функцій

а – порогова; б – напівлінійна з насиченням;

в – сигмоїд (логістична функція);

г – сигмоїд (гіперболічний тангенс)

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

$$f(s) = \frac{1}{1 + e^{-s}}.$$

З виразу для сигмоїда очевидно, що вихідне значення нейрона лежить в діапазоні $[0, 1]$. Одна з цінних властивостей сигмоїдальної функції – простий вираз для її похідної:

$$f'(s) = f(s) \cdot [1 - f(s)].$$

Крім того, дана функція володіє властивістю «підсилювати» слабкі сигнали краще, ніж великі, і запобігає насиченню від великих сигналів, оскільки вони відповідають областям аргументів, де сигмоїд має пологий нахил.

Таблиця А.1 - Перелік функцій активації нейронів

Назва	Формула	Область значень
Порогова	$f(s) = \begin{cases} 0, & s < \theta \\ 1, & s \geq \theta \end{cases}$	$(0, 1)$
Знакова (сигнатурна)	$f(s) = \begin{cases} 1, & s > 0 \\ -1, & s \leq 0 \end{cases}$	$(-1, 1)$
Сигмоїдальна (логістична)	$f(s) = \frac{1}{1 + e^{-s}}$	$(0, 1)$
Напівлінійна	$f(s) = \begin{cases} s, & s > 0 \\ 0, & s \leq 0 \end{cases}$	$(0, \infty)$
Лінійна	$f(s) = s$	$(-\infty, \infty)$
Радіальна базисна (гаусова)	$f(s) = \exp(-s^2)$	$(0, 1)$
Напівлінійна з насиченням	$f(s) = \begin{cases} 0, & s \leq 0 \\ s, & 0 < s < 1 \\ 1, & s \geq 1 \end{cases}$	$(0, 1)$
Лінійна з насиченням	$f(s) = \begin{cases} -1, & s \leq -1 \\ s, & -1 < s < 1 \\ 1, & s \geq 1 \end{cases}$	$(-1, 1)$
Гіперболічний тангенс (сигмоїдальна)	$f(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$	$(-1, 1)$
Трикутна	$f(s) = \begin{cases} 1 - s , & s \leq 1 \\ 0, & s > 1 \end{cases}$	$(0, 1)$

ДОДАТОК Б

КЛАСИФІКАЦІЯ НЕЙРОННИХ МЕРЕЖ І ЇХ ВЛАСТИВОСТІ

Штучна нейронна мережа – це набір нейронів, сполучених між собою. Як правило, передавальні (активаційні) функції всіх нейронів в мережі фіксовані, а ваги є параметрами мережі і можуть змінюватися. Деякі входи нейронів помічені як зовнішні входи мережі, а деякі виходи – як зовнішні виходи мережі. Подаючи будь-які числа на входи мережі, ми отримуємо якийсь набір чисел на виходах мережі. Таким чином, робота нейронної мережі полягає в перетворенні вхідного вектора X у вихідний вектор Y , причому це перетворення задається вагами мережі.

Практично будь-яке завдання можна звести до завдання, вирішуваною нейронною мережею.

Питання про методику побудови нейронної мережі вирішується в два етапи:

- 1 – вибір типу (архітектура) мережі;
- 2 – підбір вагів (навчання) мережі.

На першому етапі слід вибрати наступне:

- 1 – які нейрони ми хочемо використовувати (число входів, передавальні функції);
- 2 – яким чином слід з'єднати їх між собою;
- 3 – що взяти за входи і виходи мережі.

Це завдання на перший погляд здається незрозумілим, але, на щастя, необов'язково придумувати нейронну мережу «з нуля» – існує декілька десятків різних нейромережових архітектур, причому ефективність багатьох з них доведена математично. Найбільш популярні і вивчені архітектури – це багатошаровий перцептрон, нейромережа із загальною регресією, мережі Кохонена та інші.

На другому етапі слід «навчити» вибрану мережу, тобто підібрати такі значення, її вагів, щоб мережа працювала потрібним чином. У використовуваних на практиці нейромережах кількість вагів може складати декілька десятків

тисяч, тому навчання – дійсно складний процес. Для багатьох архітектури розроблені спеціальні алгоритми навчання, які дозволяють набудувати ваги мережі певним чином.

Залежно від функцій, що виконуються нейронами в мережі, можна виділити три їх типу:

1 – вхідні нейрони – на які подається вхідний вектор, що кодує вхідну дію або образ зовнішнього середовища; у них зазвичай не здійснюється обчислювальних процедур, інформація передається з входу на вихід нейрона шляхом зміни його активації;

2 – вихідні нейрони – вихідні значення яких представляють вихід мережі; перетворення в таких нейронах здійснюються по приведених вище виразах;

3 – проміжні нейрони – складають основу штучних нейронних мереж, перетворення в них виконуються по цих же виразах.

У більшості нейронних моделей тип нейрона пов'язаний з його розташуванням в мережі. Якщо нейрон має тільки вихідні зв'язки, то це вхідний нейрон, якщо навпаки – вихідний нейрон. Проте може зустрітися випадок, коли вихід внутрішнього нейрона розглядається як частина виходу мережі. В процесі функціонування мережі здійснюється перетворення вхідного вектора у вихідний, тобто деяка переробка інформації. Конкретний вид виконуваного мережею перетворення інформації обумовлюється не тільки характеристиками нейроподібних елементів (функціями активації і т. п.), але і особливостями її архітектури. Зокрема, тією або іншою топологією міжнейронних зв'язків, вибором певних підмножин нейроподібних елементів для введення і виведення інформації, способами навчання мережі, наявністю або відсутністю конкуренції між нейронами, напрямом і способами передачі інформації між нейронами.

Класифікуючи нейронні мережі по топології, можна виділити три основні типи таких мереж:

- 1 – повнозв'язні мережі (рис. Б.1 а);
- 2 – багатошарові або шаруваті мережі (рис.Б.1 б);
- 3 – слабозв'язані мережі (нейронні мережі з локальними зв'язками) (рис.Б.1в).

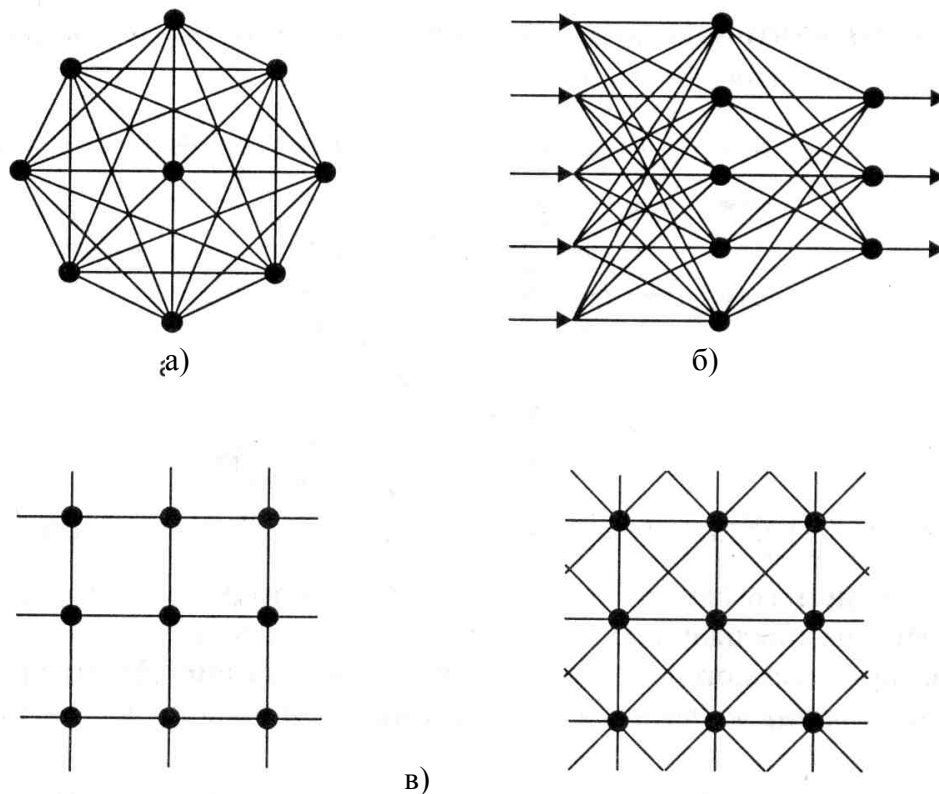


Рисунок – Б.1 Архітектура нейронних мереж

а – повнозв'язна мережа

б – багатошарова мережа з послідовними зв'язками;

в – слабозв'язні мережі

Повнозв'язна мережа – це ШНМ, кожен нейрон якої передає свій вихідний сигнал решті нейронів, у тому числі і самому собі. Всі вхідні сигнали подаються всім нейронам. Вихідними сигналами мережі можуть бути все або деякі вихідні сигнали нейронів після декількох тактів функціонування мережі.

У багатошарових мережах нейрони об'єднуються в шари. Шар містить сукупність нейронів з єдиними вхідними сигналами. Число нейронів в кожному шарі може бути будь-яким і ніяк наперед не пов'язано з кількістю нейронів в інших шарах. У загальному випадку мережа складається з Q шарів, пронумерованих зліва направо. Зовнішні вхідні сигнали подаються на входи нейронів першого шару (вхідний шар часто нумерують як нульовою), а виходами мережі є вихідні сигнали останнього шару. Вхід нейронної мережі можна розглядати як

вихід «нульового шару» вироджених нейронів, які служать лише як розподільні крапки, підсумовування і перетворення сигналів тут не проводиться. Окрім вхідного і вихідного шарів в багатошаровій нейронній мережі є один або декілька проміжних (прихованих) шарів. Зв'язки від виходів нейронів деякого шару q до входів нейронів наступного шару ($q + 1$) називаються послідовними.

У свою чергу, серед шаруватих мереж виділяють наступні типи.

Мережі без зворотних зв'язків. У таких мережах нейрони вхідного шару отримують вхідні сигнали, перетворюють їх і передають нейронам 1-го прихованого шару, далі спрацьовує 1-й прихований шар і т.д. до 0-го шару, який видає вихідні сигнали для інтерпретатора і користувача. Якщо не обумовлене осоружне, то кожен вихідний сигнал q -го шару подається на вхід всіх нейронів ($q + 1$)-го шару; проте можливий варіант з'єднання q -го шару з довільним ($q + p$)-м шаром.

Слід зазначити, що класичним варіантом шаруватих мереж є мережі прямого розповсюдження (рис. Б.2).

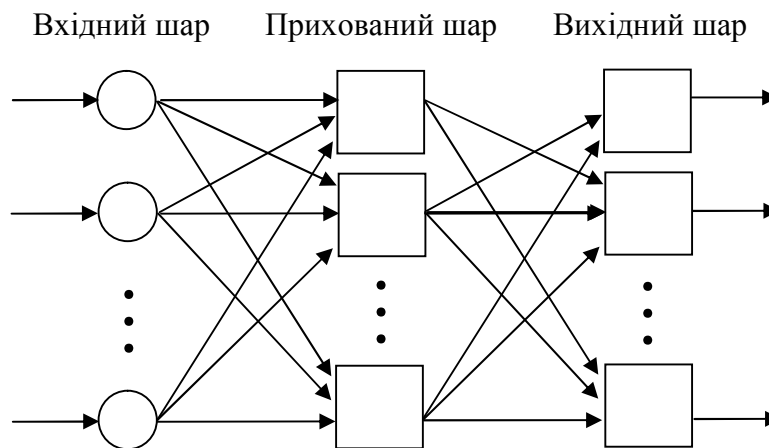


Рисунок Б.2 – Багатошарова (двошарова) мережа прямого розповсюдження

Мережі із зворотними зв'язками. Це мережі, у яких інформація з подальших шарів передається на попередні.

Як приклад мереж із зворотними зв'язками на рис. Б.3 представлені так звані частково-рекурентні мережі Елмана і Жордана.

Мережі можна класифікувати також по числу шарів.

Теоретично число шарів і число нейронів в кожному шарі може бути довільним, проте фактично воно обмежене ресурсами комп'ютера або спеціалізованої мікросхеми, на яких зазвичай реалізуються ШНМ. Чим складніше ШНМ, тим масштабніші завдання, підвладні їй.

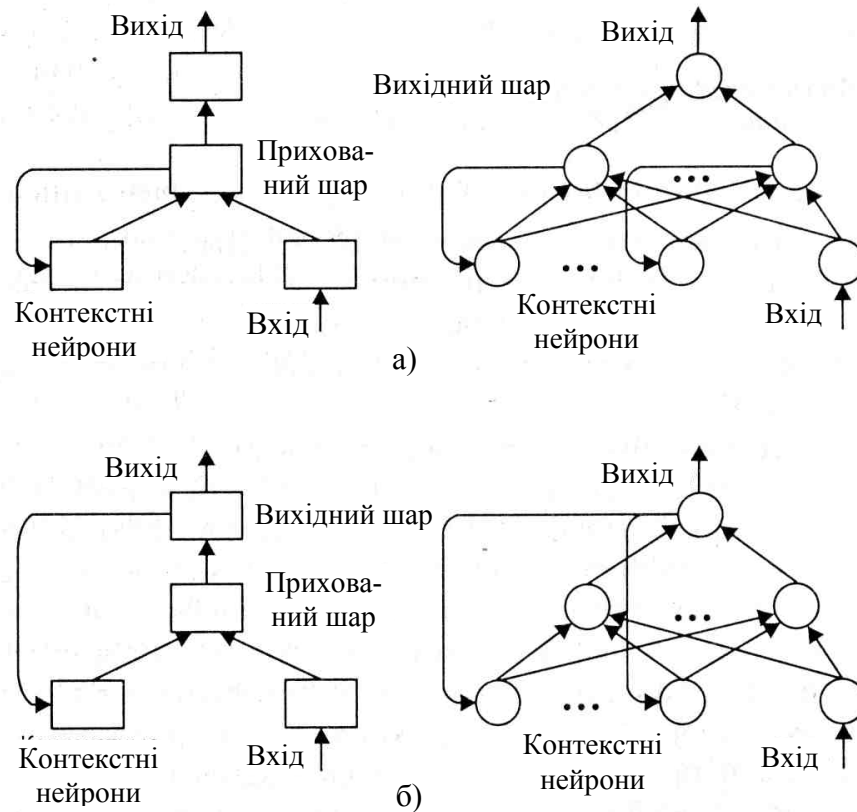


Рисунок Б.3 – Частково-рекурентні мережі
а – мережа Елмана, б – мережа Жордана

Вибір структури ШНМ здійснюється відповідно до особливостей і складності завдання. Для вирішення деяких окремих типів завдань вже існують оптимальні (на сьогоднішній день) конфігурації. 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:

1 – можливості мережі зростають із збільшенням числа осередків мережі, щільність зв'язків між ними і числом виділених шарів;

2 – введення зворотних зв'язків разом із збільшенням можливостей мережі піднімає питання про так звану динамічну стійкість мережі;

3 – складність алгоритмів функціонування мережі (зокрема, наприклад, введення декількох типів синапсів – збудливих, гальмуючих і ін.) також сприяє посиленню можливостей 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. В більшості випадків оптимальний варіант виходить на основі інтуїтивного підбору, хоча в літературі приведені докази того, що для будь-якого алгоритму існує нейронна мережа, яка може його реалізувати. Зупинимось на цьому докладніше.

Багато завдань: розпізнавання образів (зорових, мовних), виконання функціональних перетворень при обробці сигналів, управління, прогнозування, ідентифікації складних систем, зводяться до наступної математичної постановки. Необхідно побудувати відображення $X \rightarrow Y$ таке, щоб у відповідь на кожен можливий вхідний сигнал X формувався правильний вихідний сигнал Y . Відображення задається кінцевим набором пар ($\langle \text{вхід} \rangle$, $\langle \text{відомий вихід} \rangle$). Число таких пар (навчальних прикладів) істотно менше загального числа можливих поєднань значень вхідних і вихідних сигналів. Сукупність всіх навчальних прикладів носить назву навчальної вибірки.

У завданнях розпізнавання образів X – деяке представлення образу (зображення, вектор чисел), Y — номер класу, до якого належить вхідний образ.

У завданнях управління X – набір контрольованих параметрів керованого об'єкту, Y – код, що визначає дію, що управляє, відповідну поточним значенням контрольованих параметрів.

У завданнях прогнозування як вхідні сигнали використовуються тимчасові ряди, що представляють значення контрольованих змінних на деякому інтервалі часу. Вихідний сигнал – безліч змінних, яка є підмножиною змінних вхідного сигналу.

При ідентифікації $\mathbf{X}$ і $\mathbf{Y}$ представляють вхідні і вихідні сигнали системи відповідно.

Взагалі кажучи, велика частина прикладних завдань може бути зведена до реалізації деякого складного багатовимірного функціонального перетворення $\mathbf{X} \rightarrow \mathbf{Y}$.

В результаті побудови такого перетворення (відображення) необхідно добитися того, щоб забезпечувалося формування правильних вихідних сигналів у відповідності:

- 1 – зі всіма прикладами навчальної вибірки;
- 2 – зі всіма можливими вхідними сигналами, які не увійшли до навчальної вибірки.

Друга вимога в значній мірі ускладнює завдання формування навчальної вибірки. У загальному вигляді ця задача в даний час ще не вирішена, проте у всіх відомих випадках було знайдено приватне рішення. Подальші міркування припускають, що навчальна вибірка вже сформована.

Відзначимо, що теоретичною основою для побудови нейронних мереж є наступне твердження: для будь-якої безлічі пар вхідних – вихідних векторів довільної розмірності $\{(\mathbf{X}^k \rightarrow \mathbf{Y}^k), k=1...N\}$ існує двошарова однорідна нейронна мережа з послідовними зв'язками, з сигмоїдальними передавальними функціями і з кінцевим числом нейронів, яка для кожного вхідного вектора $\mathbf{X}^k$ формує відповідний йому вихідний вектор $\mathbf{Y}^k$.

Таким чином, для представлення багатовимірних функцій багатьох змінних може бути використана однорідна нейронна мережа, що має всього один прихований шар, з сигмоїдальними передавальними функціями нейронів.

Для оцінки числа нейронів в прихованих шарах однорідних нейронних мереж можна скористатися евристичною формулою для оцінки необхідного числа синаптичних вагів L_w (у багатошаровій мережі з сигмоїдальними передавальними функціями):

$$\frac{mN}{1 + \log_2 N} \leq L_w \leq \left(\frac{N}{m} + 1 \right) (n + m + 1) + m ,$$

де n – розмірність вхідного сигналу

m – розмірність вихідного сигналу

N – число елементів навчальної вибірки.

Оцінивши необхідне число вагів, можна розрахувати число нейронів в прихованих шарах. Наприклад, число нейронів в двошаровій мережі складе

$$L = \frac{L_w}{n + m} .$$

Відомі і інші подібні формули, наприклад, вигляду

$$2 \cdot (L + n + m) \leq N \leq 10 \cdot (L + n + m) ;$$

$$\frac{N}{10} - n - m \leq L \leq \frac{N}{2} - n - m .$$

Також можна розрахувати число нейронів в мережах з великим числом шарів, які іноді доцільно використовувати: такі багатошарові нейронні мережі можуть мати менші розмірності матриць вагів нейронів одного шару, ніж двошарові мережі, що реалізують те ж саме відображення. На жаль, строга методика побудови даних мереж поки відсутня.

ДОДАТОК В

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

В.1 Сутність процесу навчання

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

Таким чином, нейронні мережі є універсальними апроксимуючими системами.

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

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

Навчання штучною нейронною полягає в настройці вагових коефіцієнтів $w_{i,j}$ її базових процесорних елементів, результатом чого є виконання мережею конкретних завдань – розпізнавання, оптимізації, апроксимації, управління. Досягнення подібної мети формалізується критерієм якості Q , мінімальне значення $\min_w Q = Q^*$ якого відповідає якнайкращому рішення поставленої задачі.

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

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

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

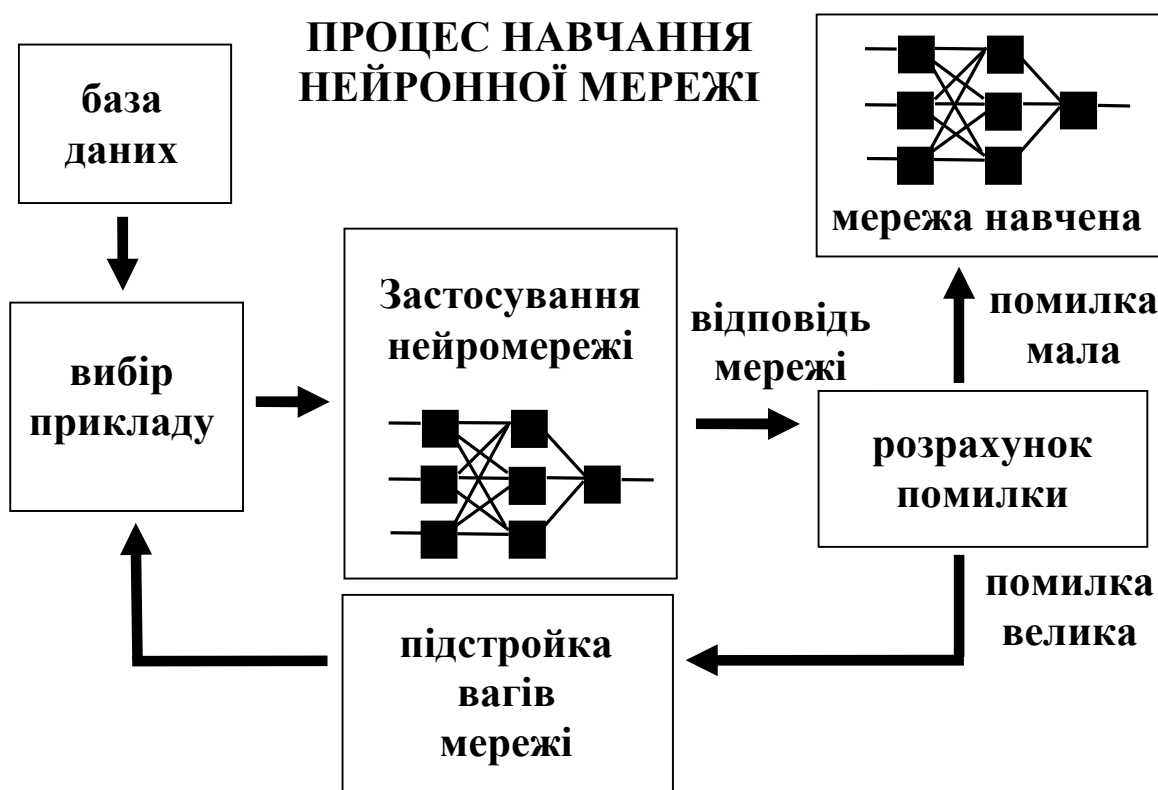


Рисунок В.1 – Ілюстрація процесу навчання ШНМ

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

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

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

Математично процес навчання можна описати таким чином.

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

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

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

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

В.2 Алгоритми навчання нейронних мереж

3.2.1 Класифікація алгоритмів навчання

Для вирішення цього завдання можуть бути використані наступні (ітеративні) алгоритми [1-6]:

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

градієнтний алгоритм (метод швидкого спуску);

методи з одновимірною і двовимірною оптимізацією цільової

функції у напрямі антиградієнта;
метод зв'язаних градієнтів;
методи, що враховують напрям антиградієнта на декількох
кроках алгоритму;

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

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

3 – стохастичні алгоритми оптимізації:

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

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

В.2.2 Алгоритм зворотного розповсюдження.

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

Алгоритм зворотного розповсюдження використовується для навчання багатошарових нейронних мереж з послідовними зв'язками. Як зазначено вище, нейрони в таких мережах діляться на групи із загальним вхідним сигналом – шари, при цьому на кожен нейрон першого шару подаються всі елементи зовнішнього вхідного сигналу, а всі виходи нейронів q -го шару подаються на ко-

жен нейрон шару $(q + 1)$. Нейрони виконують зважене (з синаптичними вагами) підсумовування елементів вхідних сигналів; до даної суми додається зсув нейрона. Над отриманим результатом потім виконується нелінійне перетворення за допомогою активаційної функції. Значення функції активації є вихід нейрона.

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

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

Приведемо словесний опис алгоритму.

Крок 1. Вагам мережі присвоюються невеликі початкові значення.

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

Крок 3. Обчислюється вихід мережі.

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

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

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

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

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

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

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

Для усунення цих недоліків були запропоновані численні модифікації алгоритму зворотного розповсюдження, які зв'язані з використанням різних функцій помилки, різних процедур визначення напрямку і величини кроку і т.п. Розглянемо один з них.

В.3 Метод зворотного розповсюдження помилки

Як було зазначено вище, метод зворотного розповсюдження помилки – метод ВР розроблений для навчання статичних нелінійних багат шарових нейронних мереж і використовує технологію послідовної і пошарової настройки базових процесорних елементів, починаючи з останнього, вихідного шару і закінчуючи настройкою елементів першого шару. «Урок» навчання штучної нейронної мережі може бути повторений необхідне число разів. Для настройки вагових коефіцієнтів базових елементів застосовується δ -правило в його загальній формі: у вигляді алгоритму градієнтного методу мінімізації критерію навчання $Q(\mathbf{e}_u)$. Тут $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)}$ – вектор помилки навчання мережі щодо бажаного (еталонного) виходу $\mathbf{u}^*$ при нелінійному згідно (1.10) перетворенні ввідного вектора $\mathbf{r}$ у вихідний вектор $\mathbf{u} = \mathbf{q}^{(K)}$ останнього шару K . Помилка $\mathbf{e}_u$, явним чином залежить від коефіцієнтів $w_{i,j}^{(K)}$ і може бути використана як аргумент функціонала Q для настройки базових елементів K -го шару. Явної залежності вектора $\mathbf{e}_u$ і функції $Q(\mathbf{e}_u)$ від вагових коефіцієнтів $w_{i,j}^{(l)}$ ($l = \overline{1, K-1}$) немає. Тому у разі використання критерію навчання $Q(\mathbf{e}_u)$ для настройки коефіцієнтів $w_{i,j}^{(l)}$ помилка $\mathbf{e}_u$ в процесі навчання багат шарової нейронної мережі перераховується послідовно в узагальнені помилки $\boldsymbol{\sigma}^{(l)}$, що явно залежать від значень $w_{i,j}^{(l)}$. Настройка коефіцієнтів $w_{i,j}^{(l)}$ здійснюється за δ -правилом, де використовуються узагальнені помилки $\boldsymbol{\sigma}^{(l)}$. Очевидно, що для шару K помилка $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$. У методі ВР далі використовується локальний квадратичний функціонал $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$. Мінімізація $Q(\boldsymbol{\sigma})$ по вагових коефіцієнтах $w_{i,j}^{(l)}$ ($l = K-1, \dots, 1$), що настраюються, методом градієнтного спуску з перерахунком помилки $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$ для шару K в помилку $\boldsymbol{\sigma}^{(l)} = \mathbf{e}(\mathbf{w}^{(l)})$ для шарів $l = K-1, K-2, \dots, 1$ визначає сутність ВР-технології навчання багат шарових нейронних мереж.

В.3.1 Налаштування коефіцієнтів вихідного шару.

Для шару K процедура налаштування вектора всіх вагових коефіцієнтів в K -му шарі $\mathbf{w}^{(k)}$ відповідає зазвичай алгоритму градієнтного спуску:

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

де:

$$\nabla_{\mathbf{w}} Q(\cdot) = \text{col} \left(\frac{\partial Q(\cdot)}{\partial \mathbf{w}_1^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_j^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_{n_k}^{(K)}} \right)$$

вектор розмірністю $(n_k \times 1)$. Вектор $\mathbf{w}^{(k)}$ – складний:

$$\mathbf{w}^{(K)} = \text{col}(\mathbf{w}_1^{(K)}, \dots, \mathbf{w}_i^{(K)}, \dots, \mathbf{w}_{n_K}^{(K)})$$

де:

$$\mathbf{w}^{(K)} = \text{col}(w_{i,1}^{(K)}, \dots, w_{i,m}^{(K)}, \dots, w_{n_{K-1}}^{(K)})$$

$\mathbf{q}^{(K-1)}(k)$ – вектор розмірністю $((n_{K-1} + 1) \times 1)$ базових елементів в шарі K :

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

Сигнал ініціалізації $q_0^{(K-1)}$ зазвичай приймається рівним +1,

Градієнт

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

де похідна:

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

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

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

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

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

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

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

З урахуванням приведених матриць Якобі градієнт функціонала настройки K -го шару багат шарової нейронної мережі можна записати таким чином:

$$\nabla_{\mathbf{w}} Q(\mathbf{e}_u) = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \right) \left(\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} \right) \mathbf{e}_u = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \text{diag}(f'_s(\mathbf{s}^{(K)})) \right) \mathbf{e}_u,$$

а алгоритм настройки вагових коефіцієнтів K -го шару має вигляд

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

Рекурентний алгоритм (B.2) можна розділити на n_K алгоритмів навчання i -х базових елементів, $i = \overline{1, n_K}$, оскільки діагональна матриця $\partial \mathbf{s}^{(K)} / \partial \mathbf{w}^{(K)}$ складається з однакових елементів $\mathbf{q}^{(K-1)}$.

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

Узагальнена помилка σ , навчання мережі для шару K рівна $f'_s(s_i^{(K)})e_{ui}$. У багатошарових мережах останній вихідний шар виконується зазвичай у вигляді мадалини з лінійними функціями активації і $f'_s(s) = 1$. Тоді $\sigma_i = e_{ui}$ і алгоритм настройки вагових коефіцієнтів шару K приймає вигляд:

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

У алгоритмах (B.2)-(B.4) $k = 1, 2, \dots$ – номер інтеграції. Початкові значення вагових коефіцієнтів $\mathbf{w}^{(K)}(0)$ задаються довільно, але бажано в околиці значень $\mathbf{w}^{*(K)}$, для яких $Q(\mathbf{w}_{(K)}^*) = \min_{\mathbf{w}} Q(\mathbf{w}^{(K)})$.

В.3.2. Налаштування вагових коефіцієнтів прихованих шарів нейронної мережі

Для прихованих шарів $l = \overline{K-1, 1}$ помилки навчання в явному вигляді не можуть бути визначені. Проте очевидно, що якщо $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)} \neq 0$, то через архітектуру мережі існує ненульова узагальнена помилка і для всіх попередніх шарів. Суть методу ВР якраз і полягає в послідовному визначенні «прихованих» помилок, починаючи з обчислення помилки $\sigma^{(K)}$ у вихідному шарі K , потім помилки $\sigma^{(K-1)}$ в шарі $K-1$ і так до вхідного шару 1. Всі приховані по-

милки $\sigma^{(l)} = \sigma(\mathbf{w}_i^l)$ залежать явно від вагових коефіцієнтів, що настраюються $\mathbf{w}_i^l$. Розглянемо суть необхідних перетворень в δ -правилі (В.2), яке використовує ідею обчислення прихованих помилок в рекурентній процедурі настройки вагових коефіцієнтів в шарах $l = \overline{K-1, 1}$. З цією метою складемо алгоритм настройки вагових коефіцієнтів для окремих i -х базових елементів в шарі l :

$$\mathbf{w}_i^{(l)} = \text{col}(w_{i,0}^{(l)}, \dots, w_{i,j}^{(l)}, \dots, w_{i,n_l}^{(l)}).$$

При виведенні алгоритму настройки вагових коефіцієнтів в шарах $K-1, K-2, \dots, 1$ вкажемо, що в загальному випадку градієнт функціонала Q щодо вектора вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ можна обчислити як:

$$\nabla \mathbf{w}_i^{(l)} Q = \frac{\partial \mathbf{s}^{(l)}}{\partial \mathbf{w}_i^{(l)}} \frac{\partial \mathbf{q}^{(l)}}{\partial \mathbf{s}^{(l)}} \frac{\partial Q}{\partial \mathbf{q}^{(l)}}.$$

Тут, як і для шару K , вважаємо для спільності і для спрощення подальших перетворень вектор $\mathbf{q}^{(l)} = \text{col}(q_0^{(l)}, q_1^{(l)}, \dots, q_{n_l}^{(l)})$.

У практичних застосуваннях $q_0^{(l)} = +1$. У подальшому виводі вважатимемо, що ініціалізуючі сигнали $q_0^{(l)}$ є виходами фіктивних, «нульових» базових елементів з функціями активації $f(s_0^{(l)})$ того ж типу, що і в реальних елементах. Тоді випадку $q_0^{(l)} = +1$ відповідає робоча точка $f(s_0^{(l)}) = 1$.

З урахуванням зробленого припущення розмірність матриці Якобі $\partial \mathbf{s}^{(l)} / \partial \mathbf{w}_i^{(l)}$, де $\mathbf{s}^{(l)} = \text{col}(s_0^{(l)}, s_1^{(l)}, \dots, s_{n_l}^{(l)})$, рівна $(n_{l-1} + 1) \times (n_l + 1)$, відповідно розмірність матриці Якобі $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)} = \dim \text{diag}(f'_s(s_0^{(l)})) = (n_l + 1) \times (n_l + 1)$. Якщо покласти $f'_s(s_0^{(l)}) = 1$, то верхній діагональний елемент матриці $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)}$ буде рівний нулю. Отже, матриці

$$\left(\frac{\partial \mathbf{s}^{(l)}}{\partial \mathbf{w}_i^{(l)}} \right) = \left\| \frac{\partial s_i^{(l)}}{\partial w_{i,j}^{(l)}} \right\|, \quad i = \overline{0, n_l}, \quad j = \overline{0, n_{l-1}}; \quad \left(\frac{\partial \mathbf{q}^{(l)}}{\partial \mathbf{s}^{(l)}} \right) = \text{diag}(f'_s(s^{(l)}))$$

складені. Позначимо першу з них символом $\mathbf{S}^{(l)}$, другу $\mathbf{F}^{(l)}$. Елементами матриці служать приватні похідні $\partial s_i^{(l)} / \partial w_{i,j}^{(l)}$. Оскільки

$$s_i^{(l)} = \sum_{j=0}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)},$$

матриця $\mathbf{S}^{(l)}$ має наступну структуру:

$$\mathbf{S}^{(l)} = \begin{bmatrix} q_0^{(l-1)} & q_0^{(l-1)} & \dots & q_0^{(l-1)} \\ q_1^{(l-1)} & q_1^{(l-1)} & \dots & q_1^{(l-1)} \\ \dots & \dots & \dots & \dots \\ q_{n_{j-1}}^{(l-1)} & q_{n_{j-1}}^{(l-1)} & \dots & q_{n_{j-1}}^{(l-1)} \end{bmatrix} = [\mathbf{q}^{(l-1)} : \mathbf{q}^{(l-1)} : \dots : \mathbf{q}^{(l-1)}],$$

розмірність $\mathbf{S}^{(l)}$ рівна $(n_{l-1} + 1) \times (n_{l-1} + 1)$ і $\mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)})$.

Елементами діагональної $\mathbf{F}^{(l)}$ матриці є приватні похідні функцій активації $f(s)$ по своєму аргументу $s_i (i = \overline{0, n_l})$. Можна показати, що добуток матриць $\mathbf{S}^{(l)}$ і $\mathbf{F}^{(l)}$ зводиться до зовнішнього добутку векторів $\mathbf{q}^{(l-1)}$ і $\mathbf{f}'_s(s^{(l)}) = (f'_s(s_0^{(l)}), \dots, f'_s(s_{n_l}^{(l)}))$.

$$\mathbf{S}^{(l)} \mathbf{F}^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(\mathbf{s}^{(l)}))^T.$$

Похідна скалярного функціонала $Q(\boldsymbol{\sigma})$ по векторному аргументу $\mathbf{q}^{(l)} (l = \overline{K-1, 1})$ по своєму сенсу є вектор чутливості функціонала Q до зміни вектора виходу $\mathbf{q}^{(l)}$ l -го шару мережі. Для шару l введемо вектор чутливості функціонала Q :

$$\boldsymbol{\Lambda}^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \text{col} \left(\frac{\partial Q}{\partial q_0^{(l)}}, \frac{\partial Q}{\partial q_1^{(l)}}, \dots, \frac{\partial Q}{\partial q_{n_l}^{(l)}} \right) = \text{col}(\lambda_0^{(l)}, \lambda_1^{(l)}, \dots, \lambda_{n_l}^{(l)}).$$

Приватні функції чутливості $\lambda_i^{(l)}$ обчислюються, по-перше, відповідно до правила диференціювання складних функцій; по-друге, відповідно до структури багатошарової нейронної мережі і мають вигляд:

$$\lambda_i^{(l)} = \frac{\partial Q}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \frac{\partial Q}{\partial q_j^{(l+1)}} \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \lambda_j^{(l+1)} d_{j,i}^{(l+1)} = (\boldsymbol{\Lambda}^{(l+1)})^T \mathbf{d}_i^{(l+1)}, \quad i = \overline{1, n_l}.$$

Повний вектор функцій чутливості функціонала Q до зміни вектора виходу нейронів l -го шару є

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

Матрицю $\mathbf{D}^{(l+1)}$ назвемо матрицею переходу у зворотному напрямі, оскільки для розрахунку функції $\Lambda^{(l)}$ використовуються функції чутливості $\Lambda^{(l+1)}$ і $\mathbf{D}^{(l+1)}$ подальшого шару. Матриця $\mathbf{D}^{(l+1)}$ складена з вектор-рядків $(\mathbf{d}_i^{(l+1)})^T$ де $i = \overline{0, n_l}$. Нагадаємо, що за умовчанням $q_0^{(l)} = +1$, а вагові коефіцієнти ініціалізуючих входів $w_{j,0}^{(l+1)}$ всіх базових елементів в шарі $l+1$ також настроюються. Елементами векторів за визначенням є приватні похідні:

$$\frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \frac{\partial}{\partial q_i^{(l)}} f \left(\sum_{j=0}^{n_l} w_{j,i}^{(l+1)} q_i^{(l)} \right) = \left(\frac{\partial f(s_j^{(l+1)})}{\partial s_j^{(l+1)}} \right) \left(\frac{\partial s_j^{(l+1)}}{\partial q_i^{(l)}} \right) = f'_s(s_j^{(l+1)}) w_{j,i}^{(l+1)}.$$

Для всієї множини індексів $i = \overline{0, n_l}, j = \overline{1, n_{l+1}}$ матриця переходу $\mathbf{D}^{(l+1)}$ має наступну структуру:

$$\mathbf{D}^{(l+1)} = \frac{\partial \mathbf{q}^{(l+1)}}{\partial \mathbf{q}^{(l)}} = \begin{bmatrix} \frac{\partial}{\partial q_0^{(l)}} \\ \frac{\partial}{\partial q_1^{(l)}} \\ \dots \\ \frac{\partial}{\partial q_{n_l}^{(l)}} \end{bmatrix} \times (q_0^{(l+1)}, q_1^{(l+1)}, \dots, q_{n_{l+1}}^{(l+1)}) = \begin{bmatrix} \frac{\partial q_0^{(l+1)}}{\partial q_0^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_0^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_0^{(l)}} \\ \frac{\partial q_0^{(l+1)}}{\partial q_1^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_1^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_1^{(l)}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial q_0^{(l+1)}}{\partial q_{n_l}^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_{n_l}^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_{n_l}^{(l)}} \end{bmatrix}.$$

Відзначимо, що розмірність матриці переходу $\mathbf{D}^{(l+1)}$ рівна $[(n_{l+1} + 1) \times (n_l + 1)]$. Приватні похідні в матриці $\mathbf{D}^{(l+1)}$ визначені раніше. Тепер можна скласти вектор-рядки $(\mathbf{d}_i^{(l+1)})^T$:

$$(\mathbf{d}_i^{(l+1)})^T = (f'_s(s_0^{(l+1)})w_{0,i}^{(l+1)}, f'_s(s_1^{(l+1)})w_{1,i}^{(l+1)}, \dots, f'_s(s_{n_{l+1}}^{(l+1)})w_{n_{l+1},i}^{(l+1)})$$

розмірністю $[1 \times (n_{l+1} + 1)]$. Введемо матриці:

а)

$$\mathbf{W}^{(l+1)} = [\mathbf{w}_0^{(l+1)} : \mathbf{w}_1^{(l+1)} : \dots : \mathbf{w}_{n_{l+1}}^{(l+1)}] = \|\mathbf{w}_{i,j}^{(l)}\|, \quad i = \overline{0, n_{l+1}}; \quad j = \overline{0, n_l},$$

тобто

$$\dim \mathbf{W}^{(l+1)} = (n_l + 1) \times (n_{l+1} + 1),$$

де

$$\mathbf{w}_i^{(l+1)} = \text{col}(w_{i,0}^{(l+1)}, \dots, w_{i,j}^{(l+1)}, \dots, w_{i,n_l}^{(l+1)});$$

б)

$$\mathbf{F}^{(l+1)} = \text{diag}(f'_s(\mathbf{s}^{(l+1)})) = \begin{bmatrix} f'_s(s_0^{(l+1)}) & \dots & 0 & 0 \\ 0 & f'_s(s_1^{(l+1)}) & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & f'_s(s_{n_{j-1}}^{(l+1)}) \end{bmatrix}$$

$$\dim \mathbf{F}^{(l+1)} = (n_{l+1} + 1) \times (n_{l+1} + 1).$$

Відмітимо, що для фіктивного «нульового» базового елементу $q_0^{(l+1)} = +1$ і елемент матриці $\mathbf{F}^{(l+1)} f'_s(s_0^{(l+1)}) = 0$. З урахуванням введених матриць $\mathbf{W}^{(l+1)}$ і $\mathbf{F}^{(l+1)}$ матрицю $\mathbf{D}^{(l+1)}$ запишемо у вигляді:

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

Тепер можна записати формулу обчислення градієнта функціонала навчання багат шарової нейронної мережі $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$ по вагових коефіцієнтах будь-якого шару $l = \overline{K-1, 1}$:

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

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

В.3.3 Алгоритм навчання багатошарової нейронної мережі по методу зворотного розповсюдження помилки.

Приведемо алгоритм навчання нейронної мережі методом зворотного розповсюдження помилки в одній з можливих форм, витікаючої з прийнятої методики виводу алгоритму:

а)

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k+1)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i^{(K)}(k+1) \mathbf{q}^{(K-1)}(k+1); \\ \gamma &> 0; \quad k = 0, 1, 2, \dots; \quad e_{ui} = u_i^* - u_i; \quad i \in \overline{1, n_K}; \\ \mathbf{q}^{(K-1)} &= \text{col}(q_0^{(K-1)}, q_1^{(K-1)}, \dots, q_j^{(K-1)}, \dots, q_{n_{K-1}}^{(K-1)}); \end{aligned} \quad (\text{В.8})$$

б)

$$\begin{aligned} \mathbf{w}_i^{(l)}(k+1) &= \mathbf{w}_i^{(l)}(k) - \gamma \mathbf{q}^{(l-1)}(k+1) (\mathbf{f}'_s(s_i^{(l)}(k+1)))^T \Lambda^{(l)}(k); \\ \Lambda^{(l)}(k) &= \mathbf{D}^{(l+1)} \Lambda^{(l+1)}(k); \\ \mathbf{D}^{(l+1)}(k) &= (\mathbf{W}^{(l+1)}(k))^T \text{diag}(f'_s(\mathbf{s}^{(l)}(k))); \\ \gamma &> 0; \quad l = \overline{K-1, 1}; \quad \mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, q_1^{(l-1)}, \dots, q_{l_{K-1}}^{(l-1)}); \quad i = \overline{0, n_l} \end{aligned} \quad (\text{В.9})$$

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

$$\begin{aligned} \mathbf{w}_i^{(K-1)}(K+1) &= \mathbf{w}_i^{(K-1)}(K) - \gamma \mathbf{q}^{(K-2)}(K+1) (\mathbf{f}'_s(s_i^{(K-1)}(K+1)))^T \Lambda^{(K-1)}(K), \\ \Lambda^{(K-1)}(K) &= \mathbf{D}^{(K)}(K) \Lambda^{(K)}(K); \\ \mathbf{D}^{(K)}(K) &= \mathbf{W}^{(K)}(K) \text{diag}(f'_s(\mathbf{s}^{(K)}(K))) = \mathbf{W}^{(K)}(K); \\ \Lambda^{(K)}(K) &= \text{col} \left(\frac{\partial Q}{\partial \mathbf{q}_1^{(K)}}, \dots, \frac{\partial Q}{\partial \mathbf{q}_{n_K}^{(K)}} \right) = -\mathbf{e}_u(K). \end{aligned}$$

тоді

$$\begin{aligned}\Lambda^{(K-1)}(k) &= -\mathbf{W}^{(K)}(k)\mathbf{e}_u(k), \\ (\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k) &= \\ &= -(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \mathbf{W}^{(K)}(k)\mathbf{e}_u(k) = \sigma_1^{(K-1)}(k+1)\end{aligned}$$

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

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

Аргумент k позначає момент використання відповідних значень заданих множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$ на k -й ітерації обчислень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ на «уроках» навчання, число яких залежить від числа шарів мережі і кількості нейронів в шарах.

Фаза навчання завершена, коли досягнутий для заданих в своєму класі множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$. Мережа фіксує цей стан в значеннях своїх вагових коефіцієнтів і відповідних «віртуальних» зсувах центрів симетрії функцій активації і їх похідних активованих базових елементів і зберігає цей стан після зняття сигналу $\mathbf{q}^{(0)} = \mathbf{r}$ на входах мережі а виді віртуальної статичної нелінійної характеристики, зібраної з фрагментів сигмоїдних функцій. Якщо тепер па входах пред'явити ту ж множину або близьку до неї з того ж класу, то нейронна мережа практично миттєво відновить на виході ту ж множину $\mathbf{u} = \mathbf{q}^{(K)}$, що і після фази навчання. Цей етап функціонування мережі називають фазою спогаду. Мережа, таким чином, виконує функцію асоціативної пам'яті. Якщо ж множини $\mathbf{u}^*$, $\mathbf{q}^{(0)} = \mathbf{r}$ є функціями часу: $\mathbf{q}^{(0)} = \mathbf{r}(t)$, $\mathbf{u}^* = \mathbf{u}^*(t)$, то стандартний алгоритм ВР

стає неавтономним і вимагає завдання всієї множини значень змінних $\mathbf{r}(t)$ і $\mathbf{u}^*(t)$. Неминуче зростання числа шарів і базових елементів в них і пов'язане з цим збільшення тривалості навчання викликає необхідність забезпечення стійкості процесу настройки мережі по алгоритму (В.8) (В.9).

Очевидно, що в завданнях управління $\mathbf{u}^*(t)$ – функція оптимального управління динамічним об'єктом. Управління за допомогою багатошарової нейронної мережі в двохетапному режимі можливо, якщо тільки функція оптимального управління наперед розрахована і може бути визначена вся множина значень помилки навчання мережі $\mathbf{e}_u(t)$ на інтервалі її визначення. Використання іншої вимірjuвальної інформації, наприклад помилки відтворення заданої функції $\mathbf{r}(t)$ на виході динамічного об'єкту, вносить запізнювання до алгоритму настройки і приводить до нестійкого процесу. Виникає проблема синтезу динамічних алгоритмів навчання багатошарової нейронної мережі, що враховують вплив інерційності, що вноситься до процесу настройки по алгоритму ВР.

В.4 Перенавчання і узагальнення нейронних мереж

Одна з найбільш серйозних труднощів викладеного підходу полягає в тому, що таким чином ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати, – помилку, яку можна чекати від мережі, коли їй подаватимуться абсолютно нові спостереження. Інакше кажучи, ми хотіли б, щоб нейронна мережа володіла здатністю узагальнювати результат на нові спостереження. Насправді мережа навчається мінімізувати помилку на навчальній множині, і у відсутність ідеальної і нескінченно великої навчальної множини це зовсім не те ж саме, що мінімізувати «справжню» помилку на поверхні помилок в наперед невідомій моделі явища.

Найсильніше ця відмінність виявляється в проблемі перенавчання. При дуже близькій підгонці явище простіше буде продемонструвати не для нейронної мережі, а на прикладі апроксимації за допомогою поліномів, при цьому суть явища абсолютно та ж.

Поліном (або многочлен) – це вираз, що містить тільки константи і цілі ступені незалежної змінної. Графіки поліномів можуть мати різну форму. Чим вище ступінь многочлена (і, тим самим, ніж більше членів в нього входить), тим більше складною може бути ця форма. Якщо у нас є деякі дані, ми можемо поставити мету підігнати до них поліноміальну криву (модель) і отримати, таким чином, пояснення для наявної залежності. Проте наші дані можуть бути зашумлені, тому не можна вважати, що сама краща модель задається кривою, яка в точності проходить через все наявні крапки. Поліном низького порядку може бути недостатньо гнучким засобом для апроксимації даних, тоді як поліном високого порядку може виявитися занадто гнучким і буде точно слід даним, приймаючи при цьому хитромудру форму, що не має ніякого відношення до форми справжньої залежності.

Нейронна мережа стикається з точно такою ж трудностю. Мережі з великим числом вагів моделюють складніші функції і, отже, схильні до перенавчання. Мережа ж з невеликим числом вагів може опинитися недостатньо гнучкою, щоб змодельовати наявну залежність. Наприклад, мережа без проміжних шарів насправді моделює звичайну лінійну функцію.

Як же вибрати «правильний» ступінь складності для мережі? Складніша мережа майже завжди дає меншу помилку, але це може свідчити не про хорошу якість моделі, а про перенавчання.

Відповідь полягає в тому, щоб використовувати механізм контрольної перекресної перевірки (крос-перевірки), при якому частина навчальних спостережень резервується і не використовується в процесі навчання по алгоритму зворотного розповсюдження. Натомість у міру роботи алгоритму вона використовується для незалежної перевірки результату. На самому початку роботи помилка мережі на навчальній і перевірочній множинах буде однаковою, але у міру того як мережа навчається, помилка навчання, природно, убиває, і, поки навчання зменшує дійсну функцію помилок, помилка на контрольній (перевірочній) множині також убиватиме. Якщо ж контрольна помилка перестала убивати або навіть стала рости, це указує на те, що мережа почала дуже близько апроксимувати дані і навчання слід зупинити.

Відмічене явище занадто точної апроксимації в процесі навчання і називається перенавчанням. Якщо таке трапляється, то рекомендується зменшити число прихованих елементів і/або шарів, бо мережа є дуже могутньою для даного завдання. Якщо ж мережа, навпаки, була узята недостатньо складною для того, щоб моделювати наявну залежність, то перенавчання, швидше за все, не відбудеться, і обидві помилки – навчання і перевірки – не досягнуть достатнього рівня трохи.

Описані проблеми з локальними мінімумами і вибором розміру мережі призводять до того, що при практичній роботі з нейронними мережами, як правило, доводиться експериментувати з великим числом різних мереж, деколи навчаючи кожную з них по кілька разів (щоб не бути введеним в оману локальними мінімумами) і порівнюючи отримані результати. Головним показником якості результату тут є контрольна помилка. При цьому, відповідно до загальнонаукового принципу, згідно якому при інших рівних слід віддати перевагу простішій моделі, з двох мереж з приблизно рівними помилками контролю має сенс вибрати ту, яка менше.

Необхідність багатократних експериментів веде до того, що перевірна множина починає грати ключову роль у виборі моделі, тобто стає частиною процесу навчання. Тим самим ослабляється її роль як незалежного критерію якості моделі – при великому числі експериментів є ризик вибрати «вдалу» мережу, що дає добрий результат на перевірочній (контрольному) множині.

Для того, щоб додати остаточній моделі належну надійність, часто (принаймні, коли об'єм навчальних даних це дозволяє) поступають так: резервують ще одну – тестову множину спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на навчальній і контрольній множині, реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки один раз: якщо її використовувати повторно для коректування процесу навчання, то вона фактично перетвориться на контрольну множину.

ДОДАТОК Г

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ

Г.1 Моделі систем, використовувані при нейромережевій ідентифікації

Основним питанням в нейрорегулювання є отримання адекватних моделей динамічних систем. Питання про існування дискретних моделей, або іншими словами про спостереженість нелінійних систем, є ключовим. У загальному випадку завдання побудови математичної моделі об'єкту полягає у виборі структури і оцінці її параметрів так, щоб при використанні критерію мінімуму деякої функції різниці розрахункових і експериментальних даних дотримувалася умова близькості моделі досліджуваному процесу. Застосування нейронних мереж як моделі може служити альтернативою класичним методам математичного опису систем, оскільки в цьому випадку точне знання внутрішніх процесів не є необхідною умовою моделювання.

При нейронній ідентифікації нелінійних систем найчастіше знаходять застосування наступні основні нелінійні дискретні моделі.

У найбільш загальному вигляді, прийнятому в класичній теорії управління, опис стохастичних нелінійних систем в просторі станів розглядався в роботі Наренди К.С. і Партасараті К [37].

$$z(k+1) = \Psi[z(k), u(k), \phi(k)] ;$$

$$y(k) = \Phi[z(k), u(k), \phi(k)],$$

де $z(k) = [z_1(k), z_2(k), \dots, z_{k_z}(k)]^T$ – вектор стану системи;

$u(k) = [u_1(k), u_2(k), \dots, u_{k_u}(k)]^T$ – вектор вхідних сигналів;

$y(k) = [y_1(k), y_2(k), \dots, y_{k_y}(k)]^T$ – вектор вихідних сигналів;

$\Psi[\cdot]$, $\Phi[\cdot]$ – деякі статичні нелінійні функції;

$\phi(k)$, (k) – шум процесу і шум вимірювань відповідно.

Леонартіс І. Дж. і Биллінгз С.Д.[38] запропонували для опису нелінійних систем використовувати так звану нелінійну модель авторегресії ковзаючого середнього з додатковими вхідними сигналами (NARMAX-модель).

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u), e(k-1), \dots, e(k-n_e)] + e(k) \quad (\Gamma.1)$$

де $y(k) \in \mathbf{R}^{M \times 1}$ – вихідний сигнал об'єкту;

$u(k) \in \mathbf{R}^{N \times 1}$ – вхідний сигнал об'єкту;

$e(k) \in \mathbf{R}^{M \times 1}$ – різниця вихідних сигналів об'єкту і моделі;

$f[\cdot]$ – нелінійна функція перетворення $f: \mathbf{R}^{(k_y M + k_u N + k_e M) \times 1} \rightarrow \mathbf{R}^{M \times 1}$;

n_y, n_u, n_e – порядки запізнювання по вихідному, вхідному сигналам об'єкту і помилці вимірювань відповідно.

Скалярна форма рівняння (Г.1) має вигляд:

$$\begin{aligned} y_1(k) = & f_1[y_1(k-1), \dots, y_1(k-n_y), y_2(k-1), \dots, y_2(k-n_y), y_M(k-1), \dots, y_M(k-n_y), \\ & u_1(k-1), \dots, u_1(k-n_u), u_2(k-1), \dots, u_2(k-n_u), u_N(k-1), \dots, u_N(k-n_u), \\ & e_1(k-1), \dots, e_1(k-n_e), e_2(k-1), \dots, e_2(k-n_e), e_M(k-1), \dots, e_M(k-n_e)] + e_i(k) \end{aligned} \quad (\Gamma.2)$$

Нелінійні системи можуть бути представлені нелінійною NARX-моделлю:

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)] + e(k) \quad (\Gamma.3)$$

або в скалярному вигляді

$$\begin{aligned} y_1(k) = & f_1[y_1(k-1), \dots, y_1(k-k_y), y_2(k-1), \dots, y_2(k-k_y), y_M(k-1), \dots, y_M(k-k_y), \\ & u_1(k-1), \dots, u_1(k-k_u), u_2(k-1), \dots, u_2(k-k_u), u_N(k-1), \dots, u_N(k-k_u)] + e_i(k). \end{aligned} \quad (\Gamma.4)$$

Представлення об'єктів у вигляді NARMAX або NARX моделей грають фундаментальну роль при дослідженні нелінійних об'єктів за допомогою штучних нейронних мереж. При цьому важливо розглянути узагальнення моделей (Г.1) - (Г.4), що враховує різні види нелінійностей по вхідних і вихідних змін-

них. Наренда виділив чотири такі моделі:

– модель 1 (рис. Г.1):

$$y(k+1) = \sum_{i=1}^{k_y-1} a_i y(k-i) + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\Gamma.5)$$

– модель 2 (рис. Г.2):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + \sum_{i=1}^{k_u-1} b_i u(k-i); \quad (\Gamma.6)$$

– модель 3 (рис. Г.3):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\Gamma.7)$$

– модель 4 (рис. Г.4):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)]. \quad (\Gamma.8)$$

Модель (Г.4) є найбільш загальною зі всіх моделей вхід/вихід, описаних вище. При цьому модель (Г.8) є аналітично вирішуваною і найбільш зручною для практичних застосувань. Внаслідок цього вона використовується в роботі при ідентифікації нелінійних об'єктів за допомогою штучних нейронних мереж.

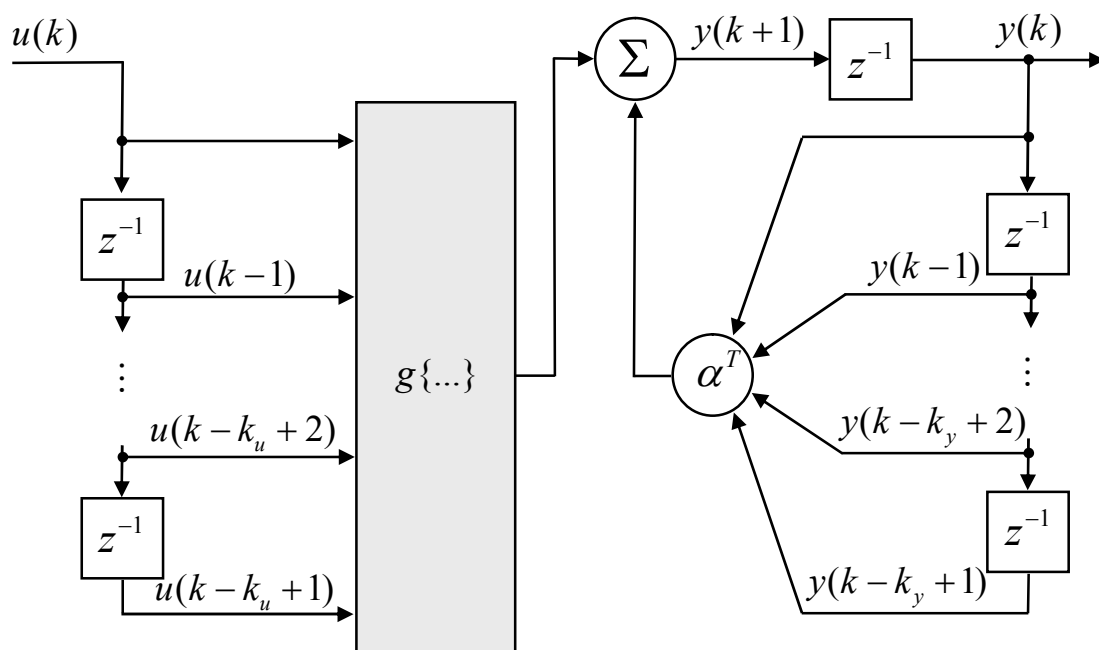


Рисунок Г.1 – Модель 1. Вихід моделі нелінійно залежить від вхідних сигналів і лінійно від вихідних

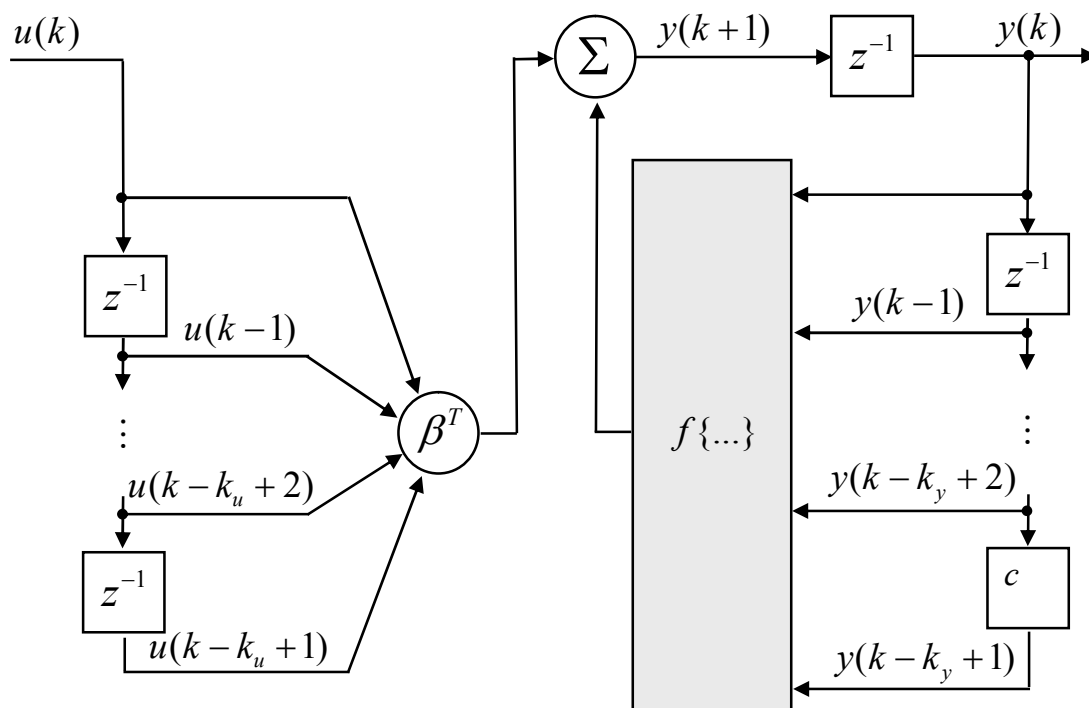


Рисунок Г.2 – Модель 2. Вихід моделі лінійно залежить від вхідних сигналів і нелінійно від вихідних

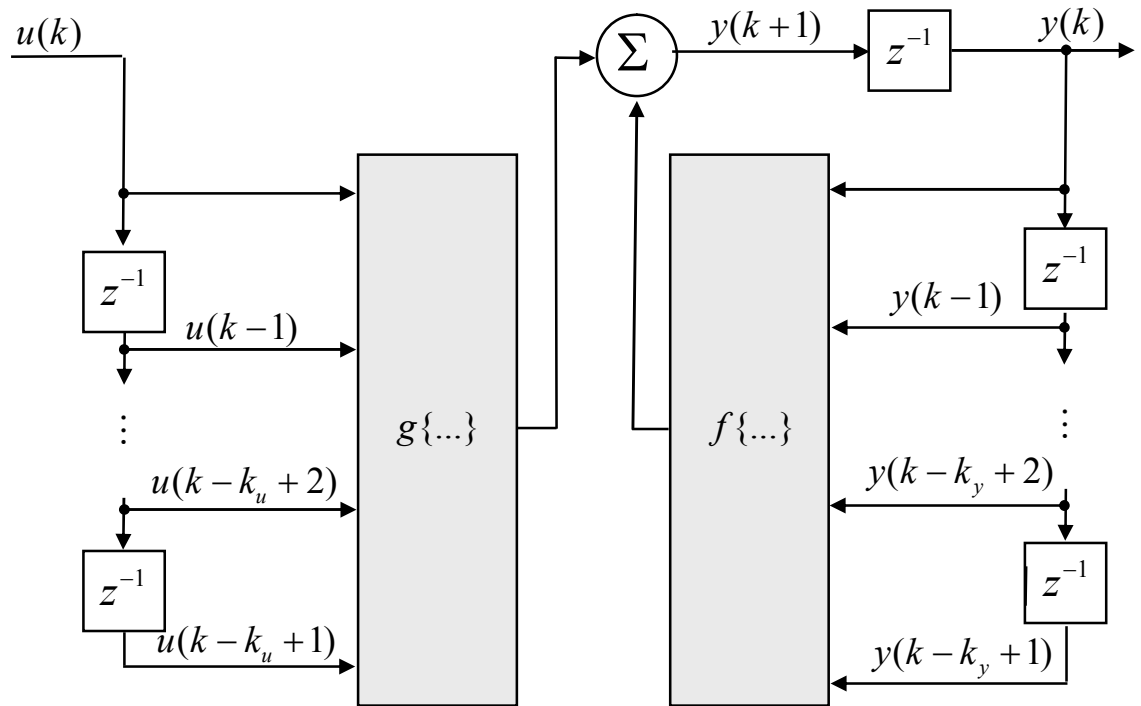


Рисунок Г.3 – Модель 3. Результовий сигнал є адитивною функцією нелінійних перетворень вхідних і вихідних сигналів

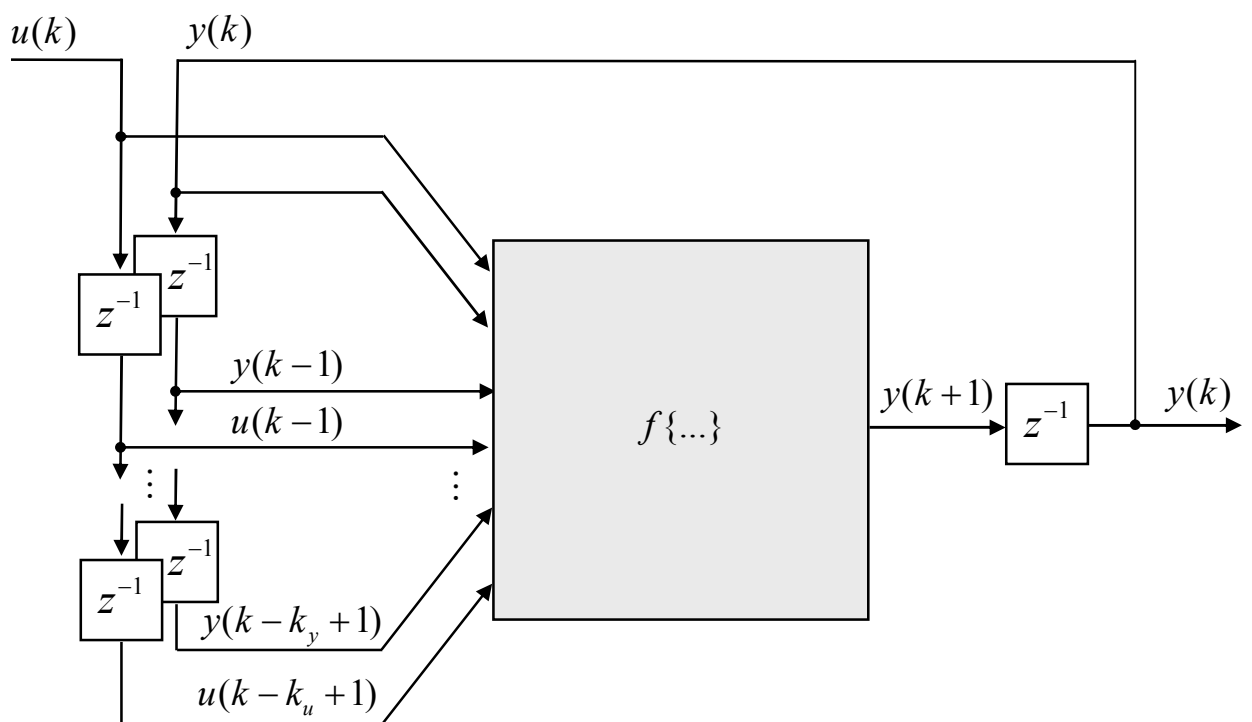


Рисунок Г.4 – Модель 4. Вихід моделі нелінійно залежить як від вхідних, так і від вихідних сигналів

Г.2 Нейронні мережі як універсальні моделі

Оскільки тип і взаємозв'язок нелінійностей в об'єкті, як правило, не відомі, для моделювання слід використовувати якомога більш універсальні структури моделей. Нейронні мережі є саме такими структурами.

Нелінійні моделі в просторі станів можуть бути реалізовані за допомогою рекурсивних нейронних мереж з внутрішніми зворотними зв'язками (мережі Елмана). Ідентифікація параметрів цих мереж вимагає спеціальної і щодо трудомісткої форми алгоритму навчання. Крім того, доказ стійкості таких нелінійних динамічних моделей практично неможливо зробити.

У роботі використовуються прямонаправленні мережі, що характеризуються наявністю зв'язків між нейронами тільки в прямому напрямі без зворотних зв'язків усередині мережі, – багат шаровий персептрон. Найбільш часто використовується мережа, багат шаровий персептрон (БП). За допомогою БП можна апроксимувати з бажаною точністю не тільки будь-які статичні функції. Попередні значення вхідних/вихідних координат у вхідному векторі дозволяють додати прямонаправленим мережам динамічні властивості.

Володіючи схожими з радіально-базисними мережами (РБМ) властивостями, БП забезпечує компактніше представлення нелінійного об'єкту, оскільки число параметрів в мережі РБМ експоненціально зростає із збільшенням розмірності вхідного простору, тоді як для БП ця залежність лінійна. Важливою є наочність, що виражається в тому, що персептронне представлення моделі еквівалентне традиційному. Це дозволяє використовувати для визначення параметрів ШНМ добре розвинені методи оцінювання, використовувані при традиційному підході.

Модель системи в просторі станів може бути успішно реалізована за допомогою прямонаправлених мереж, якщо змінні стани сформувати на виході мережі і потім завести їх на вхід мережі із затримкою. Але для цього потрібний, щоб всі змінні стани були вимірювані, оскільки тільки в цьому випадку можна забезпечити тренування мережі.

NARX мережеві моделі параметризуються лінійно і тому використовують стандартні методи тренування мережі. Внаслідок цього саме даний вид моделі найчастіше використовується при нейронній ідентифікації систем. Для простоти викладу надалі розглядаємо системи з одним входом і одним виходом SISO (singl input singl output), хоча всі висновки можуть бути поширені і на MIMO (multi input multi output) системи. Прямий нейронний NARX моделі нелінійної динамічної системи відповідає наступний вираз:

$$\hat{y}(k) = f\left(y(k-1)\dots y(k-n_y), u(k-n_k)\dots u(k-n_u-n_k+1), \mathbf{w}\right), \quad (\Gamma.9)$$

а нейронній інверсній NARX моделі:

$$\hat{u}(k) = g\left(y(k+n_k)\dots y(k+n_k-n_y), u(k-1)\dots u(k-n_u+1), \mathbf{w}\right) \quad (\Gamma.10)$$

де f, g – функції, що реалізуються нейронними мережами;

u, y – вхідна і вихідна координати відповідно;

$\hat{u}, \hat{y}$ – значення координат на виході мережі;

$\mathbf{w}$ – матриця лінійних параметрів мережі;

n_k – запізнювання в системі, найчастіше $n_k = 1$.

Вирази (Г.9) і (Г.10) можна представити графічно у вигляді структурних схем, показаних на рис. Г.5 і рис. Г.6. NARMAX моделі характеризуються вищою якістю ідентифікації при зашумлених тренувальних даних. У [39] пропонується спосіб модифікації критерію якості алгоритму тренування. Результати ідентифікації за допомогою NARX моделей наближаються в цьому випадку до результатів, отриманих за допомогою NARMAX моделей. Дані NARX нейромоделі можуть бути успішно застосовані для вирішення поставлених завдань.

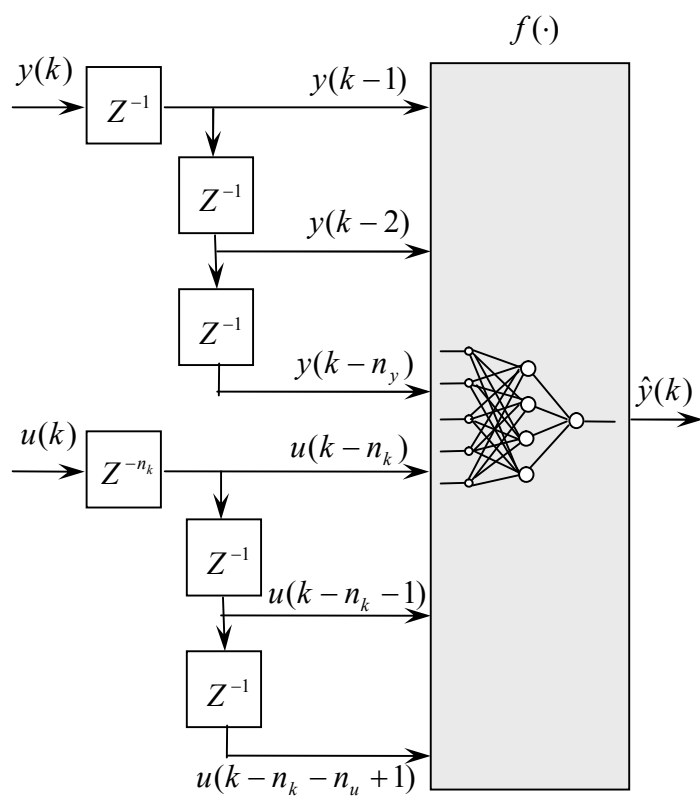


Рисунок Г.5— Пряма мережева NARX модель

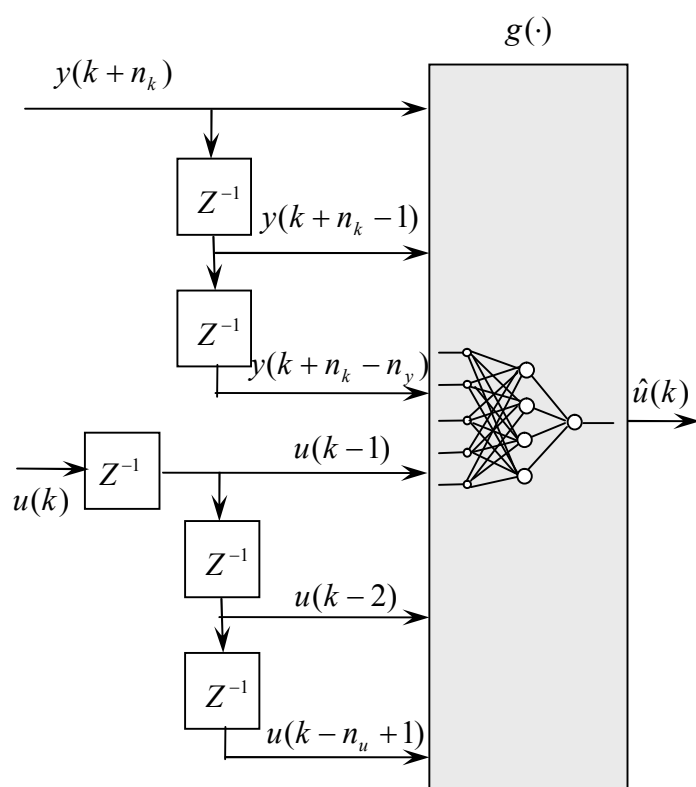


Рисунок Г.6 – Інверсна мережева NARX модель

Г.3 Ідентифікація динамічних систем за допомогою нейронних мереж

Завдання теорії регулювання, що займається пошуком моделей динамічних систем на основі апріорної інформації і вимірюваних входних/вихідних сигналів системи, носить назву ідентифікація. Відомі класичні методи ідентифікації нелінійних систем застосовні тільки у поєднанні з певним типом моделі (наприклад, ряди Вальтера, моделі Хаммерштейна і Вінера), що описує певний клас нелінійних динамічних систем. У роботі під термінами «ідентифікація системи», «тренування» і «навчання» розуміється процес підстроювання вагових коефіцієнтів мережі за допомогою градієнтних методів. Метою цього процесу є по можливості точне наближення поведінки моделі до спостережуваної поведінки динамічної системи.

Принципово розрізняють пряму і інверсну модель об'єкту. На рис. Г.7 представлені схеми тренування цих нейронних моделей. Для спрощення на рис. Г.7 не зображені ланки затримки за часом, що формують попередні значення координат на вході нейронних мереж.

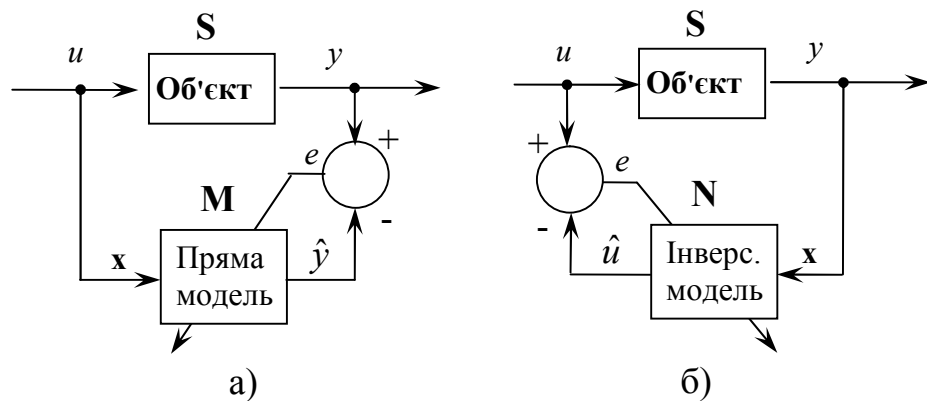


Рисунок Г.7 – Схеми тренування прямої (а) і інверсної (б) моделей

Вхідний сигнал x нейронній моделі об'єкту M на рис. Г.7,а ідентичний вхідному сигналу об'єкту S . Для інверсної нейромоделі N як вхідний сигнал використовується вихідний сигнал об'єкту S тоді як нейромодель повинна відновити вхідний сигнал об'єкту (див. рис. Г.7,б). Поки присутня помилка між виходом нейромоделі і її зразковим сигналом, ця помилка використовується для

тренування мережі. У обох випадках ідентифікації обчислюється градієнт функціонала якості (див. таблицю Г.1). Функціонал якості найчастіше при цьому квадратичний.

Таблиця Г.1 – Обчислення градієнта при тренуванні нейромоделей

	Пряма модель	Інверсна модель
Бажаний вихідний сигнал	y	u
Вихідний сигнал нейромоделі	$\hat{y} = \mathbf{M}(\mathbf{x}, \mathbf{w})$	$\hat{u} = \mathbf{N}(\mathbf{x}, \mathbf{w})$
Помилка навчання	$e = y - \hat{y}$	$e = u - \hat{u}$
Градієнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{y}}{\partial \mathbf{w}}$	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{u}}{\partial \mathbf{w}}$

Оскільки $\frac{\partial \hat{y}}{\partial \mathbf{w}}$ і $\frac{\partial \hat{u}}{\partial \mathbf{w}}$ обчислювані, то це дозволяє вести підстроювання вагових коефіцієнтів мережі згідно розглянутим градієнтним методам. Результатом цього буде мінімізація функціонала якості. Оскільки набір тренувальних даних однозначно відображає лише поведінку модельованого об'єкту $\mathbf{S}$ у робочому діапазоні, то після завершення тренування маємо $\mathbf{M} \approx \mathbf{S}$ і $\mathbf{N} \approx \mathbf{S}^{-1}$.

Типова послідовність дій, що приводить до формування нейромоделі (ідентифікації) динамічної системи, відображена в таблиці Г.2.

Таблиця Г.2 – Ідентифікація за допомогою нейронних мереж

Шаг	Дія
1	Знімання тренувальних даних
2	Вибір структури мережі
3	Вибір типу моделі
4	Тренування мережі
5	Тестування нейромоделі
6	При незадов. результаті тестування, перейти до п.3 або до п.2

Оскільки тренувальні дані є носієм інформації про систему, що ідентифікується, то наявності, кількості і якості цих даних слід приділити особливу увагу. Тренувальних дані повинні мати наступні характеристики: якісна дискретизація, рівномірність дисперсії і величини варіації, рівномірність амплітуд всіх сигналів і їх середньоквадратичних значень.

При отриманні позитивного результату апроксимації, слід також перевірити здібність нейромоделі до узагальнення, тобто її здатність повторювати істотні характеристики об'єкту. Для цього тесту необхідний ще один набір даних, що не брав участь при тренуванні мережі. Цей тест дає остаточний висновок про адекватність нейромоделі.

Представлені теоретичні положення по нейронній ідентифікації, а також характеристики прямонаправлених мереж є основою успішного регулювання нелінійних динамічних систем.

ДОДАТОК Г

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;
```

```
% Система управління електроприводом рольганга  
%(3-х роликів секція перед ножицями)  
% 2024 рік
```

```
% Вихідні дані
```

```
kd=0.28;  
In=282;  
Mn=In/kd;  
ktp=18;  
kn=0.109;  
kc=0.034;  
Tmc=5e-3 ;  
Ta=0.083;  
Te=0.061;  
Re=0.0664;  
Jd=3.9875;  
Jm=15.1125; %завантаженого  
Js=Jd+Jm;  
w0=6;  
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);  
c12=100.0;  
b12=10;  
b12=5;  
b12=0;  
Tme=2*Tmc+Ta;  
kic=Re/(2*Tmc*kc*ktp);  
kpc=kic*Te;  
kpe=Js*kd^2*kc/(2*Tme*kn);  
kie=kpe/(4*Tme);
```

```
Uz=24;
```

```
% Матриці
```

```
A=[-b12/Jm 1/Jm b12/Jm zeros(1,5);  
-c12 0 c12 zeros(1,5);  
b12/Jd -1/Jd -b12/Jd 1/Jd zeros(1,4);  
zeros(1,4) 1/kd 0 0 0;  
0 0 0 -kd/(2*Tmc^2) -1/Tmc kpe/(2*Tmc^2*kc) -kpe/(2*Tmc^2*kc)  
1/(2*Tmc^2*kc);
```

```

zeros(1,5) -1/Ta 0 0;
zeros(1,2) kn/(Ta*kd) zeros(1,3) -1/Ta 0
zeros(1,5) kie -kie 0];
B=[0 -1/Jm;
zeros(4,2);
1/Ta 0;
0 0
0 0];
C=zeros(5,8);
C(1,1)=1;C(2,2)=1;C(3,3)=1;C(4,4)=1;C(5,5)=1;C(6,8)=1;
D=zeros(6,2);

```