

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

НЕЙРОМЕРЕЖЕВА СИСТЕМА УПРАВЛІННЯ РОЛЬГАНГОМ
ЗА НОЖИЦЯМИ СТАНУ ГАРЯЧОЇ ПРОКАТКИ

(тема кваліфікаційної роботи)

Виконав: студент 2 курсу, групи ЗЕА-А24мг
спеціальності: 174 Автоматизація, комп'ютерно-
інтегровані технології та робототехніка

(код і найменування спеціальності)

_____ / Дмитро ДУДЧЕНКО
(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)

Харків – 2024

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) Дмитро ДУДЧЕНКО
(ім'я, прізвище)

1. Тема роботи Нейромережева система управління рольгангом за ножицями стану гарячої прокатки
затверджена наказом по університету № 4801-5/3664 від “06” жовтня 2025 р.
2. Термін здачі закінченої роботи “5” грудня 2025 р.
3. Вихідні дані до роботи Маса зливку $m_m=5,3-6,8 \cdot 10^3$ кг; діаметр бочки ролика $d_p=0,4$ м; діаметр шийки ролика $d_{ш}=0,2$ м; довжина бочки ролика $l_p=1$ м; кількість роликів секції $Z_p=3$; маса ролика $m_p=1 \cdot 10^3$ кг; окружна швидкість на бочці ролика $V_p=0,2-2$ м/с; передавальне число редуктора $i=5,75$; двигун постійного струму Д810; тиристорний електропривод КТЕУ 500/220; трансформатор ТСП-160/0,7У3. 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. Магістерська кваліфікаційна робота. Формальні ознаки. 2. Рольганг за ножицями. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейромережеві системи управління. Структурні схеми. 6. Управління з прогнозом. Математична модель. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14,15. Магістерська кваліфікаційна робота. Висновки

6. Консультант:

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

7. Дата видачі завдання “ 9 ” вересня 2025 р.

Керівник

(підпис)

Тетяна ВАСИЛЕЦЬ
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Дмитро ДУДЧЕНКО
(ім'я, прізвище)

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфікаційної роботи**

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про виконання завдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму рольганга за ножицями стана гарячої прокатки.	09.09.2025- 18.09.2025	
2	Розрахунок системи підлеглого регулювання координат електроприводу з оптимізацією контурів за модульним та симетричним критеріями. Моделювання системи на ЕОМ.	19.09.2025- 15.10.2025	
3	Дослідження системи з урахуванням пружних елементів. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ЕОМ.	16.10.2025- 05.11.2025	
4	Синтез нейромережової системи управління на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2025- 25.11.2025	
5	Оформлення роботи	26.11.2025- 05.12.2025	

Студент (ка)

(підпис)

Дмитро ДУДЧЕНКО
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 212 сторінок, 64 рисунків, 1 таблиці, 73 використаних джерела, 5 додатків.

Головною метою проведеного дослідження є створення та обґрунтування методів побудови системи керування електроприводом рольгангового механізму, розташованого за ножицями стана гарячої прокатки. Основний акцент зроблено на підвищенні якісних показників функціонування такої системи шляхом застосування нейромережевих регуляторів.

У роботі на основі класичних принципів теорії автоматичного керування виконано визначення параметрів необхідної системи керування, використовуючи аналітичні розрахунки. Для опису динаміки багатомасової електромеханічної структури сформовано математичні моделі, побудовані із застосуванням інструментарію математичного моделювання. Далі, спираючись на апарат штучних нейронних мереж, реалізовано синтез нейрорегулятора для керування двомасовою електромеханічною ланкою. Дослідження переходових процесів та оцінювання роботи нейромережевої системи здійснювалися методом комп'ютерного моделювання в середовищі MATLAB.

У результаті виконання магістерської кваліфікаційної роботи доведено наукову та практичну доцільність використання технологій нейронних мереж для регулювання роботи багатомасових електромеханічних приводів рольгангу, що працює за ножицями стана гарячої прокатки. Створено систему керування координатами електроприводу механізму, визначено необхідні параметри системи автоматичного керування та проведено її моделювання на ЕОМ.

Сформовано математичний опис системи з урахуванням пружних з'єднань та деформацій механічної частини приводу. Для двомасової моделі визначено її динамічні властивості та особливості реакції на керуючі впливи.

Розглянуто архітектуру нейромережевої системи керування, а також виконано синтез нейрорегулятора типу NN Predictive Controller у середовищі MATLAB. Проведено чисельне моделювання повної нейромережевої системи з розробленим регулятором. Встановлено, що спроектована система демонструє високі показники якості, швидкодію та стійкість до збурень.

Основні результати та теоретичні положення магістерської роботи впроваджені у навчальний процес кафедри АМЕТ ННІ «УПА» Харківського національного університету ім. В.Н. Каразіна та використовуються при викладанні лекцій, виконанні курсових проєктів і проведенні практичних занять.

Ключові положення та отримані наукові результати були представлені та обговорені на науково-практичній конференції студентів ННІ «УПА» (Харків, 2025 р.).

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

ABSTRACT

An explanatory message consists of: 212 pages, 64 figures, 1 table, 73 used sources, 5 additions.

The main goal of the research is to create and substantiate methods for building a control system for an electric drive of a roller table mechanism located behind the shears of a hot rolling mill. The main emphasis is on improving the quality indicators of the functioning of such a system by using neural network regulators.

In the work, based on the classical principles of the theory of automatic control, the parameters of the necessary control system were determined using analytical calculations. To describe the dynamics of a multi-mass electromechanical structure, mathematical models were formed, built using mathematical modeling tools. Then, relying on the apparatus of artificial neural networks, the synthesis of a neural regulator for controlling a two-mass electromechanical link was implemented. The study of transient processes and the evaluation of the operation of the neural network system were carried out by computer modeling in the MATLAB environment.

As a result of the master's qualification work, the scientific and practical feasibility of using neural network technologies to regulate the operation of multi-mass electromechanical drives of a roller table operating behind the shears of a hot rolling mill was proven. A system for controlling the coordinates of the electric drive of the mechanism has been created, the necessary parameters of the automatic control system have been determined and its simulation has been carried out on a computer.

A mathematical description of the system has been formed taking into account elastic connections and deformations of the mechanical part of the drive. For the two-mass model, its dynamic properties and features of the response to control influences have been determined.

The architecture of the neural network control system has been considered, and a neural regulator of the NN Predictive Controller type has been synthesized in the MATLAB environment. Numerical simulation of the complete neural network system with the developed regulator has been carried out. It has been established that the designed system demonstrates high quality indicators, speed and resistance to disturbances.

The main results and theoretical provisions of the master's thesis have been implemented in the educational process of the Department of AMET of the Research Institute of Mechanical Engineering "UIPA" of the V.N. Karazin Kharkiv National University and are used when teaching lectures, performing course projects and conducting practical classes.

The key provisions and obtained scientific results were presented and discussed at the scientific and practical conference of students of the National Research Institute "UIPA" (Kharkiv, 2025).

Keywords: ARTIFICIAL NEURO, 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;
 J – момент інерції;
 k_d – коефіцієнт посилення двигуна;
 T_e – електромагнітна постійна часу електроприводу;
 T_m – електромеханічна постійна часу електроприводу;
 $W(p)$ – передатна функція;
 c – коефіцієнт жорсткості;
 β – коефіцієнт внутрішнього в'язкого тертя;
 A – матриця стану;
 B – матриця управління;
 C – матриця виходу;
 $\vec{X}(t)$ – вектор стану системи;
 $\vec{U}(t)$ – вектор управління;
 $\vec{Y}(t)$ – вектор виходу;
САУ – система автоматичного управління;
М – двигун.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.	
ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	14
1.1 Перспективи розвитку штучних нейронних мереж	14
1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж	19
1.3 Нейрокерування динамічними об'єктами	25
1.4 Призначення та класифікація рольгангів	36
1.5 Технічні вимоги до електроприводів рольгангів	37
1.7 Постановка задачі синтезу нейромережевої системи управління	38
ВИСНОВКИ ДО РОЗДІЛУ 1	41
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ ЕЛЕКТРОПРИВОДОМ РОЛЬГАНГА СТАНА ГАРЯЧОЇ ПРОКАТКИ	43
2.1 Технічні дані механізму і силового електроустаткування. Вимоги до системи управління	43
2.2 Перевірка електродвигуна за нагрівом та перевантажувальною здатністю	45
2.3 Розрахунок параметрів силового кола електроприводу	50
2.4 Вибір структури системи автоматичного регулювання координат електроприводу	54
2.5 Синтез послідовного корегуючого пристрою контура струму	56
2.6 Синтез послідовного корегуючого пристрою контура ЕРС	59
2.7 Дослідження динамічних характеристик системи	65
ВИСНОВКИ ДО РОЗДІЛУ 2	71
РОЗДІЛ 3 ДОСЛІДЖЕННЯ РОБОТИ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ВЛАСТИВОСТЕЙ МЕХАНІЧНИХ ЕЛЕМЕНТІВ	73
3.1 Математична модель механічної частини електроприводу з урахуванням пружних зв'язків	73

3.2 Рівняння стану одномасової системи.....	75
3.3 Математичне моделювання двомасової системи	78
3.4 Розрахунок перехідних процесів двомасової системи.....	86
ВИСНОВКИ ДО РОЗДІЛУ 3	95
РОЗДІЛ 4. СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	96
4.1 Регулятори, реалізовані засобами пакета Neural Network Toolbox у середовищі MATLAB	96
4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller	98
4.3 Проектування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB	144
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	148
ВИСНОВКИ ДО РОЗДІЛУ 4	154
ВИСНОВКИ	156
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	158
ДОДАТОК А. БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ.....	165
А.1 Класифікація штучних нейронних мереж	165
А.2 Базовий процесорний елемент (штучний нейрон).....	167
А.3 Статичні багатошарові нейронні мережі	169
ДОДАТОК Б. ВЛАСТИВОСТІ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ.....	174
Б.1 Здібність до навчання і накопичення інформації.....	174
Б.2 Апроксимаційні властивості нейронних мереж	174
Б.3 Апроксимація і пам'ять в багатошарових нейронних мереж.....	177
ДОДАТОК В. НАВЧАННЯ СТАТИЧНИХ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ.....	181
В.1 Загальна характеристика алгоритмів навчання.....	181
В.2 Метод і алгоритм зворотного розповсюдження помилки.....	184
В.3 Перенавчання і узагальнення нейронних мереж.....	195

ДОДАТОК Г. УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ	
З ПРОГНОЗОМ	198
Г.1 Алгоритм нейроуправління з прогнозомі	198
Г.2 Структура нейронної мережі для моделювання об'єкту.....	205
Г.3 Прогноз з використанням нейронної мережі.....	209
Г.4 Похідні рівнянь нейронної мережі.....	210
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	212

ВСТУП

Актуальність теми. Рівень розвитку національної економіки у значній мірі залежить від масштабів та ефективності металургійного виробництва, оскільки саме метали формують матеріальну основу більшості галузей промисловості. Чавун, сталь і різні види прокату становлять основу продукції металургійних підприємств. Зокрема, чавун і сталь є базовими матеріалами для виготовлення верстатів, промислового та сільськогосподарського обладнання, а прокатна продукція – рейки, балки, труби, дріт, сортовий і листовий метал різних марок – широко застосовується у будівництві, транспорті, машинобудуванні та багатьох інших сферах господарської діяльності. Без стабільного функціонування металургійної промисловості неможливе повноцінне існування та розвиток сучасної держави, а досягнення високих темпів економічного зростання без потужної металургійної бази стає принципово недосяжним.

Ефективність роботи металургійних агрегатів, їх продуктивність та якість готового металу значною мірою визначаються рівнем технічної досконалості електроприводів. Тому питання вдосконалення силової частини електроприводів, алгоритмів керування та систем автоматизації мають першочергове значення. Сучасні вимоги до систем керування електроприводами передбачають високу точність, швидкодію, стійкість і адаптивність до змін у технологічному процесі. Зростання складності технологічних ліній, збільшення їх швидкостей та необхідність дотримання жорстких регламентів роботи роблять ручне управління практично неможливим. За таких умов оператор не здатен контролювати значну кількість взаємопов'язаних параметрів без застосування засобів обчислювальної техніки.

Розвиток промислової електроніки, комп'ютерних систем та цифрових технологій спричинив появу нових компонентів і методів побудови систем автоматичного керування, здатних відповідати зростаючим технічним вимогам сучасної промисловості. Одним із найбільш перспективних напрямів у цій галузі

зі вважається застосування адаптивних систем керування, поєднаних з методами теорії штучних нейронних мереж (ШНМ) [1–12].

Сфера ШНМ об'єднує науковців із різних напрямів – від комп'ютерних наук і математики до біології, психології та нейрофізіології. Це призводить до того, що штучні нейронні мережі не утворюють єдиної «монолітної» теорії, а становлять багатогранну міждисциплінарну область знань із власними підходами, методами та спеціалізованими напрямками. На сучасному етапі вже сформувалися такі галузі, як проектування нейронних мереж (яке охоплює питання їх архітектури та структури), нейроінформатика (зосереджена на дослідженні навчання мереж), нейрокомп'ютинг (орієнтований на паралельні обчислення), а також нейроморфна інженерія, що займається створенням апаратних засобів та прикладних рішень на основі нейромережових принципів.

Якщо розглядати ШНМ з погляду теорії інтелектуальних систем, їх можна характеризувати як окремий напрям, здатний накопичувати й узагальнювати інформацію, використовувати її для вирішення різних задач і генерувати нові знання під час експлуатації. Основною властивістю, яка відрізняє нейронні мережі від інших інтелектуальних систем, є їх здатність до навчання на множині прикладів, запам'ятовування і відтворення навіть частково спотворених даних, формування асоціативних зв'язків, а також їх висока стійкість і надійність у разі збоїв чи неповної інформації.

Розвиток ШНМ значною мірою базується на спробах зрозуміти принципи роботи мозку живих організмів, який істотно відрізняється від класичної комп'ютерної архітектури. Мозок – це нелінійна паралельна система, здатна до мислення, керування інформаційними потоками, нагромадження та відновлення інформації. Його основним елементом є нейрон, що отримує сигнали через дендрити, обробляє їх і передає далі. У штучних нейронних мережах цей принцип реалізується через взаємодію великої кількості простих обчислювальних елементів, що працюють паралельно та здатні навчатися, узагальнювати досвід і формувати рішення у зручній для інтерпретації формі.

На сьогодні неймережеві технології успішно застосовуються для розв'язання складних наукових і прикладних задач у техніці, фінансах, медицині, бізнесі та інших сферах, де потрібні інтелектуальні обчислення. Використання ШНМ у системах керування відкриває можливості для створення нових підходів до розв'язання задач, які раніше не могли бути виконані з необхідною точністю. Саме тому обрана тема дослідження є актуальною.

Мета і завдання дослідження. Метою роботи є розроблення методів синтезу системи керування електроприводом рольгангового механізму, що розташований за ножицями стана гарячої прокатки, з метою підвищення якісних характеристик функціонування на основі неймережевих регуляторів. Для досягнення поставленої мети необхідно розв'язати такі завдання:

1. Провести аналіз існуючих систем керування та оцінити можливості впровадження неймережевих технологій.
2. Розвинути математичну модель динаміки об'єкта керування.
3. Запропонувати структурну схему системи з неймережевим регулятором.
4. Виконати синтез неймережевої системи з прогнозуючим регулятором, яка забезпечуватиме високі показники якості.
5. Провести комп'ютерне моделювання роботи неймережевої системи та здійснити аналіз отриманих результатів.

Об'єкт і предмет дослідження.

Об'єктом дослідження є динамічні процеси в системі керування електроприводом рольганга перед ножицями стана гарячої прокатки з урахуванням пружних властивостей механічних ланок.

Предмет дослідження – моделі та методи синтезу неймережевої системи керування двомасовою електромеханічною системою рольгангового механізму перед ножицями стана гарячої прокатки.

Методи дослідження. Теоретичні положення роботи спираються на фундаментальні принципи теорії автоматичного керування. Для формування математичних моделей використано методи математичного та імітаційного моделю-

вання. Для синтезу нейромережевої моделі об'єкта та розроблення процедури її навчання залучено апарат теорії штучних нейронних мереж. Методи нейроуправління стали основою побудови нейромережевої системи керування, а моделювання на ЕОМ дало можливість оцінити ефективність розроблених рішень.

Наукова новизна отриманих результатів. Отримані у процесі виконання роботи результати містять такі науково нові положення:

1. Уперше обґрунтовано доцільність застосування нейромережевих технологій для керування багатомасовою електромеханічною системою рольгангового механізму перед ножицями стана гарячої прокатки.

2. Запропоновано та вперше синтезовано нейромережеву систему керування електроприводом рольганга з урахуванням пружних елементів, що забезпечує високі показники якості.

3. Уперше розроблено математичну модель системи керування електроприводом рольганга у формі багатошарового персептрона прямого поширення.

4. Удосконалено математичні моделі об'єктів керування зі складними кінематичними зв'язками, зокрема двомасових електромеханічних систем.

5. Синтез нейромережевих структур з прогнозуванням для керування складними динамічними об'єктами отримав подальший розвиток.

Практичне значення роботи. Практична цінність результатів полягає у створенні нейромережевої системи керування двомасовою електромеханічною системою рольганга перед ножицями стана гарячої прокатки, що забезпечує точне та надійне регулювання режимів його роботи.

Результати дослідження вже використовуються в освітньому процесі кафедри АМЕТ ННІ «УПА» ХНУ імені В.Н. Каразіна під час проведення лекційних занять, виконання курсових проєктів та практичних робіт.

Особистий внесок здобувача

Усі результати, винесені на захист, отримані автором самостійно. Це, зокрема, синтез системи регулювання координат електроприводу, розроблення математичної моделі з урахуванням пружних елементів, створення структурної схеми нейромережевої системи, синтез нейрорегулятора NN Predictive

Controller у середовищі MATLAB (Neural Network Toolbox), а також моделювання роботи системи та аналіз отриманих результатів. У роботі містяться посилання на авторів і джерела при використанні відомих наукових положень і даних.

Апробація результатів роботи. Основні результати дослідження були представлені та обговорені на студентській науковій конференції ННІ «УПА» ХНУ імені В.Н. Каразіна (Харків, 2024 р.).

Структура та обсяг роботи. Магістерська робота складається зі вступу, чотирьох розділів основної частини, висновків, додатків і списку використаних джерел. Загальний обсяг становить 212 сторінок, включаючи 64 рисунки (38 у тексті та 26 на 24 окремих сторінках), 1 таблицю, 5 додатків загальним обсягом 48 сторінок та список із 73 джерел на 7 сторінках.

РОЗДІЛ 1

АНАЛІЗ СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Перспективи розвитку штучних нейронних мереж

1.1.1 Загальні тенденції розвитку ШНМ

Штучні нейронні мережі (ШНМ) сьогодні залишаються однією з найважливіших та найдинамічніших технологій у галузі штучного інтелекту. Їх еволюція визначається безперервним пошуком шляхів підвищення гнучкості, швидкодії та здатності до самонавчання у складних і швидкозмінних середовищах. Сучасні здобутки апаратної інфраструктури, математичних методів і алгоритмічних підходів забезпечують умови для активного впровадження ШНМ у найрізноманітніші сфери: від аналітики великих даних і розпізнавання образів до управління складними технічними, біологічними, промисловими та автономними системами [13].

Однією з ключових тенденцій є перехід до фундаментальних (foundation) моделей – масштабних нейромережевих систем, здатних навчатися на величезних, універсальних наборах даних і надалі адаптуватися до різноманітних прикладних задач без потреби в повному перенавчанні [13]. Такі моделі демонструють виняткову здатність до генералізації: вони можуть опрацьовувати та узгоджувати інформацію з різних доменів – текстових, візуальних, сенсорних, часових. Найяскравішим прикладом є архітектури на основі трансформерів, які стали стандартом для задач обробки природної мови, комп'ютерного зору, аналізу послідовностей і навіть інтелектуального керування складними технічними процесами [14].

Ще одним вагомим напрямом розвитку є оптимізація обчислювальної та енергетичної ефективності ШНМ. Сучасні нейронні моделі, особливо великі мовні чи мультимодальні системи, потребують колосальних ресурсів: їх навчання може тривати тижнями та вимагати сотень тисяч GPU, а енергоспожи-

вання сягає мегаватів. У такій ситуації особливої актуальності набувають методи оптимізації нейронних мереж – квантування, структурне зрідження (pruning), дистиляція знань, факторизація параметрів тощо [15]. Ці технології дозволяють будувати компактні, але достатньо точні моделі, придатні для використання у мобільних пристроях, IoT-системах або автономних роботизованих комплексах.

Суттєвий інтерес дослідників привертають нейроморфні обчислення – апаратні системи, побудовані за принципами роботи біологічного мозку. Вони використовують спайкові моделі нейронів і характеризуються надзвичайно низьким енергоспоживанням у поєднанні з можливістю паралельної обробки сигналів. Одним із найвідоміших проєктів останніх років є «Darwin Monkey» – мозкоподібний суперкомп’ютер, представлений у 2025 році, який здатний виконувати мільярди паралельних операцій при мінімальних витратах енергії [16]. Подібні системи відкривають перспективи побудови інтелектуальних «периферійних» обчислень, здатних до автономного прийняття рішень без звернення до хмарних ресурсів.

Важливою проблемою залишається інтерпретованість і прозорість роботи нейронних мереж. Глибокі моделі з мільйонами параметрів часто працюють як «чорні скриньки»: користувач не може зрозуміти, які внутрішні залежності сформувала система та чому вона приймає конкретне рішення. У відповідь на це активно розвиваються методи Explainable AI (XAI) – інструменти, що дозволяють виявляти ключові ознаки вхідних даних, візуалізувати вплив ваг, аналізувати внутрішні шари мережі та надавати користувачеві зрозуміле пояснення результату [17]. Це критично важливо для областей, де довіра та прозорість рішень є обов’язковими: медицина, енергетика, авіація, фінанси, транспорт.

Серед новітніх тенденцій також варто відзначити зростаючий інтерес до гібридних обчислювальних систем, що інтегрують ШНМ з методами нечіткої логіки, генетичними алгоритмами, фізичними моделями та квантовими обчислювальними технологіями. Така інтеграція дозволяє створювати більш стійкі, адаптивні й ефективні моделі, здатні давати точні прогнози навіть за умов не-

повноти даних або високої невизначеності. Особливо перспективним напрямом вважається використання квантових алгоритмів для оптимізації вагових коефіцієнтів нейронних мереж, що потенційно може радикально прискорити процес навчання.

Значного розвитку набуває й етичний вимір впровадження ШНМ. Масове використання штучного інтелекту порушує питання безпеки даних, недискримінаційності алгоритмів, запобігання помилковим рішенням та відповідальності розробників. Європейські регуляторні органи та міжнародні стандарти впроваджують комплексні правила, спрямовані на створення безпечних та відповідальних AI-систем [18].

Не менш важливою тенденцією є мультидисциплінарна інтеграція ШНМ. Нейронні мережі стають універсальним інструментом, який ефективно працює в науці про матеріали, біоінформатиці, кліматології, аграрній інженерії та енергетиці. Використання *physics-informed neural networks* дозволяє враховувати фізичні закони у процесі навчання, що суттєво підвищує достовірність результатів [19].

1.1.2 Перспективи застосування нейронних мереж у системах управління

Окремо необхідно відзначити стрімкий розвиток напрямку застосування нейронних мереж у задачах автоматичного, адаптивного та інтелектуального управління. Однією з найбільш перспективних тенденцій є створення гібридних систем управління, що інтегрують нейронні мережі з традиційними методами контролю, такими як ПІД-регулятори або моделі предиктивного управління (Model Predictive Control, MPC). Така інтеграція дозволяє поєднати адаптивність і здатність ШНМ моделювати складні нелінійні залежності з формальною стійкістю класичних підходів [20].

У традиційних системах управління значну роль відіграє точність математичної моделі об'єкта. Проте більшість технічних систем мають нелінійний або змінний характер, їх поведінка залежить від шумів, стохастичних факторів і зовнішніх впливів. Це ускладнює побудову адекватних моделей. Нейронні ме-

режі здатні ефективно апроксимувати невідомі залежності, адаптуватися до зміни параметрів у часі та враховувати вплив завад, що робить їх природним інструментом для складних задач управління [21].

Швидко розвивається напрям нейромережевого предиктивного управління (NMPC), у якому нейронна мережа виступає або як внутрішня модель об'єкта, або як самостійний генератор керуючої дії. Це дає змогу суттєво зменшити час налаштування регулятора, підвищити точність та забезпечити стабільну роботу системи навіть у випадку невизначеностей параметрів чи різких змін зовнішніх умов [14; 22].

Перспективні сфери застосування нейромережевих методів управління включають:

- автоматизацію промислових процесів, де ШНМ здатні прогнозувати майбутні значення технологічних параметрів у реальному часі;
- керування автономним транспортом, включно з дронами, роботизованими платформами та безпілотними автомобілями;
- енергетику, зокрема оптимізацію розподілу навантажень, прогнозування генерації ВДЕ, управління мікромережами;
- робототехніку, де нейронні мережі формують стратегії маніпуляції, балансування, навігації та взаємодії з довкіллям;
- біомедичні та медичні системи, наприклад адаптивне дозування ліків або керування апаратами життєзабезпечення.

Важливим і швидко зростаючим напрямом є поєднання нейронних мереж з підкріплювальним навчанням (Reinforcement Learning, RL). На основі RL будуються повністю автономні системи, які здатні самостійно формувати оптимальні стратегії управління, досліджуючи середовище та отримуючи зворотний зв'язок у вигляді винагороди [23]. Такий підхід відкриває можливості для створення роботів та технічних систем, здатних до тривалого самонавчання й адаптації без участі оператора.

Узагальнюючи, перспективи розвитку ШНМ у задачах управління визначаються такими ключовими факторами:

1. Масштабування моделей та появою універсальних архітектур, здатних виконувати різні функції в межах однієї системи.
2. Зростанням апаратної та енергетичної ефективності через впровадження оптимізаційних методів і спеціалізованого обладнання.
3. Підвищенням надійності, прозорості й етичності сучасних інтелектуальних систем.
4. Інтеграцією ШНМ із сучасними технологіями, такими як квантові обчислення, біоінформатика, робототехніка, кіберфізичні системи.

Таким чином, розвиток штучних нейронних мереж формує основу для створення нового покоління інтелектуальних систем управління – гнучких, адаптивних, автономних і безпечних, здатних ефективно діяти в умовах складних багатокомпонентних технічних середовищ.

У додатках А–Г наведено базові поняття теорії нейронних мереж, що становлять необхідний фундамент для розуміння виконаних у роботі досліджень.

У даній роботі для синтезу нейромережевої системи використано багатосарові прямонаправлені мережі (MLP), які складаються з нейронів, з'єднаних тільки в прямому напрямку, без рекурентних зв'язків. У додатку А подано детальний розгляд структури штучного нейрона, у додатку Б – будову багатосарових мереж.

Огляд методів навчання та вибір конкретного методу, застосованого в роботі, наведено в додатку В.

У додатку Г розглянуто ефективну реалізацію узагальненого прогнозного управління (generalized predictive control) на основі багатосарової прямонаправленої мережі, яка виконує роль нелінійної моделі об'єкта управління. Саме цей алгоритм став основою для побудови й функціонування нейрорегулятора NN Predictive Controller, застосованого у магістерській роботі для синтезу нейромережевої системи управління.

1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж

Одним із ключових етапів розв'язання задач керування, прогнозування поведінки та моделювання нелінійних динамічних систем є побудова їх адекватних математичних моделей. Саме математична модель визначає можливість оптимального управління об'єктом, забезпечує формування прогнозів його майбутньої поведінки та дозволяє аналізувати систему у широкому діапазоні режимів роботи, включно з режимами, які неможливо або небезпечно відтворити експериментально.

Традиційно процес моделювання ґрунтується на теоретичному та експериментальному аналізі властивостей досліджуваного об'єкта.

Теоретичний аналіз базується на вивченні фізичних, хімічних, біологічних або інших процесів, що відбуваються в системі. Результатом є побудова аналітичної моделі у вигляді диференціальних або різницевих рівнянь. Такий підхід дозволяє отримати модель, яка має чітке фізичне обґрунтування, однак потребує глибоких знань про структуру об'єкта, його параметри та внутрішні механізми.

Експериментальний аналіз виходить із реальних вимірювань вхідних та вихідних сигналів системи. На основі цих даних формується або параметрична, або непараметрична модель об'єкта. Найбільшого поширення набули саме параметричні моделі, які описуються компактним набором параметрів і потребують розв'язання задач структурної та параметричної ідентифікації. Проте побудова універсальної теорії ідентифікації нелінійних систем залишається складною через різноманіття типів нелінійностей, стохастичних впливів і невизначеностей.

У класичному підході апроксимація нелінійних операторів виконується за допомогою таких математичних моделей, як ряди Вольтерра, моделі Гаммерштейна, Вінера, поліноми Колмогорова–Габора та ін. Хоча ці інструменти мають усталену теоретичну базу, сфера їх застосування обмежена, а точність може суттєво знижуватися при складних або змінних нелінійностях.

Додатковою складністю є наявність вимірювальних завад, які спотворюють сигнали й потребують застосування попередньої фільтрації. У результаті побудова адекватної моделі в умовах шуму та неповної інформації часто вимагає використання гнучкіших інструментів.

Саме тому значного поширення набув підхід, заснований на застосуванні штучних нейронних мереж, які здатні апроксимувати довільні нелінійні функції з високою точністю та не потребують попереднього аналітичного опису об'єкта.

Ідентифікація за NARMAX-моделлю.

Розглянемо задачу ідентифікації нелінійного динамічного об'єкта, який описується загальною NARMAX-моделлю (Nonlinear AutoRegressive Moving Average with eXogenous inputs):

$$\tilde{y}(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n), k] + \xi(k), \quad (1.1)$$

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

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

$f(\cdot)$ – невідома нелінійна функція, яка відображає внутрішню динаміку об'єкта;

ξ – завада вимірювання (шум).

Для зручності введемо вектор узагальненого вхідного сигналу розмірністю $(n+m) \times 1$:

$$x(k) = [\tilde{y}(k-1), \tilde{y}(k-2), \dots, \tilde{y}(k-m), u(k-1), u(k-2), \dots, u(k-n)]. \quad (1.2)$$

Тоді модель (1.1) може бути записана у компактній формі:

$$\tilde{y}(k) = f[x(k), k] + \xi(k). \quad (1.3)$$

Завдання ідентифікації полягає у відновленні невідомої функції $f(\cdot)$ на основі спостережуваних вхідних $u(k)$ та вихідних $\tilde{y}(k)$ сигналів

Схему ідентифікації нелінійного об'єкта за допомогою нейронної мережі наведено на рис. 1.1. Мережа виконує роль апроксиматора функції $f(\cdot)$, тобто навчається відображати внутрішню динаміку об'єкта.

Нейронні мережі для ідентифікації.

У сучасних дослідженнях найбільш широке застосування для задач ідентифікації нелінійних динамічних об'єктів отримали два класи моделей:

1. Багатошаровий перцептрон (БП) (рис. 1.2) – універсальна апроксимуюча структура, яка відомо здатна наближати довільні неперервні функції.

2. Радіально-базисні мережі (РБМ) (рис. 1.3) – моделі, що використовують локальні базисні функції та забезпечують швидке збіжне навчання.

Обидва підходи розглядають нейронну мережу як інструмент апроксимації нелінійного оператора $f(\cdot)$.

Модель багатошарового перцептрона.

Рівняння БП має вигляд:

$$\hat{y}(k) = f^q \left[(\omega^q)^T f^{q-1} \left[(\omega^{q-1})^T f^{q-2} \left[\dots f^1 \left[(\omega^1)^T \right] x(k) \right] \right] \right], \quad (1.4)$$

де q – кількість шарів мережі;

ω^i – матриці вагових коефіцієнтів i -го шару;

$f(\cdot)$ – активаційні функції i -го шару.

Багатошаровий перцептрон добре підходить для моделювання складних нелінійних залежностей, але зазвичай потребує достатнього часу на навчання та велику кількість навчальних даних.

Модель радіально-базисної мережі.

РБМ апроксимує функцію $f(\cdot)$ за допомогою набору локальних базисних функцій $\Phi_i(x)$:

$$\hat{y}(k) = \sum_{i=1}^N c_i \Phi_i(x), \quad (1.5)$$

де c_i – вагові коефіцієнти.

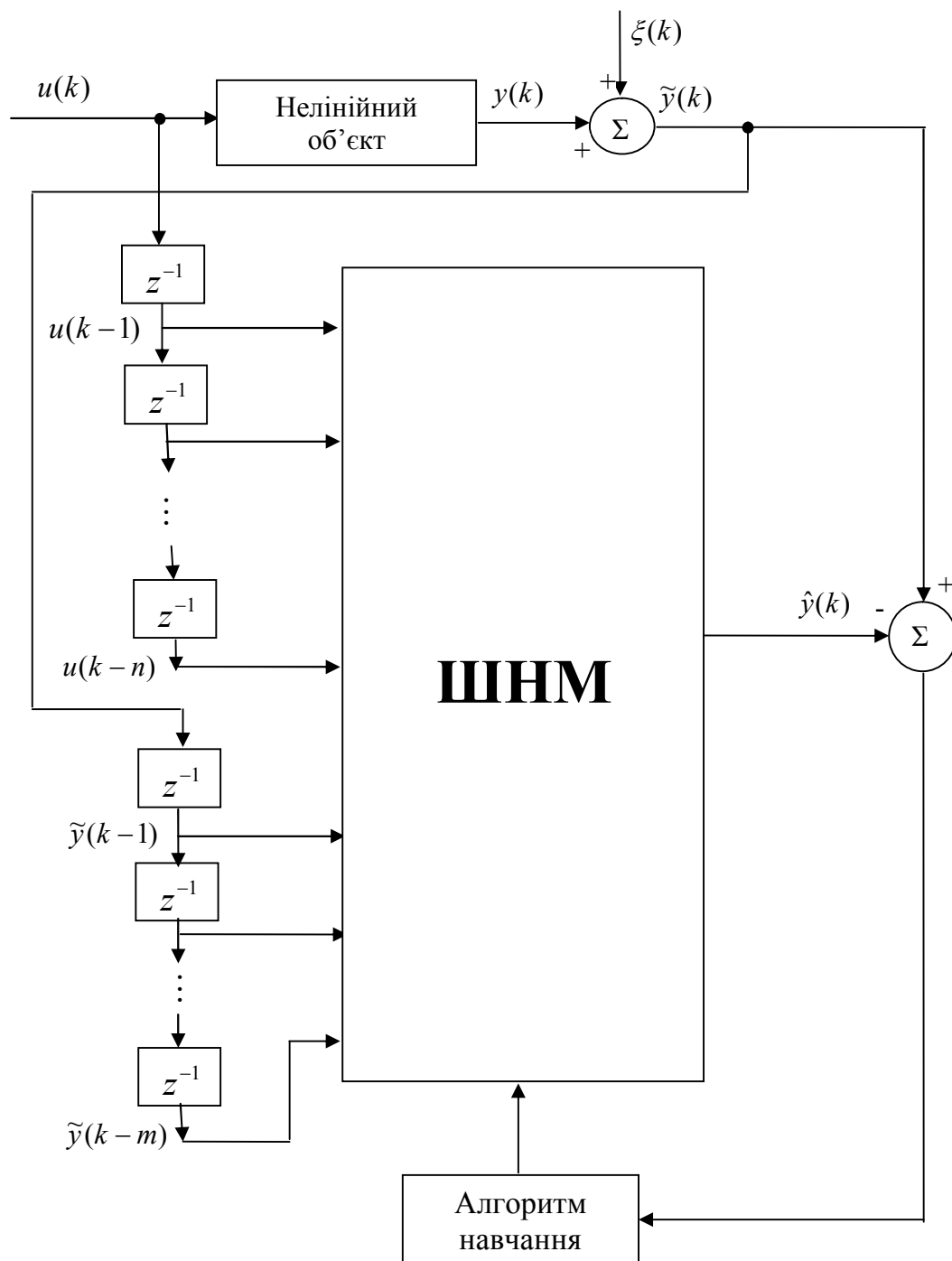


Рисунок 1.1 – Схема ідентифікації динамічного об'єкта
за допомогою нейронної мережі

Найчастіше як базисну використовують гаусівську функцію

$$\Phi_i(x) = \exp \left\{ -\frac{\|x - \mu_i\|^2}{\sigma_i^2} \right\}, \quad (1.6)$$

де μ_i, σ_i – центри і радіуси базисних функцій;

$\|\cdot\|$ – евклідова норма.

У цьому випадку параметрами, що підлягають визначенню, є:

$$\omega(k) = (c_0(k), c_1(k), \mu_1^T(k), \sigma_1(k), \dots, c_N(k), \mu_N^T(k), \sigma_N(k))^T.$$

У практичних застосуваннях часто фіксують $\Phi_0(x) = 1$, що спрощує процедуру навчання.

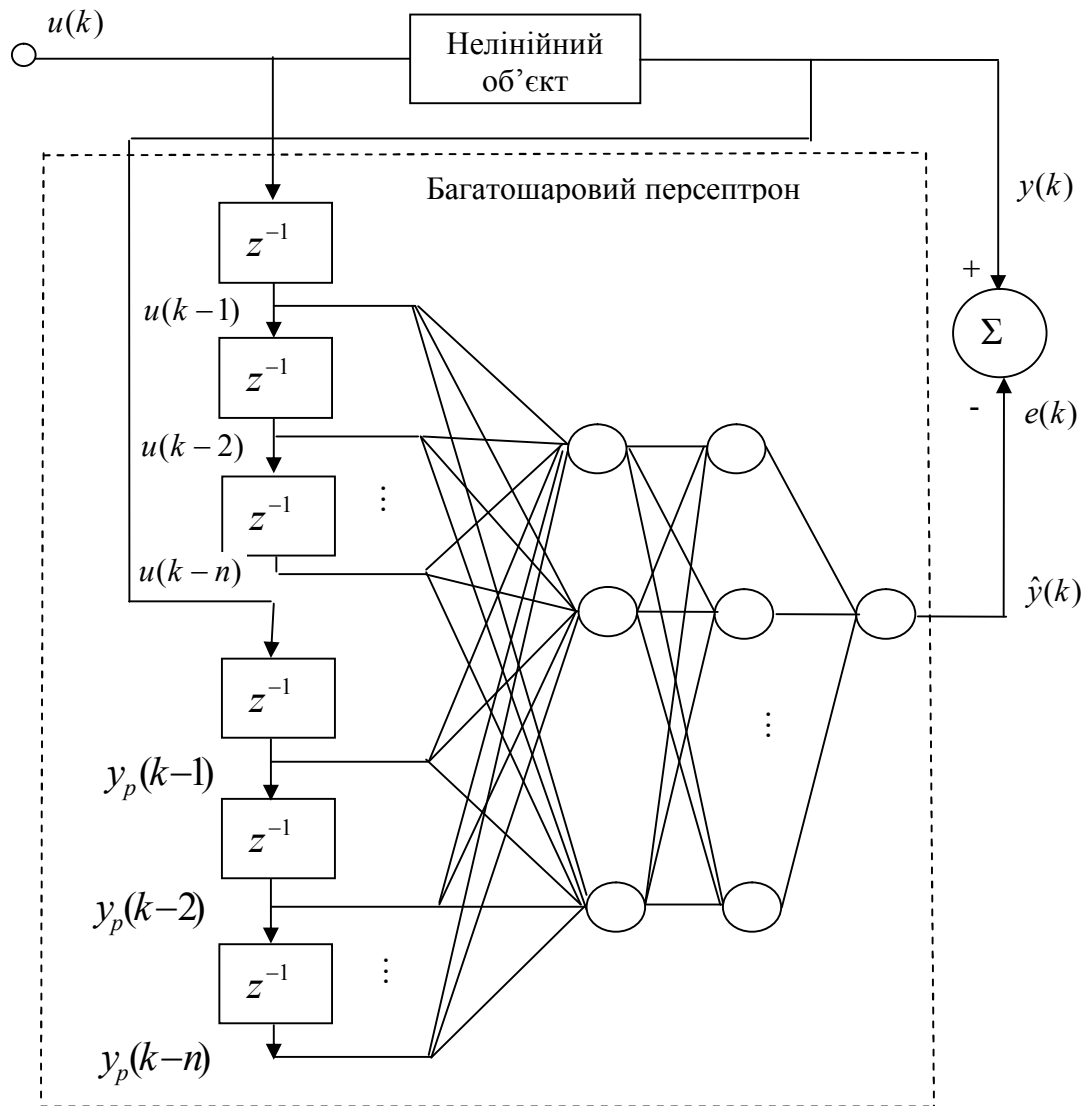


Рис 1.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового персептрона

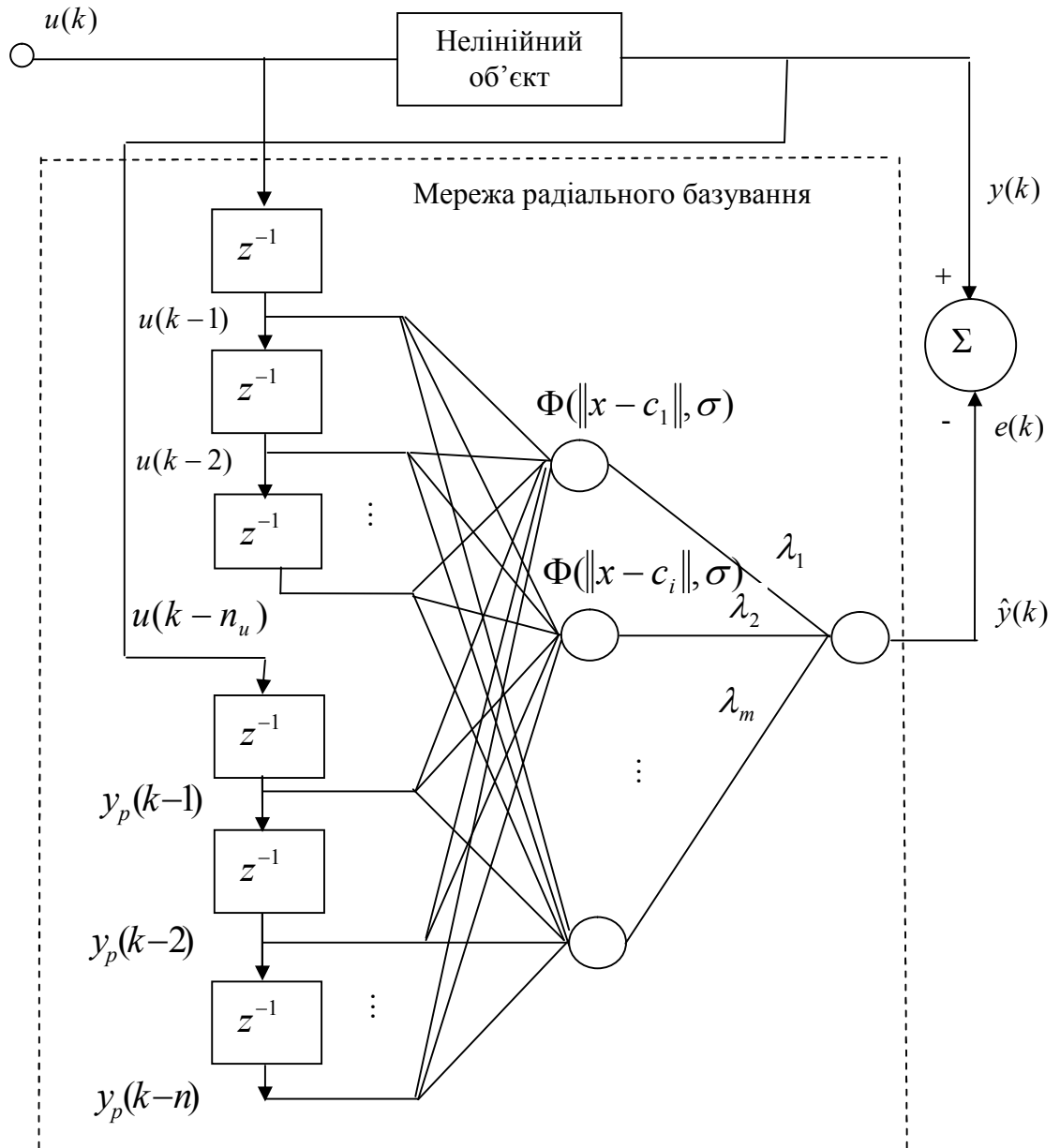


Рисунок 1.3 – Ідентифікація динамічного об'єкта за допомогою радіально-базисної мережі (РБМ)

Постановка задачі навчання.

Ідентифікація об'єкта полягає у налаштуванні параметрів нейронної мережі таким чином, щоб її вихід якомога точніше відповідав реальному вихідному сигналу.

Метою є мінімізація функціоналу помилки:

$$J = e^2(k) = M \left\{ [\tilde{y}(k) - \hat{y}(k)]^2 \right\} . \quad (1.7)$$

Навчання мережі виконується за допомогою одного з алгоритмів оптимізації (градієнтний спуск, метод Левенберга–Марквардта, стохастичні методи, еволюційні підходи тощо), залежно від структури моделі та властивостей динамічного об'єкта.

Застосування нейронних мереж у задачах ідентифікації нелінійних динамічних систем дає можливість:

- моделювати об'єкти, для яких неможливо або складно отримати аналітичний опис;
- враховувати нелінійності довільного типу;
- підвищувати точність і стійкість моделі в умовах шуму;
- будувати адаптивні моделі, що можуть змінюватися разом із динамікою об'єкта.

Таким чином, нейронні мережі виступають перспективним і ефективним інструментом для розв'язання задач ідентифікації, особливо у випадках складних технічних систем і процесів, де класичні методи моделювання є недостатніми.

1.3 Нейрокерування динамічними об'єктами

Процес побудови ефективних систем керування реальними технічними об'єктами ґрунтується на використанні їх математичних моделей. Сам вибір моделі, її точність, структурна складність і характер нелінійностей визначають тип і складність алгоритмів керування. У практиці автоматизації нерідко застосовують спрощення, зокрема заміну нелінійної моделі об'єкта її лінеаризованим аналогом у робочій точці. Це дозволяє використати відомі лінійні методи синтезу регуляторів. Однак таке наближення забезпечує прийнятну якість керування лише поблизу вибраного режиму. Для того щоб система могла ефективно працювати в ширшому діапазоні станів, параметри регулятора необхідно корегувати відповідно до зміни умов функціонування.

У цьому контексті актуальною є розробка адаптивних систем керування, що поєднують методи класичної теорії адаптації з апаратом штучних нейрон-

них мереж (ШНМ). Такий підхід надає можливість створювати компактні моделі складних нелінійних об'єктів та одночасно адаптивно налаштовувати їх параметри разом із параметрами регулятора при зміні властивостей об'єкта. Узагальнена структура подібної нейромережевої системи адаптивного керування подана на рис. 1.4.

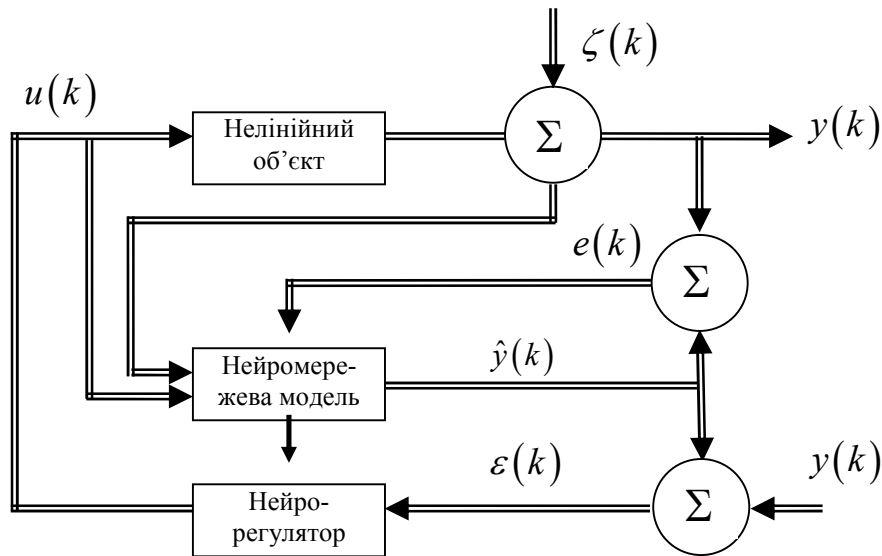


Рисунок 1.4 – Структура нейромережевої адаптивної системи управління

Для розв'язання задач нейрокерування найчастіше застосовують багатопшарові персептрони та радіально-базисні мережі, які завдяки своїм апроксимаційним властивостям здатні відтворювати складні нелінійні залежності.

Розглянемо загальний випадок багатовимірного нелінійного динамічного об'єкта, роботу якого описує рівняння

$$y(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)] + \xi(k), \quad (1.8)$$

де $y = (y_1(i), \dots, y_{N_y}(i))^T$; $u(i) = (u_1(i), \dots, u_{N_u}(i))^T$ – відповідно вектори вихідних та керуючих сигналів розмірностей $N_y \times 1$ і $N_u \times 1$;

$f(\cdot) = (f_1(\cdot), f_2(\cdot), \dots, f_{N_y}(\cdot))^T$ – невідома нелінійна вектор-функція об'єкта;

$\zeta(k) = [\zeta_1(k), \dots, \zeta_{N_y}(k)]^T$ – вектор зовнішніх або вимірювальних завад.

Введемо вектор узагальненого сигналу, який подається на вхід моделі:

$$x(k) = [y^T(k-1), \dots, y^T(k-m), u^T(k-1), \dots, u^T(k-n)]^T, \quad (1.9)$$

і перетворимо рівняння (1.8) до вигляду

$$y(k) = f(x(k) + \zeta(k)). \quad (1.10)$$

Процедура навчання нейронної мережі в задачі ідентифікації зводиться до корекції її параметрів таким чином, щоб мінімізувати різницю між фактичним вихідним сигналом об'єкта $y(k)$ та виходом моделі $\hat{y}(k)$. Це досягається через мінімізацію квадратичного функціоналу помилки (1.7), що уже був визначений у попередньому пункті.

Після завершення етапу ідентифікації навчена нейромережа використовується як частина контуру керування, забезпечуючи прогноз наступного стану об'єкта або реалізуючи сам регулятор.

Завдання синтезу керування полягає у визначенні таких значень керуючого впливу $u(k)$, які забезпечують мінімізацію функціоналу якості

$$J(k) = \|\varepsilon(k)\|^2 = \frac{1}{2} \|y^*(k) - \hat{y}(k)\|^2, \quad (1.11)$$

де $y^*(k)$ – бажані (задані) значення вектора вихідних сигналів.

Як і у випадку задачі ідентифікації, для пошуку мінімуму функціонала (1.11) застосовують різні рекурентні процедури оптимізації. Одним із найпоширеніших підходів є використання градієнтних методів:

$$u(k) = u(k-1) + \gamma(k)(\nabla_u \varepsilon(k)), \quad (1.12)$$

де $\nabla_u \varepsilon(k) = \frac{\partial \varepsilon(k)}{\partial u(k)}$; $\gamma(k) > 0$.

1.4 Огляд структур нейромережевих систем управління. Вибір структури системи

Історично процес створення систем управління (СУ) формується як послідовність взаємопов'язаних етапів, кожен із яких відіграє суттєву роль у досягненні бажаних показників роботи майбутньої системи. До основних етапів можна віднести:

- формування концептуальної ідеї побудови СУ та визначення її принципів;
- побудову моделі системи відповідно до сформованої концепції;
- аналіз поведінки отриманої моделі, оцінка її адекватності та ефективності;
- корегування або модернізацію первинної концепції на основі результатів моделювання та аналізу.

Протягом усіх цих стадій тривають теоретичні дослідження, спрямовані на вдосконалення базових положень створеної системи та адаптацію її компонентів до ширшого кола задач. Аналогічна еволюція характерна і для систем керування, що використовують штучні нейронні мережі (ШНМ). Варто підкреслити, що практичні рішення – як апаратні, так і програмні – у сфері нейромережевих систем керування нерідко випереджають глибоке теоретичне осмислення процесів, які в них реалізуються.

Сьогодні запропоновано та апробовано значну кількість структур нейромережевих регуляторів. Частина з них ґрунтується на структурованих методах управління, у яких використовується аналітичний закон регулювання та нейро-модель об'єкта. У таких системах нейронна мережа слугує для відновлення окремих елементів матриць моделі об'єкта. Однак максимальний потенціал нейронних мереж як універсальних апроксиматорів реалізується у неструктурованих методах керування, де регулятор синтезується без попереднього задання аналітичної структури моделі.

У цьому розділі коротко розглянуто найбільш поширені та актуальні сьогодні структури нейромережевих систем регулювання.

Однією з базових та широко досліджених моделей СУ є система зі зворотним зв'язком, в якій параметри регулятора змінюються в реальному часі. Прикладом такої структури є самоналагоджувальний контролер Астрома [24]. У подібних регуляторах коефіцієнти автоматично корегуються на кожному циклі керування відповідно до оцінки параметрів об'єкта. Узагальнена блок-схема такої системи наведена на рис. 1.5.

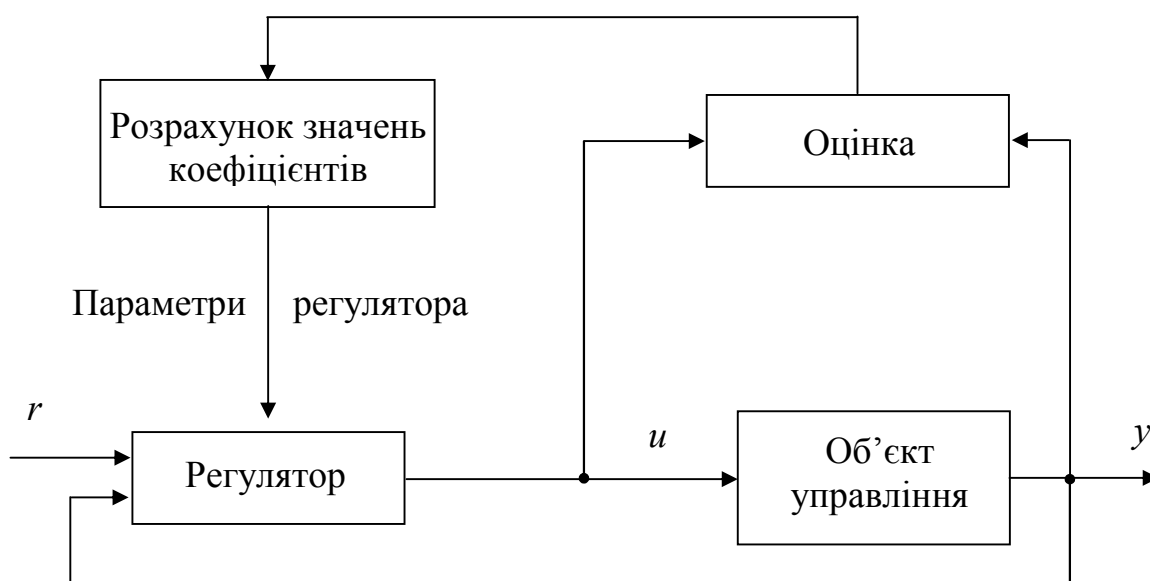


Рисунок 1.5 – Блок-схема управління зі зворотним зв'язком і коефіцієнтами, що налаштовуються в реальному часі

Іншим добре відомим підходом є застосування еталонної моделі Ляпунова. Адаптивні системи керування цього типу синтезують так, щоб вихід реального об'єкта поступово збігався з виходом заздалегідь визначеної моделі, яка описує бажані властивості системи [25]. Головною вимогою є асимптотична стійкість: з часом помилка між виходом об'єкта та еталонної моделі прямує до нуля. Крім того, на перехідних режимах забезпечуються гарантовані обмеження на величину відхилень. Структура такої системи наведена на рис. 1.6.

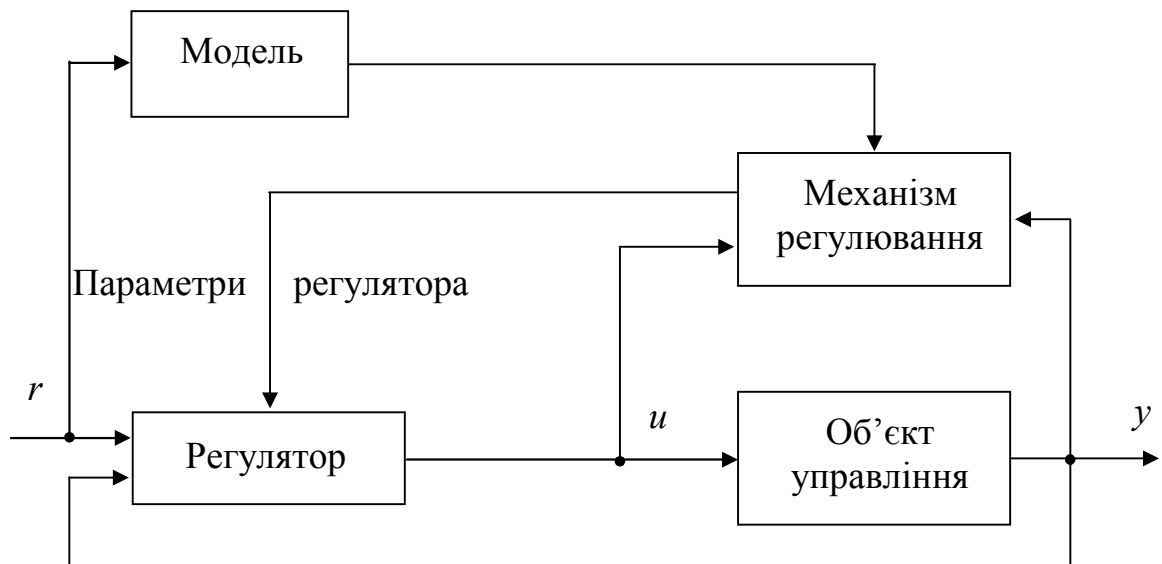


Рисунок 1.6 – Блок-схема адаптивного управління з еталонною моделлю

Значного розвитку набули й системи керування, побудовані із застосуванням штучних нейронних мереж. Вони розглядаються як перспективна альтернатива класичним підходам завдяки властивостям апроксимації складних нелінійних залежностей. Відомо, що нейромережа з двома шарами та достатньою кількістю нейронів у прихованому шарі здатна апроксимувати будь-яку неперервну функцію з довільною точністю. Це твердження базується на формулюванні теореми Вейєрштраса та докладно описано в [26]. Саме тому навіть відносно проста за структурою ШНМ із одним прихованим шаром може бути ефективним інструментом як для ідентифікації динамічних систем, так і для побудови регуляторів.

Одним із перших методів побудови нейромережових систем управління був підхід, заснований на «копіюванні» (імітації) вже існуючого контролера. Застосувавши цей метод у 1964 році, Уїдроу запропонував створювати експертні системи шляхом передавання знань від наявного експерта. Архітектура такої системи управління показана на рис. 1.7.

Аналізуючи цю архітектуру, можна поставити запитання щодо доцільності використання нейромережового контролера, якщо вже існує працездатний регулятор. Проте існує декілька вагомих аргументів:

- по-перше, наявний контролер може бути незручним для безпосереднього використання (зокрема, якщо роль контролера виконує оператор-людина);
- по-друге, нейронна мережа дозволяє застосувати іншу форму подання інформації про стан об'єкта управління, що може бути суттєво зручнішою для вимірювання, формалізації та подальшої обробки.

На сьогодні існує велика кількість різноманітних архітектур нейромережевих систем управління, які добре досліджені та широко застосовуються. У всіх випадках нейромережевий контролер призначений для формування управляючого сигналу, що забезпечує переведення об'єкта зі start-стану в бажаний кінцевий стан за оптимальною траєкторією.

Організація контролю та спосіб реалізації нейромережевого регулятора значною мірою визначаються вибраним алгоритмом навчання та структурою системи. Найчастіше використовують дві базові схеми:

1. Схему прямого (безпосереднього) управління.
2. Схему непрямого управління.

Типовим методом навчання у таких системах є алгоритм зворотного розповсюдження помилки.

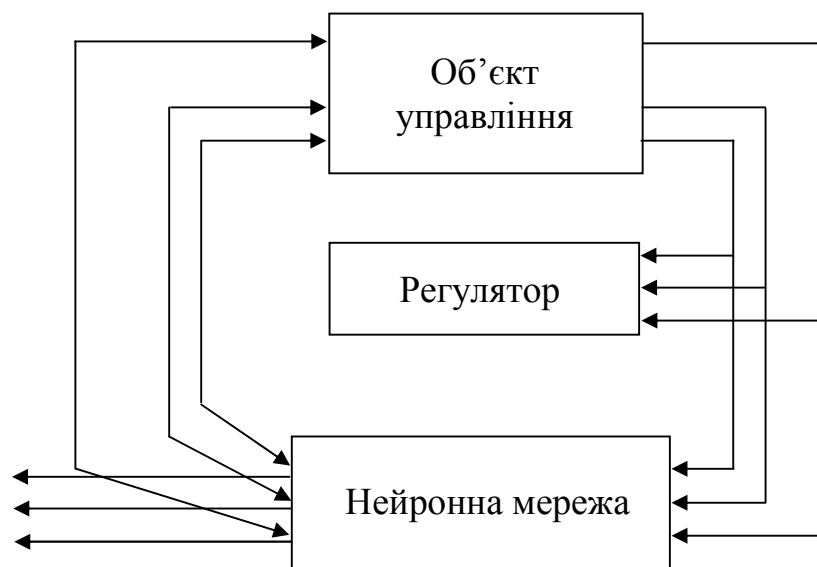


Рисунок 1.7 – Нейромережева СУ, заснована на «копіюванні» існуючого контролера

Непряме управління.

У схемі непрямого управління (рис. 1.8) на кожному кроці часу виконується ідентифікація параметрів моделі об'єкта управління. Отримані оцінки параметрів використовуються для корекції параметрів контролера. Таким чином, процес ідентифікації є явно вираженим та невіддільним від управління.

Недоліком цієї схеми є те, що як ідентифікація, так і формування управляючого сигналу базуються виключно на помилці моделі e_u , тобто на різниці між реальним та оціненим виходами. У такому випадку зменшення вихідної помилки системи e_y гарантувати неможливо.

Пряме управління.

У схемі прямого управління (рис. 1.9) нейромережевий контролер налаштовується безпосередньо на зменшення помилки виходу об'єкта e_y . Тобто мережа вчиться так змінювати управляючий сигнал, щоб миттєво мінімізувати відхилення виходу від заданого значення.

Цільова функція.

У більшості нейромережевих систем управління мінімізується середньоквадратична помилка (СКП):

$$E_y = \frac{1}{2}(y_d - y)^2. \quad (1.13)$$

Питання стійкості та керованості.

Аналіз стійкості, збіжності та керованості нейромережевих систем управління є складною теоретичною проблемою. Ці питання детально досліджуються в численних роботах (зокрема, [27–37]), де розглядаються умови стійкості адаптивних і навчальних алгоритмів, а також гарантії поведінки системи у перехідних процесах.

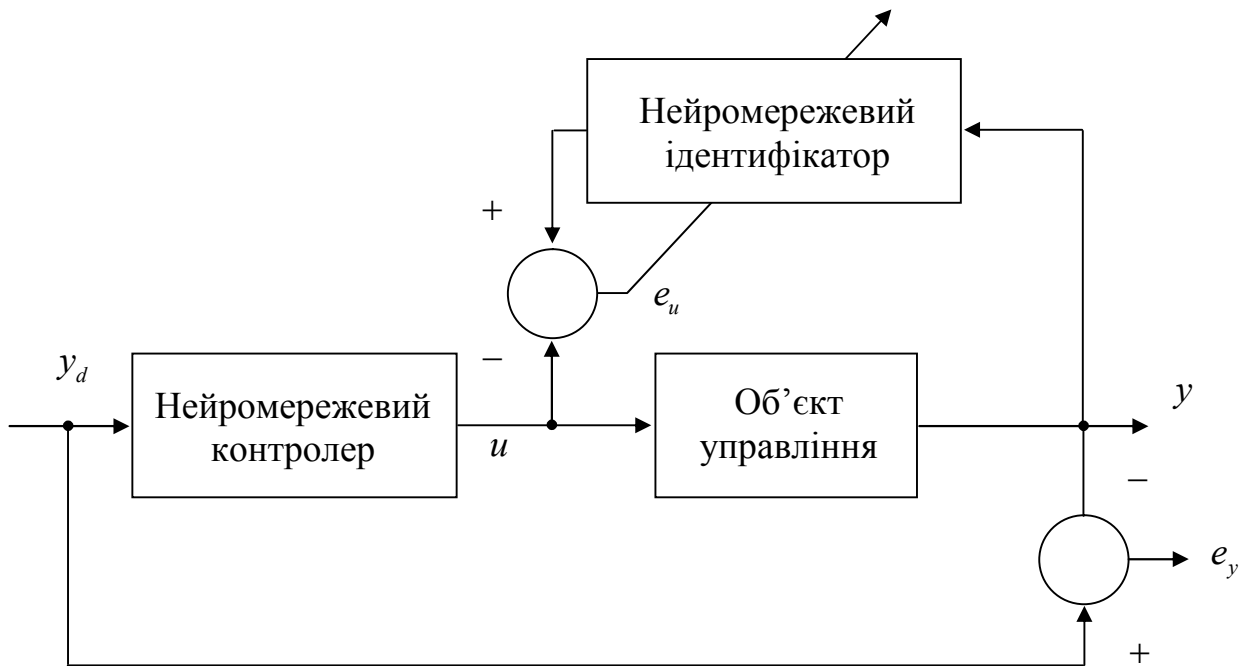


Рисунок 1.8 – Схема непрямого управління нейромережею

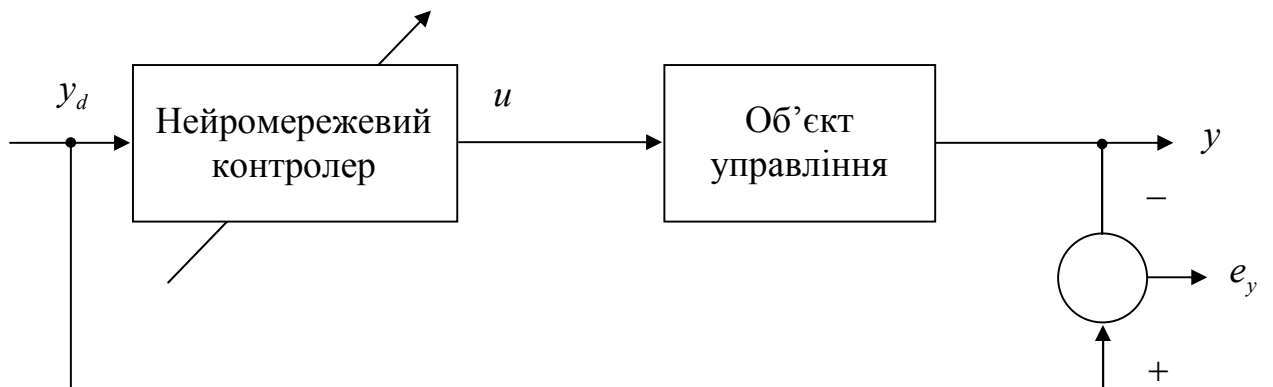


Рисунок 1.9 – Схема прямого управління нейромережею

Принцип предиктивного регулювання на основі моделі (model predictive control, MPC) [38–42] (рис. 1.10) полягає у формуванні такої послідовності керуючих впливів на об'єкт, яка мінімізує відхилення між заданим значенням і вихідним сигналом, що прогнозується моделлю процесу на певному горизонті прогнозування (див. пунктирні лінії на рис. 1.10).

У системах MPC регулятор реалізується у вигляді ітераційного алгоритму оптимізації. Основні його особливості такі:

- оптимізація виконується чисельними методами;
- мінімізується функціонал якості, який описує бажані характеристики процесу регулювання;
- для реального впливу на об'єкт використовується лише перший елемент знайденої оптимальної послідовності керуючих дій;
- процедура оптимізації повторюється на кожному кроці дискретизації, що забезпечує ковзне (receding horizon) планування.

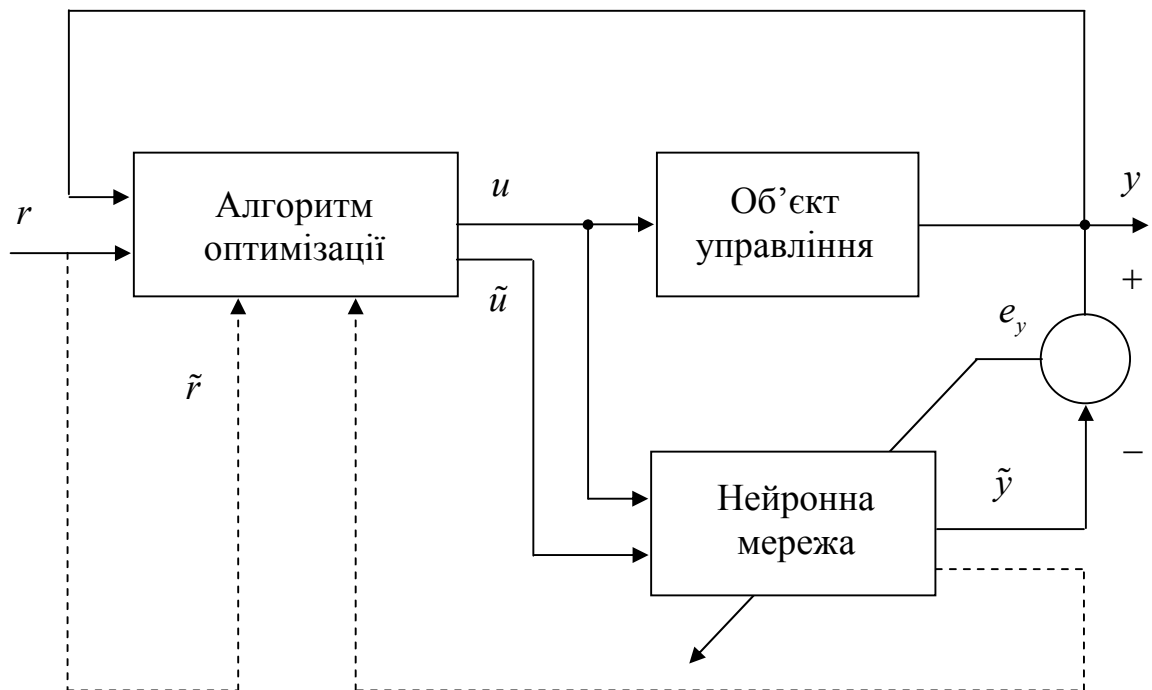


Рисунок 1.10 – Схема предиктивного регулювання на основі моделі

Перевагою методу MPC є висока якість регулювання, навіть при наявності істотних нелінійностей у динаміці об'єкта. Водночас суттєвими недоліками є висока обчислювальна складність та підвищені вимоги до точності моделі об'єкта управління, оскільки ефективність MPC значною мірою залежить від якості прогнозування.

У цій магістерській роботі розглядається використання нелінійного предиктивного регулювання.

Порівняння методів управління.

Окремим напрямком теоретичних досліджень, присвячених застосуванню ШНМ у системах управління, є їх порівняння з іншими відомими методами. Мета таких досліджень – виявлення сильних і слабких сторін, визначення специфічних характеристик нейромережевих регуляторів та їх придатності до різних умов експлуатації.

У роботі [43] подано огляд характеристик різних систем управління. Порівняння охоплює такі критерії, як стійкість, швидкість збіжності, ефективність у шумових умовах, вимоги до обчислювальних ресурсів, пам'яті та ін.

Результати узагальнено в табл. 1.1.

Таблиця 1.1 – Огляд характеристик методів управління

Критерій	Управління із зворотним зв'язком і регульованими коефіцієнтами	Адаптивне управління з еталонною моделлю Ляпунова	Нейромережеве управління
Стійкість зворотного зв'язку	Якнайгірша	Якнайкраща	Середня
Швидкість збіжності	Якнайкраща	Середня	Якнайгірша
Робота в реальному часі	Середня	Середня	Якнайкраща
Складність програми управління	Якнайгірша	Середня	Середня
Помилка стеження	Середня	Якнайкраща	Середня
Зменшення завад	Якнайкраща	Якнайгірша	Середня
Робастність Розузгодження моделі	Якнайгірша	Середня	Якнайкраща

1.4 Призначення та класифікація рольгангів

Рольганги є спеціалізованим обладнанням, призначеним для переміщення металевого прокату за допомогою обертових роликів. Вони відіграють ключову роль у процесі прокатки, забезпечуючи плавний та безперервний рух металу між різними вузлами прокатного стану. Рольганги поділяються на кілька основних типів залежно від їх функціонального призначення.

Робочі рольганги встановлюються безпосередньо у вузлі стану та відповідають за подачу металу до валків і його відведення після прокатки. У великих станах перші ролики робочих рольгангів розташовані у станині кліті, що забезпечує надійне захоплення металу, особливо коротких злитків.

Допоміжні робочі рольганги, або подовжувальні, виконують роль продовження основних рольгангів і застосовуються у випадках, коли довжина прокатуваного металу перевищує довжину базових рольгангів. Вони дозволяють підтримувати стабільний контакт металу з роликами протягом усього руху по стану.

Транспортні рольганги призначені для переміщення металу між різними механізмами стану або між окремими станами. Залежно від напрямку руху металу вони бувають підвідними або відвідними, а рольганг, розташований на початку стану для прийому металу, називають приймальним.

Основними технічними параметрами рольгангів є діаметр і довжина бочки ролика, крок рольганга (відстань між осями сусідніх роликів) та кількість роликів. Діаметр роликів підбирається мінімально можливим, щоб знизити масу конструкції, при цьому забезпечуючи необхідну міцність. Наприклад, для робочих рольгангів обтискових станів і слябінгів діаметр роликів зазвичай становить 600–400 мм, а для листових і сортових станів – 350–200 мм.

Довжина бочки роликів, як правило, відповідає або трохи перевищує довжину бочки валків стану, а крок рольганга підбирається таким чином, щоб метал одночасно спирався мінімум на два ролики, запобігаючи його прогинанню під власною вагою. Кількість роликів визначається довжиною прокату і мо-

же сягати декількох десятків у великих станах. Лінійна швидкість робочих і відвідних рольгангів зазвичай перевищує швидкість прокатки на певну величину, забезпечуючи безперервний рух металу.

Регулювання швидкості рольгангів може бути плавним або ступінчастим. Плавне регулювання застосовується у робочих рольгангах нереверсивних регульованих станів, тоді як ступінчасте – у робочих рольгангах реверсивних обтискових станів. Це дозволяє забезпечити оптимальні умови руху металу, враховуючи його довжину, масу та характер прокатки.

1.5 Технічні вимоги до електроприводів рольгангів

Електроприводи робочих і розкочувальних рольгангів працюють у повторно-короткочасному реверсивному режимі з високою частотою включень. Наприклад, частота включень за годину може досягати 1000–1200 для робочих рольгангів блюмінга та слябінгу, 450 – для розкочувальних рольгангів блюмінга та слябінгу, і 850 – для робочих рольгангів товстолистового стану гарячої прокатки.

Висока частота включень вимагає від електроприводів підвищених динамічних характеристик під час розгону і гальмування, оскільки час реакції допоміжних механізмів істотно впливає на продуктивність прокатного стану.

Електроприводи повинні забезпечувати широкий діапазон регулювання швидкості, який складає не менше 10–20:1. Структура приводу передбачає реверсивну систему з пропорційним регулятором частоти обертання або ЕРС, а також підлеглим регулятором струму якоря. Основною вимогою до системи управління є висока швидкодія. Так, час реверсу струму від одного знаку до іншого не повинен перевищувати 25–40 мс.

Завдання швидкості задається ступінчасто або плавно, через контактний або сельсинний командоапарат, із застосуванням спеціального задатчика інтенсивності, який формує динамічний струм двигуна. Така організація дозволяє досягати високої точності регулювання швидкості та забезпечує

стабільну роботу рольгангів у режимі реального часу, навіть за умов великих навантажень і швидких змін режимів роботи стану.

1.7 Постановка задачі синтезу нейромережевої системи управління

Аналіз наукової літератури свідчить про те, що існує велика кількість підходів до синтезу систем управління, проте універсального і бездоганного регулятора, який однаково ефективно працював би для всіх об'єктів, наразі не існує. При розробці системи управління необхідно враховувати специфіку об'єкта: для електроприводу механізму розвороту екскаватора характерна наявність пружно зв'язаних з платформою інерційних мас, що включають стрілу, роторне колесо і консоль противаги. Така конструкція об'єкта призводить до появи значних коливань у перехідних процесах, а традиційні регулятори часто не здатні забезпечити необхідний рівень якості управління.

Теорія адаптивного та оптимального управління дозволяє формувати системи, які здатні мінімізувати або максимізувати обраний критерій якості управління. Сучасні підходи включають не лише методи синтезу регуляторів, але й методи ідентифікації та оцінки станів об'єкта. Алгоритми ідентифікації допомагають визначити структуру моделі об'єкта та уточнити її параметри як під час процесу управління, так і поза ним, що дозволяє створювати як стандартні, так і адаптивні системи управління.

Класична теорія управління базується на розробках лінійних систем, які широко застосовуються і для дослідження нелінійних об'єктів. Проте досягнення оптимальних результатів у такому випадку можливо лише тоді, коли нелінійність об'єкта невелика або об'єкт характеризується великими постійними часами та стабільністю у відкритому контурі. Управління динамічними об'єктами ускладнюється необхідністю постійного контролю не лише параметрів моделі, але й іноді її структури. Найпоширенішим методом апроксимації нелінійностей є використання рядів Вольтерра, Гаммерштейна, Вінера, поліномів Колмогорова-Габора та подібних підходів. Однак ці методи мають обме-

ження: для їх застосування потрібно попередньо визначити порядок моделі та вирішити задачу структурної ідентифікації. Крім того, зовнішні збурення та похибки вимірювань ускладнюють процес управління, а потреба у швидкодіючих алгоритмах для роботи у реальному часі ставить перед розробниками додаткові завдання.

У зв'язку з цим перспективним підходом стає застосування штучних нейронних мереж (ШНМ), які відкривають нові можливості при управлінні нелінійними динамічними об'єктами. Нейронні мережі, як частина штучного інтелекту, обробляють інформацію паралельно всіма елементами, що забезпечує швидку обробку сигналів і можливість управління у реальному часі. Велика кількість міжнейронних зв'язків підвищує стійкість мережі до локальних помилок, а здатність до навчання та узагальнення накопичених знань дозволяє працювати з новими, раніше не відомими даними.

Особливо важливою є наявність у структурі ШНМ нейронів з нелінійними функціями активації, що дозволяє моделювати і управляти складними нелінійними процесами. Для навчання мережі необхідно подати достатню кількість навчальних даних у форматі «вхід-вихід», яка за обсягом значно перевищує кількість зв'язків у мережі, а також коректно підібрати структуру нейромоделі, кількість шарів і нейронів, та тип функцій активації (сигмоїдальні, радіально-базисні або інші). Такі властивості ШНМ дозволяють ефективно вирішувати задачі управління, де традиційні методи не можуть забезпечити практично реалізоване рішення.

Побудова системи управління на основі нейрорегулятора передбачає послідовне виконання двох ключових етапів: ідентифікації об'єкта та розробки алгоритму управління з урахуванням критерію якості. Ідентифікація здійснюється шляхом навчання нейромоделі об'єкта на основі експериментальних даних, що відображають відповідність вхідних і вихідних параметрів системи.

Метою даної роботи є розробка нейромережевої системи управління електроприводом рольганга перед ножицями стана гарячої прокатки, який відзначається високими динамічними характеристиками та наявністю пружних еле-

ментів. Така конфігурація об'єкта робить його багатомасовою системою, а система управління повинна забезпечувати високоякісне регулювання як за задаючими, так і за збурюючими діями, навіть при ступінчастому вхідному сигналі з випадковою амплітудою.

Для досягнення поставленої мети вирішуються такі завдання:

1. Перевірка привідного двигуна електроприводу рольганга перед ножицями по нагріву та перевантажувальній здатності, синтез системи управління та моделювання її роботи на ЕОМ.

2. Розробка математичної моделі системи з урахуванням податливості механічних зв'язків, моделювання багатомасової системи та аналіз отриманих результатів.

3. Розробка структурної схеми нейромережевої системи управління, здатної забезпечити високоякісне регулювання з урахуванням кінцевої жорсткості механічних елементів.

4. Синтез моделі багатомасової системи з високою точністю ідентифікації за допомогою багатошарової прямонаправленої мережі.

5. Створення нейромережевого регулятора на основі принципу предиктивного управління з використанням нейромоделі об'єкта, що забезпечить високу швидкодію та якість управління.

6. Моделювання нейромережевої системи управління та детальний аналіз отриманих результатів для оцінки ефективності реалізованого підходу.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Проведено аналіз літератури по темі роботи, що підтвердив доцільність використання штучних нейронних мереж (ШНМ) у задачах управління. Сучасні системи керування складними динамічними об'єктами повинні швидко адаптуватися до змінюваних умов функціонування та здійснювати оперативне коригування параметрів і структури управлінських законів. Цим вимогам відповідає теорія адаптивного управління, одним із ефективних засобів реалізації якої є нейронне моделювання та управління.

2. Показано, що для задач управління доцільно застосовувати багатошарові мережі персептронного типу, включаючи їх варіанти з зворотними зв'язками та лініями затримки на входах, що формуються за вхідними та вихідними сигналами. Використання динамічних методів навчання таких мереж дозволяє створювати адаптивні системи управління, які ефективно функціонують у умовах різноманітних невизначеностей.

3. Встановлено, що важливим етапом розв'язання задач управління, прогнозування поведінки та моделювання динамічних систем є отримання адекватних математичних моделей об'єкта. Показано, що завдання ідентифікації нелінійних динамічних об'єктів успішно вирішуються за допомогою ШНМ, зокрема багатошарового персептрона та радіально-базисних мереж.

4. Для оптимізації роботи об'єкта необхідно здійснювати корекцію параметрів регуляторів відповідно до зміни робочих умов. Це обґрунтовує доцільність розробки систем управління на основі адаптивного підходу у поєднанні з методами ШНМ, що дозволяє, з одного боку, будувати прості нейромережеві моделі, а з іншого – адаптивно корегувати параметри як моделей, так і регуляторів.

5. Проаналізовано сучасні структури систем нейрорегулювання та обґрунтовано вибір принципу нелінійного предиктивного регулювання як основи для побудови нейромережевого регулятора системи.

6. Розглянуто ефективну реалізацію узагальненого управління з прогнозом на основі багатошарової прямонаправленої нейронної мережі як нелінійної моделі об'єкта управління. Показано, що застосування оптимізаційного алгоритму Ньютона-Рафсона дозволяє зменшити число ітерацій для збіжності порівняно з іншими методами, що забезпечує високу швидкодію алгоритму та можливість його використання для управління в реальному часі.

7. Проаналізовано технічні вимоги до систем управління електроприводами основних і допоміжних механізмів прокатних станів. З урахуванням пружності механічних зв'язків показано доцільність використання нейромережевих технологій для синтезу нейромережевої системи управління електроприводом рольганга перед ножицями стана гарячої прокатки.

8. Сформульовано задачу дослідження, яка полягає в розробці нейромережевої системи управління електроприводом рольганга стана гарячої прокатки з урахуванням пружних механічних зв'язків, що забезпечує високоякісне регулювання за задаючими та збурюючими впливами.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ ЕЛЕКТРОПРИВОДОМ РОЛЬГАНГА СТАНА ГАРЯЧОЇ ПРОКАТКИ

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

Для синтезу системи автоматичного управління електроприводом рольганга стана гарячої прокатки необхідно виходити з технічних характеристик механізму та електроустаткування, а також визначених вимог до показників якості функціонування системи.

Технічні дані рольганга.

Рольганг призначений для переміщення металевого зливку на прокатному стані, і його робота залежить від маси зливку, параметрів роликів та швидкості руху. Основні технічні характеристики рольганга включають:

Маса зливку $m_m = (5300 \div 6800) \text{ кг}$.

Діаметр бочки ролика $d_p = 0,4 \text{ м}$.

Діаметр шийки ролика $d_{ш} = 0,2 \text{ м}$.

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

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 залежить від стану навантаження:

незавантаженого рольганга $J_{мн} = 147 \text{ кг} \cdot \text{м}^2$;

завантаженого рольганга (масою розкату $6,8 \text{ т}$) $J_{мз} = 420 \text{ кг} \cdot \text{м}^2$;

Маса окремого ролика становить $m_p = 1110 \text{ кг}$.

Технічні дані електроустаткування.

Для забезпечення надійної роботи рольганга використовується електропривод на основі двигуна постійного струму з незалежним збудженням типу Д810. Основні параметри двигуна:

Номінальна потужність $P_n = 55 \text{ кВт}$.

Номінальна напруга $U_n = 220 \text{ В}$.

Номінальний струм якоря $I_n = 282 \text{ А}$.

Номінальна частота обертання $n_n = 550 \text{ об / хв}$.

Збудження – незалежне.

Число активних провідників обмотки якоря $N = 234$.

Число паралельних гілок якоря $2a = 2$.

Число полюсів $2p = 4$.

Магнітний потік полюса $\Phi_n = 47,5 \cdot 10^{-3} \text{ Вб}$.

Номінальний струм збудження $I_{zn} = 3,85 \text{ А}$.

Опір якоря і додаткових полюсів $R_{я} + R_{дп} = 0,0356 \text{ Ом}$.

Момент інерції $J_{дв} = 3,625 \text{ кг} \cdot \text{м}^2$.

Тривалість включення ПВ=40%.

Редуктор рольганга характеризується передавальним числом – $i = 5,75$.

Для керування двигуном використовується тиристорний електропривод КТЕУ 500/220-13222:

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

Номінальний струм $I_{дн} = 500 \text{ А}$.

Тиристорний перетворювач живиться через трансформатор ТСП-160/0,7 У4 з наступними параметрами:

Номінальна напруга мережевої обмотки $U_1 = 380 \text{ В}$.

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

Номінальна випрямлена напруга перетворювача $U_{\text{дн}} = 230 \text{ В}$.

Номінальна сила випрямленого струму $I_{\text{дн}} = 500 \text{ А}$.

Напруга короткого замикання $U_{\text{к}} = 5,8\%$.

Вимоги до показників якості системи управління.

Для забезпечення високоякісного регулювання швидкості рольганга визначено наступні основні вимоги до системи автоматичного управління:

- Перерегулювання швидкості при ступінчастій вхідній дії не повинно перевищувати 10%;
- Час регулювання швидкості – не більше 5 секунд.

Система управління повинна забезпечувати точне та стабільне керування швидкістю рольганга, оперативне реагування на зміну навантаження та високу динамічну стійкість при роботі в умовах значних інерційних і пружно-зв'язаних механічних елементів.

2.2 Перевірка електродвигуна за нагрівом та перевантажувальною здатністю

Для оцінки працездатності електродвигуна рольганга перед ножицями стана гарячої прокатки необхідно визначити моменти, що виникають під час роботи, та перевірити відповідність двигуна вимогам по перевантажувальній здатності і тепловому режиму.

Спочатку визначається момент холостого ходу рольганга, приведений до валу двигуна:

$$M_{\text{х}} = \frac{Z_{\text{р}} G_{\text{р}} d_{\text{ш}} \mu_{\text{ш}}}{2i\eta_{\text{х}}},$$

де $G_{\text{р}}$ – сила тяжіння одного ролика;

$$G_{\text{р}} = m_{\text{р}} g = 1110 \cdot 9,8 = 10878 \text{ Н},$$

$\mu_{\text{ш}}$ – коефіцієнт тертя ковзання в підшипниках ролика (зазвичай $\mu_{\text{ш}} = 0,01$);

$\eta_{\text{х}} = 0,5\eta_{\text{і}}$ – ККД редуктора при холостому ході, $\eta_{\text{х}} = 0,46$;

$\eta_{\text{і}}$ – ККД редуктора при номінальному навантаженні, $\eta_{\text{і}} = 0,92$

$$M_{\text{х}} = \frac{3 \cdot 10878 \cdot 0,2 \cdot 0,01}{2 \cdot 5,75 \cdot 0,46} = 11,9 \text{ Нм}.$$

Далі обчислюється повний момент статичного опору при транспортуванні металу:

$$M_{\text{с}} = \frac{(Z_{\text{р}} G_{\text{р}} + G_{\text{м}})(d_{\text{ш}}/2)\mu_{\text{ш}} + G_{\text{м}}f}{i\eta_{\text{х}}},$$

де $G_{\text{м}}$ – сила тяжіння металу; розрахунок моментів проведемо для максимальної маси заготовки на рольгангу $m_{\text{max}} = 6800 \text{ кг}$

$$G_{\text{м}} = m_{\text{max}}g = 6800 \cdot 9,8 = 66600 \text{ Н},$$

f – коефіцієнт тертя кочення металу по роликах. Для гарячого металу приймається $f = 0,15 \cdot 10^{-2} \text{ м}$.

Момент, необхідний для переміщення металу, невеликий, тому часто використовується значення $\eta_{\text{х}} = 0,5\eta_{\text{і}}$

$$M_{\text{с}} = \frac{(3 \cdot 10878 + 66600)(0,2/2) \cdot 0,01 + 66600 \cdot 0,15 \cdot 10^{-2}}{5,75 \cdot 0,46} = 73,8 \text{ Нм}.$$

Статичний момент при буксуванні металу визначається за формулою:

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

$\mu_{\text{б}}$ – коефіцієнт тертя металу об ролик при буксуванні (зазвичай $\mu_{\text{б}} = 0,3$).

$$M_{\text{сб}} = \frac{(3 \cdot 10878 + 66600) \cdot 0,2 \cdot 0,01 + 66600 \cdot 0,3 \cdot 0,4}{2 \cdot 5,75 \cdot 0,46} = 782 \text{ Нм}.$$

Для оцінки динамічного режиму розраховується момент двигуна, необхідний для подолання інерційних сил металу, роликів та самого 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 при транспортуванні металу без прослизання $a_{\text{кр}}$ становить $a_{\text{кр}} = 1,47 \div 2,94 \text{ м/с}^2$, приймаємо $a = 2 \text{ м/с}^2$).

J_p – момент інерції ролика відносно осі обертання;

J_m – момент інерції металу, приведений до осі ролика.

У технічних даних задано момент інерції завантаженого рольганга

$$J_{\text{з м}} = Z_p J_p + J_m = 420 \text{ кг} \cdot \text{м}^2,$$

J_d – момент інерції двигуна і редуктора.

$$J_d = 1,1 \cdot J_{\text{дв}},$$

$$J_d = 1,1 \cdot 3,625 = 3,9875 \text{ кг} \cdot \text{м}^2.$$

Підставляючи числові значення, розраховуємо динамічний момент

$$M_{\text{дин}} = 2,2 \cdot \left[\frac{420}{0,92 \cdot 5,75 \cdot 0,4} + \frac{5,75 \cdot 3,9875}{0,4} \right] = 1028 \text{ Нм}.$$

Повний момент, що розвивається двигуном під час прискорення, обчислюється як

$$M_{\Pi} = M_c + M_{\text{дин}},$$

$$M_{\Pi} = 73,8 + 1028 = 1101,8 \text{ Нм},$$

а при гальмуванні:

$$M_{\text{т}} = M_c - M_{\text{дин}},$$

$$M_{\text{т}} = 73,8 - 1028 = -947,2 \text{ Нм}.$$

Гранично-виправдовуваний момент визначається як:

$$M_{\text{гв}} = \frac{(Z_p G_p + G_m) d_{\text{ш}} \mu_{\text{ш}} \pm G_m \beta \mu_6 d_p}{2i\eta_{\text{н}}} \pm 2a_{\text{кр}} \left(\frac{Z_p J_p}{i\eta_{\text{н}} d_p} + i \frac{J_d}{d_p} \right),$$

де знак плюс відповідає режиму розгону, а мінус – гальмуванню

У вихідних даних наведено момент інерції незавантаженого рольганга, тобто $Z_p J_p = J_{\text{мн}}$.

Обчислюємо гранично-виправдовуваний момент при розгоні

$$M_{\text{гв}} = \frac{(3 \cdot 10878 + 66600) 0.2 \cdot 0.01 + 66600 \cdot 0.3 \cdot 0.4}{2 \cdot 5.75 \cdot 0.92} +$$

$$+ 2 \cdot 2.94 \left(\frac{147}{5.75 \cdot 0.92 \cdot 0.4} + \frac{5.75 \cdot 3.9875}{0.4} \right) = 1532 \text{ Нм},$$

і при гальмуванні приводу

$$M_{\text{гв}} = \frac{(3 \cdot 10878 + 66600) 0.2 \cdot 0.01 - 66600 \cdot 0.3 \cdot 0.4}{2 \cdot 5.75 \cdot 0.92} -$$

$$- 2 \cdot 2.94 \left(\frac{147}{5.75 \cdot 0.92 \cdot 0.4} + \frac{5.75 \cdot 3.9875}{0.4} \right) = -1497 \text{ Нм}.$$

Ці величини використовуються для перевірки перевантажувальної здатності двигуна.

Номінальний момент двигуна визначається за його параметрами:

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

де

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

$$\omega_{\text{н}} = \frac{3,14 \cdot 550}{30} = 57,6 \text{ c}^{-1} ,$$

$$M_{\text{н}} = \frac{55 \cdot 10^3}{57,6} = 955 \text{ Нм} .$$

Відношення фактичного моменту до номінального $\frac{M_{\text{гв}}}{M_{\text{н}}} = \frac{1532}{955} = 1,59$, яке

не перевищує допустимої перевантажувальної здатності двигунів постійного струму (зазвичай до 2,5), підтверджує правильність вибору двигуна.

Для перевірки двигуна по тепловому режиму визначають час пуску $t_{\text{п}}$, гальмування $t_{\text{г}}$ і час роботи на постійній швидкості $t_{\text{р}}$. Розглядається найбільш критичний випадок: розгін до максимальної швидкості V_{max} та гальмування від неї до зупинки.

$$t_{\text{п}} = t_{\text{г}} = \frac{V_{\text{max}}}{a} = \frac{2}{2} = 1 \text{ c} .$$

Час циклу при числі включень в годину $k_{\text{max}} = 600$ визначається як

$$t_{\text{ц}} = \frac{3600}{600} = 6 \text{ c} .$$

Час роботи двигуна при повній тривалості включення ПВ = 100% і при усталеному режимі

$$t_{\text{у}} = t_{\text{ц}} - 2t_{\text{п}} ,$$

$$t_y = 6 - 2 \cdot 1 = 4 \text{ с}.$$

Визначаємо квівалентний момент

$$M_e = \sqrt{\frac{M_{\Pi}^2 t_{\Pi} + M_{\Gamma}^2 t_{\Gamma} + M_c^2 t_y}{t_{\Pi}}},$$

$$M_e = \sqrt{\frac{1101,8^2 \cdot 1 + 947,2 \cdot 1 + 73,8^2 \cdot 4}{6}} = 589 \text{ Нм}.$$

Оскільки еквівалентний момент менший за номінальний момент двигуна, то двигун задовольняє вимоги теплового режиму.

Таким чином, проведені розрахунки показують, що двигун рольганга має достатню перевантажувальну здатність та відповідає технічним вимогам по нагріву, що дозволяє використовувати його для автоматичного управління з високою динамічною точністю.

2.3 Розрахунок параметрів силового кола електроприводу

Для забезпечення стабільної роботи електроприводу рольганга необхідно розрахувати основні параметри силового кола, враховуючи характеристики тиристорного перетворювача, трансформатора та двигуна постійного струму.

Почнемо з визначення еквівалентного активного опору тиристорного перетворювача:

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

де R_x – еквівалентний опір анодного перекриття тиристорів

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

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

$I_{\text{дн}}$ – номінальний випрямлений струм перетворювача;

$E_{\text{д0}}$ – ЕРС холостого ходу перетворювача; для трифазної мостової схеми

$$E_{\text{д0}} = 2,34U_{2\phi} = 3,34 \cdot 115 = 269 \text{ В}.$$

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

$$R_x = 0,5 \cdot 0,058 \cdot \frac{269}{500} = 0,0156 \text{ Ом},$$

R_t – приведений до ланцюга випрямленого струму активний опір обмоток трансформатора

$$R_t = (0,1 - 0,2)U_k \frac{\pi E_{\text{д0}}}{m I_{\text{дн}}},$$

m – фазність пульсації випрямленої напруги; для трифазної мостової схеми $m = 6$

$$R_t = 0,15 \cdot 0,058 \frac{3,14}{6} \cdot \frac{269}{500} = 0,0024 \text{ Ом},$$

$$R_{\text{пр}} = 0,0156 + 0,0024 = 0,018 \text{ Ом}$$

Далі визначається опір якоря електродвигуна

$$R_{\text{як}} = 1,36 \cdot (R_{\text{я}} + R_{\text{дп}} + R_{\text{ко}}).$$

Згідно паспортних даних двигуна $R_{\text{я}} + R_{\text{дп}} = 0,0356 \text{ Ом}$, $R_{\text{ко}} = 0$

$$R_{\text{як}} = 1,36 \cdot 0,0356 = 0,0484 \text{ Ом}.$$

Номінальна ЕРС двигуна

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

$$E_{\text{н}} = 220 - 282 \cdot 0,0484 = 206 \text{ В}.$$

Коефіцієнт ЕРС

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

$$c = \frac{206}{57,6} = 3,58 .$$

Коефіцієнт посилення двигуна

$$k_{\text{д}} = \frac{1}{c} ,$$

$$k_{\text{д}} = \frac{1}{3,58} = 0,28 .$$

Коефіцієнт посилення тиристорного перетворювача

$$k_{\text{тп}} = \frac{E_{\text{д0}}}{U_{\text{м}}} ,$$

Амплітудне значення напруги порівняння $U_{\text{м}}$ для перетворювача приймається рівним $U_{\text{м}} = 15B$

$$k_{\text{тп}} = \frac{269}{15} = 18 .$$

Активний опір головного ланцюга електроприводу визначається як сума опорів перетворювача (разом з трансформатором) та якоря двигуна:

$$R_{\Sigma} = R_{\text{як}} + R_{\text{пр}} ,$$

$$R_{\Sigma} = 0,0484 + 0,018 = 0,0669 \text{ Ом} .$$

Індуктивність якірного кола електродвигуна розраховується за формулою:

$$L_{\text{я}} = \gamma \frac{1}{\omega_{\text{н}} p} \cdot \frac{U_{\text{н}}}{I_{\text{н}}} ,$$

де γ – коефіцієнт, який для некомпенсованих двигунів дорівнює 0,6;
 p – число пар головних полюсів;

$$L_{\text{я}} = 0,6 \frac{1}{57,6 \cdot 2} \cdot \frac{220}{282} = 0,004 \text{ Гн}.$$

Приведена до кола випрямленого струму індуктивність трансформатора визначається як

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

f – частота мережі;

$$L_{\text{т}} = 0,058 \frac{1}{2 \cdot 6 \cdot 50} \cdot \frac{262}{500} = 0,5 \cdot 10^{-4} \text{ Гн}.$$

Загальна індуктивність головного ланцюга визначається як сума індуктивностей якоря та трансформатора:

$$L_{\Sigma} = L_{\text{я}} + L_{\text{т}},$$

$$L_{\Sigma} = 0,004 + 0,5 \cdot 10^{-4} = 0,00405 \text{ Гн}.$$

Далі обчислюються електромагнітні постійні часу:

для головного ланцюга

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

$$T_{\text{е}} = \frac{0,00406}{0,0664} = 0,061 \text{ с};$$

для якорного ланцюга електродвигуна

$$T_{\text{а}} = \frac{L_{\text{я}}}{R_{\text{як}}},$$

$$T_a = \frac{0,0043}{0,0484} = 0,083 \text{ с}.$$

Нарешті, визначається електромеханічна постійна часу електроприводу, що характеризує динаміку системи в режимі максимального навантаження:

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

де J_m – сумарний момент інерції на валу двигуна, включаючи момент інерції редуктора, роликів та металевої заготовки

$$J_m = J_d + \frac{J_{m3}}{i^2},$$

$$J_m = 3,9875 + \frac{420}{5,75^2} = 3,9875 + 12,7 = 16,69 \text{ кг} \cdot \text{м}^2,$$

$$T_m = \frac{16,69 \cdot 0,0664}{3,58^2} = 0,21 \text{ с}.$$

Ці розрахунки дозволяють оцінити динамічні властивості електроприводу та забезпечити стабільну роботу системи управління рольганга при різних режимах прокатки.

2.4 Вибір структури системи автоматичного регулювання координат електроприводу

Система автоматичного управління електроприводом рольганга повинна забезпечувати не лише необхідні механічні характеристики двигуна у статичних умовах, але й стабільну та точну роботу у динамічному режимі. Важливими завданнями такої системи є високоякісні перехідні процеси, оптимізація роботи електроприводу за різних режимів роботи механізму та підтримка заданих параметрів незалежно від зовнішніх навантажень. Досягти цього можливо лише шляхом застосування замкнених систем автоматичного регулювання, які дозво-

ляють оперативно реагувати на зміни стану об'єкта управління. Основна мета при проектуванні таких систем полягає в мінімізації часу перехідного процесу та забезпеченні допустимого рівня перерегулювання для будь-якого контролюваного параметра.

Характер переходу системи визначається співвідношенням постійних часу всіх елементів. Тому для забезпечення високої якості перехідних процесів необхідно правильно вибрати ці співвідношення, а також вводити коригуючі регулятори, які компенсують інерційні властивості об'єкта управління. Загальна методика побудови систем автоматичного регулювання передбачає складання структурної схеми, визначення функцій окремих ланок і включення регуляторів, що забезпечують необхідну реакцію системи.

На практиці для електроприводів рольгангів та інших подібних механізмів найбільше застосовуються системи підлеглого регулювання параметрів. Принцип їх роботи полягає у послідовній компенсації інерційних властивостей об'єкта за рахунок включення кількох контурів регулювання. Кількість таких контурів зазвичай відповідає кількості керованих параметрів. Кожен контур складається з двох основних компонентів: об'єкта регулювання та регулятора, що забезпечує необхідну реакцію на зміни стану об'єкта.

Для розглядуваного електроприводу рольганга доцільним є використання двоконтурної системи автоматичного управління з дворазовою інтеграцією. Вона включає внутрішній контур струму, який забезпечує швидке регулювання електричного потоку через двигун, та зовнішній контур, замкнутий по ЕРС двигуна, який контролює швидкість обертання валу. Такі системи управління швидкістю з зворотним зв'язком по ЕРС застосовуються, коли точність регулювання не потребує надвисоких значень, а важлива стабільність і швидкість реакції. Саме тому більшість електроприводів загальнопромислових механізмів відноситься до цього класу, адже вони забезпечують достатній рівень якості регулювання при відносно простій схемі управління.

2.5 Синтез послідовного коректуючого пристрою контура струму

В проектованій системі контур струму виконує функцію внутрішнього контура електроприводу. Оптимізація цього контура проводиться за певними допущеннями:

- на вході датчика струму встановлений аперіодичний фільтр із постійною часу $T_{\text{дт}}$
- на вході системи імпульсно-фазового управління (СІФУ) також встановлений фільтр із постійною часу $T_{\text{ф}}$;
- вплив проти-ЕРС двигуна на динамічні властивості об'єкта не враховується.

Динамічні властивості тиристорного перетворювача спільно із СІФУ описуються передавальною функцією виду:

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

де $k_{\text{тп}}$ – коефіцієнт підсилення тиристорного перетворювача із системою імпульсно-фазового управління;

τ – середнє запізнення перетворювача, що визначається як

$$\tau = \frac{1}{2mf},$$

де m – число пульсацій випрямленої ЕРС за період (для трифазної мостової схеми $m=6$);

f – частота мережі.

Алгоритмічна схема контура струму наведена на рис. 2.1 а.

Об'єкт регулювання контура включає:

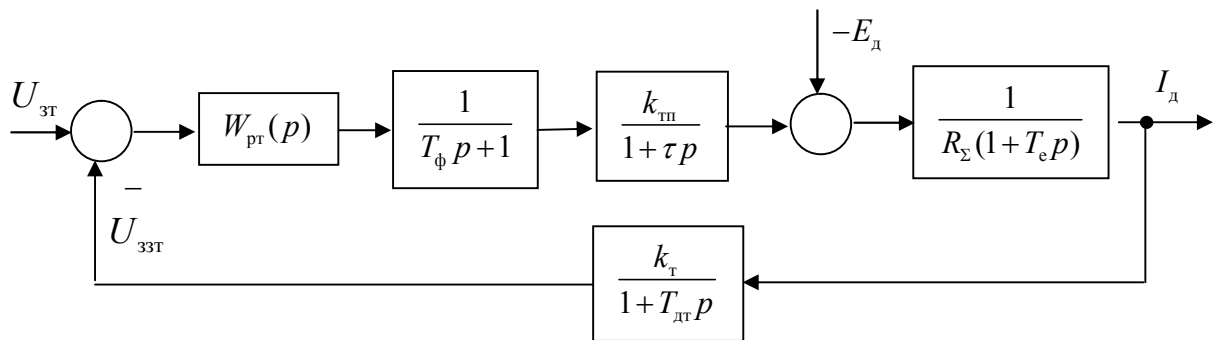
- якірний ланцюг двигуна, який можна описати як аперіодичну ланку з коефіцієнтом підсилення $1/R_{\Sigma}$ і постійною часу $T_{\text{е}}$;

- тиристорний перетворювач із СІФУ та фільтром на вході;
- датчик струму з аперіодичною ланкою на виході.

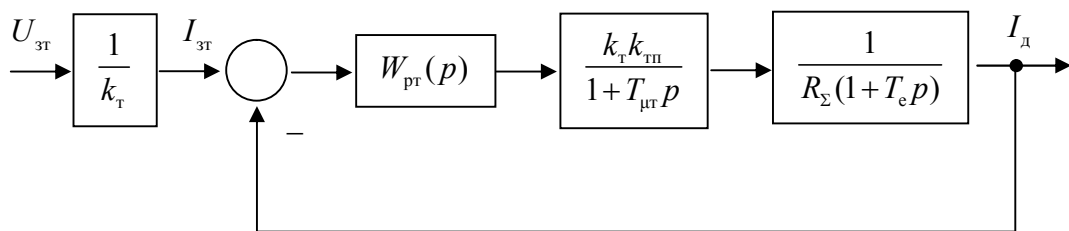
Для використання методики синтезу послідовного коректуючого пристрою контур перетворюється у схему з одиничним зворотним зв'язком. Аперіодичні ланки з малими постійними часу $T_{\text{дт}}$, $T_{\text{ф}}$ і τ можна об'єднати в одну аперіодичну ланку з малою некомпенсованою постійною часу

$$T_{\text{мт}} = \tau + T_{\text{дт}} + T_{\text{ф}},$$

Приймаємо $T_{\text{мт}} = 5 \text{ мс}$. Форсуюча ланка $(1 + T_{\text{дт}}p)$, що стоять на вході замкнутого контура, при розрахунку малої некомпенсованої постійної часу контура швидкості не враховуються, відповідно до загальноприйнятої методики синтезу. Перетворена схема показана на рис. 2.1 б.



а)



б)

Рисунок 2.1 – Алгоритмічні схеми контура струму:

а – початкова схема;

б – перетворена схема з одиничним зворотним зв'язком

Контур струму оптимізується на основі модульного критерію оптимізації, при якому бажана передавальна функція розімкненого оптимізованого контура має вигляд:

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

Передавальна функція об'єкта управління згідно алгоритмічної схеми (рис. 2.1 б) дорівнює:

$$W_{\text{об}}(p) = \frac{k_{\text{об}}}{(1 + T_{\text{е}}p)(1 + T_{\text{от}}p)},$$

де $k_{\text{об}} = k_{\text{т}}k_{\text{тп}}R_{\Sigma}^{-1}$ – коефіцієнт підсилення об'єкту управління;

$k_{\text{т}}$ – коефіцієнт посилення зворотного зв'язку за струмом

Оскільки

$$W_{\text{от}}(p) = W_{\text{рт}}(p) \cdot W_{\text{об}}(p),$$

то, передавальна функція регулятора струму набуває вигляду:

$$W_{\text{рт}}(p) = \frac{W_{\text{от}}(p)}{W_{\text{об}}(p)} = k_{\text{рт}} \cdot \frac{1 + T_{\text{е}}p}{p}, \quad (2.1)$$

де $k_{\text{рт}} = \frac{R_{\Sigma}}{2T_{\text{от}}k_{\text{тп}}k_{\text{т}}}$ – коефіцієнт посилення регулятора струму.

Реалізація регулятора здійснюється за допомогою типового ПІ-регулятора.

Величину $k_{\text{т}}$ визначаємо з

$$I_{\text{д max}} \cdot k_{\text{т}} = U_{\text{зт max}},$$

$I_{\text{д max}}$ – максимально допустимий струм якоря приймаємо рівним
 $I_{\text{д max}} = 2,5 \cdot I_{\text{н}} = 2,5 \cdot 282 = 705 \text{ A}.$

$U_{\text{зт max}}$ – максимальна напруга завдання, приймемо $U_{\text{зт max}} = 24 \text{ В}$

Отримаємо коефіцієнт підсилення зворотного зв'язку по струму:

$$k_{\tau} = \frac{U_{\text{зт max}}}{I_{\text{д max}}}$$

$$k_{\tau} = \frac{24}{705} = 0,034 \text{ В / А}.$$

У результаті отримуємо повну передавальну функцію замкненого оптимізованого за модульним критерієм контуру регулювання струму:

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

2.6 Синтез послідовного коректуючого пристрою контура ЕРС

Алгоритмічна схема системи із зворотним зв'язком по ЕДС наведена на рис. 2.2.

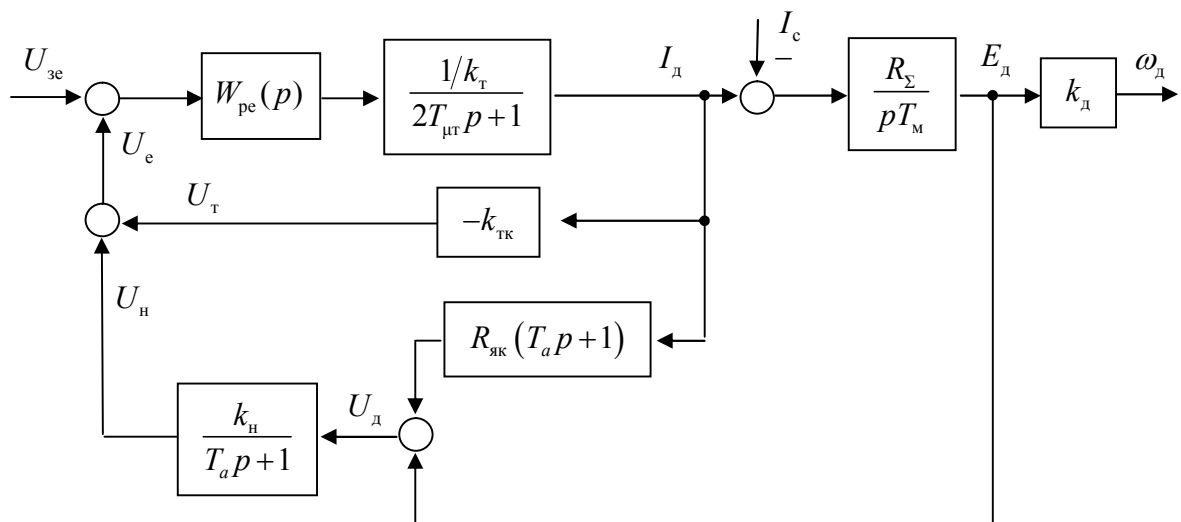


Рисунок 2.2 – Алгоритмічна схема системи керування із зворотним зв'язком за ЕРС

Якщо коефіцієнт підсилення каналу такої компенсації вибрати як

$$k_{\text{тк}} = k_{\text{н}} R_{\text{як}},$$

де $k_{\text{н}}$ – коефіцієнт посилення ланцюга зворотного зв'язку за напругою, то алгоритмічну схему, подану на рис. 2.2, можна перетворити до вигляду, зображеного на рис. 2.3.

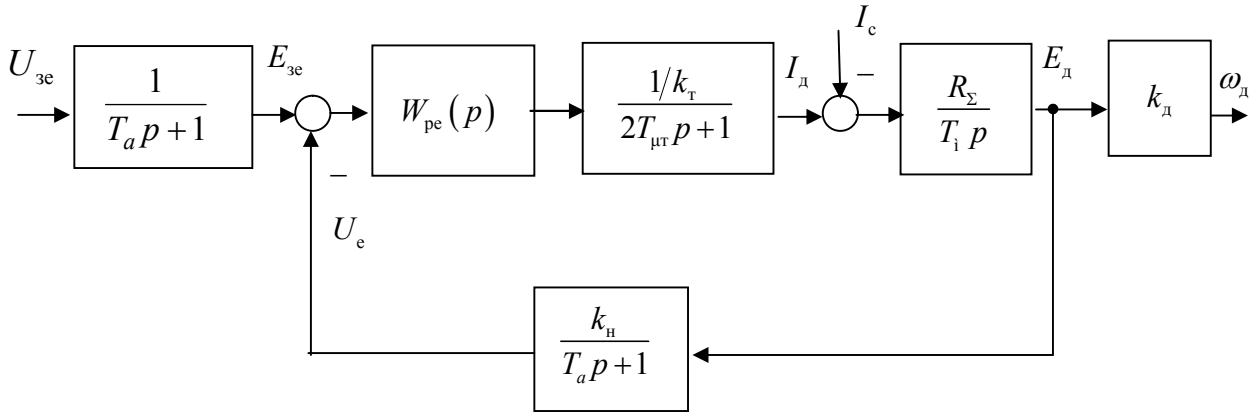


Рисунок 2.3 – Перетворена алгоритмічна схема системи

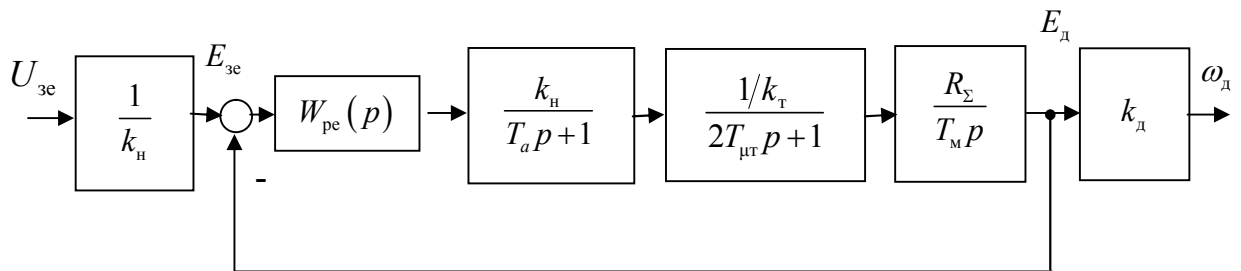


Рисунок 2.4 – Розрахункова алгоритмічна схема системи
з одиничним зворотним зв'язком

У зворотному зв'язку за напругою використовується аперіодична ланка з передавальною функцією $\frac{k_{\text{н}}}{T_a p + 1}$, то для запобігання підвищенню перерегулювання струму при збуреннях за задавальною дією на вхід системи вводиться ланка з аналогічною передавальною функцією. Розрахункова алгоритмічна схема наведена на рис. 2.4.

Оптимізація контура ЕДС здійснюється за симетричним критерієм, при цьому застосовується ПІ-регулятор з передавальною функцією:

$$W_{\text{pe}}(p) = k_{\text{pe}} \frac{4T_{\mu\epsilon}p + 1}{4T_{\mu\epsilon}p},$$

де $k_{\text{pe}} = \frac{T_{\text{м}} k_{\text{т}}}{2k_{\text{н}} T_{\mu\epsilon} R_{\Sigma}}$ – коефіцієнт посилення регулятора ЕРрС

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

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

$$T_{\mu\epsilon} = 2 \cdot 0,005 + 0,083 = 0,093 \text{ с}.$$

Коефіцієнти підсилення зворотного зв'язку по напрузі та струмової компенсації визначаються на основі максимальних значень ЕДС двигуна $E_{\text{дв max}}$ та напруги завдання $U_{\text{зе max}}$, що дозволяє отримати вихідний сигнал 24 В на вході датчика ЕРС.

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

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

$$k_{\text{тк}} = k_{\text{н}} R_{\text{як}},$$

$$k_{\text{тк}} = 0,109 \cdot 0,0484 = 0,0053.$$

На рисунку 2.5 представлена повна алгоритмічна схема системи з урахуванням обох контурів: струмового та ЕРС.

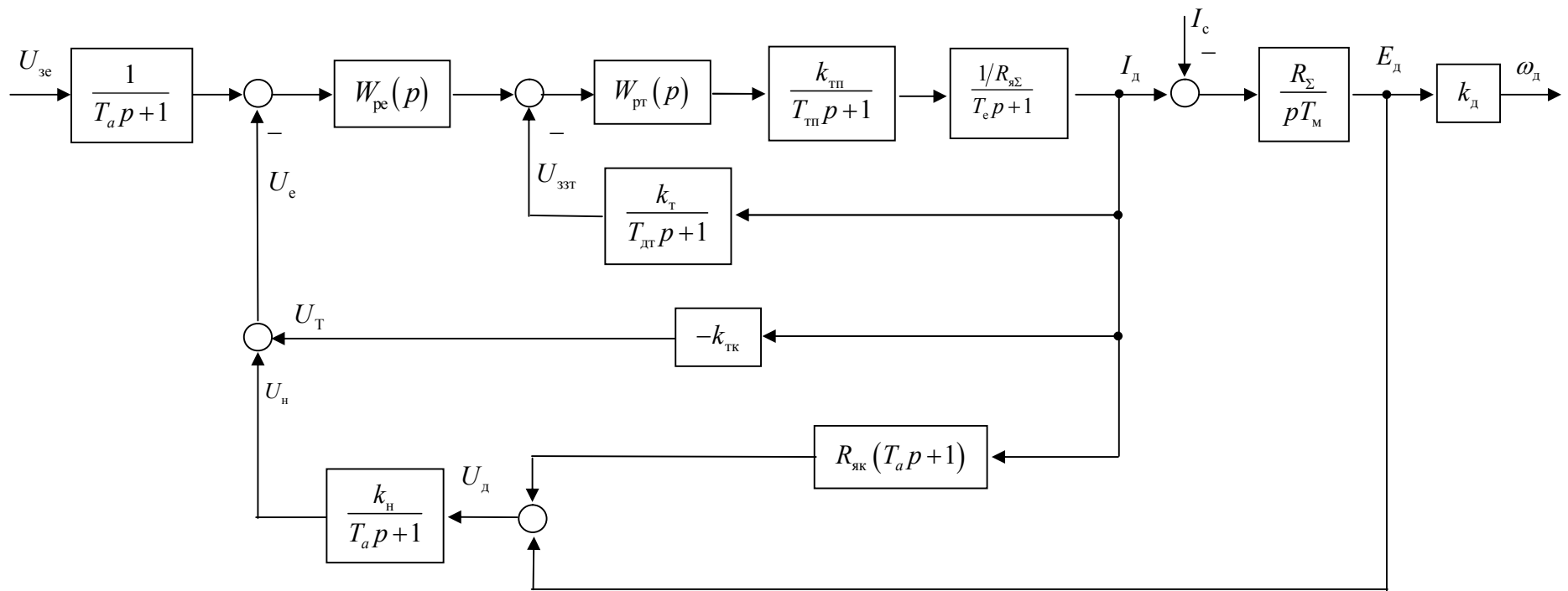


Рисунок 2.5 – Алгоритмічна схема системи підлеглого регулювання швидкості із зворотним зв'язком по ЕРС двигуна

Передавальні функції системи, оптимізованої за симетричним критерієм:

$$W_{\omega_d U_{3e}}(p) = \frac{\omega_d(p)}{U_{3e}(p)} = \frac{k_d}{k_h} \cdot \frac{B(p)}{A(p)},$$

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

$$W_{I_d U_{3e}}(p) = \frac{I_d(p)}{U_{3e}(p)} = \frac{T_m}{k_h R_{\Sigma}} \cdot \frac{pB(p)}{A(p)},$$

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

де $B(p) = 1 + 4T_{\mu e} p$ - поліном чисельника

$A(p) = 8T_{\mu e}^3 p^3 + 8T_{\mu e}^2 p^2 + 4T_{\mu e} p + 1$ - поліном знаменника

Виконаємо перетворення алгоритмічної схеми системи підлеглого регулювання швидкості, яка включає внутрішній контур струму, оптимізований за модульним критерієм, та зовнішній контур ЕРС, налаштований відповідно до симетричного критерію (рис. 2.5). Метою такого перетворення є приведення структури системи до вигляду, наведеного на рис. 2.6.

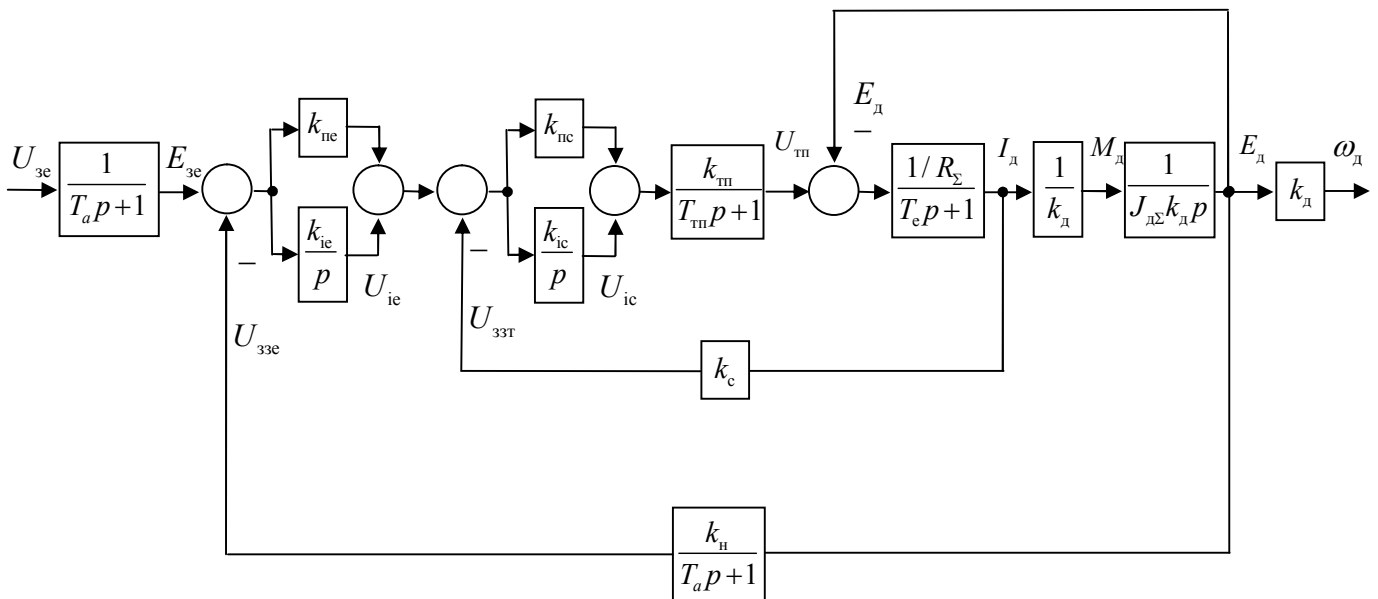


Рисунок 2.6 – Модифіковане алгоритмічне представлення системи

Отримана схема є еквівалентним представленням вихідної системи за умови, що коефіцієнт підсилення в каналі струмової компенсації вибрано як $k_{\text{пе}}$, де $k_{\text{пе}}$ – коефіцієнт передачі зворотного зв'язку по напрузі. Такий вибір забезпечує правильне суміщення каналів і дозволяє звести модель до стандартизованої форми, необхідної для подальшого аналізу та синтезу регуляторів.

У схемі, наведеній на рис. 2.6, використано такі позначення:

- $U_{\text{іе}}$ - сигнал на виході інтегруючої частини регулятора ЕРС;
- $k_{\text{пе}}, k_{\text{іе}}$ - коефіцієнти підсилення пропорційної та інтегруючої складових регулятора ЕРС. Їх значення визначаються з його передавальної функції:

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

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

- $U_{\text{іс}}$ - вихідна напруга інтегруючої складової регулятора струму;
- $k_{\text{пе}}, k_{\text{іс}}$ - коефіцієнти підсилення пропорційної та інтегруючої частин регулятора струму, що визначаються на основі передавальної функції:

$$W_{\text{рс}}(p) = \frac{R_{\Sigma}}{2T_{\text{мс}} k_{\text{тп}} k_{\text{с}}} \cdot \frac{1 + T_{\text{е}} p}{p} = \frac{R_{\Sigma} T_{\text{е}}}{2T_{\text{мс}} k_{\text{тп}} k_{\text{с}}} + \frac{R_{\Sigma}}{2T_{\text{мс}} k_{\text{тп}} k_{\text{с}}} \frac{1}{p} = k_{\text{пе}} + \frac{k_{\text{іс}}}{p},$$

$$k_{\text{пе}} = \frac{R_{\Sigma} T_{\text{е}}}{2T_{\text{мс}} k_{\text{тп}} k_{\text{с}}}, \quad k_{\text{іс}} = \frac{R_{\Sigma}}{2T_{\text{мс}} k_{\text{тп}} k_{\text{с}}}.$$

У передавальній функції електродвигуна замість параметра $T_{\text{м}}$ використано еквівалентне вираження

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

Таким чином у структурі схеми враховано динамічні властивості двигуна та зворотний зв'язок по його ЕРС, що є ключовим для формування адекватного закону регулювання швидкості.

Для подальшого спрощення аналізу алгоритмічна схема може бути приведена до компактнішого структурного вигляду (рис. 2.7). У цьому представленні внутрішній контур струму, оптимізований за модульним оптимумом, розглядається як послідовне з'єднання аперіодичної та інтегруючої ланок, замикає одиначним зворотним зв'язком. При цьому вихідний сигнал першої ланки пропорційний ривку електроприводу P , що дає можливість зручніше аналізувати динамічний відгук системи.

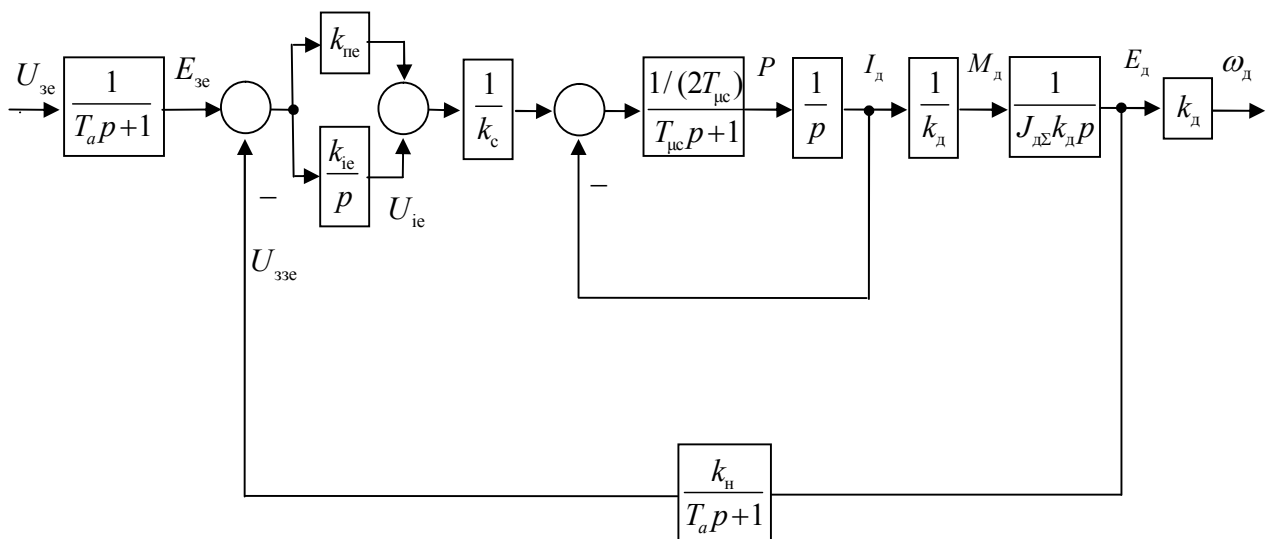


Рисунок 2.7 – Спрощене структурне подання алгоритмічної схеми системи

2.7 Дослідження динамічних характеристик системи

Дослідження перехідних процесів проводилося з використанням MATLAB, який дає можливість моделювати поведінку систем автоматичного керування як у формі передавальних функцій, так і у вигляді векторно-матричних описів. Для більш детального аналізу було створено модель системи, що повністю відповідає структурній схемі на рис. 2.6. Її реалізацію у сере-

довищі SIMULINK наведено на рис. 2.8. Додатково було побудовано спрощений варіант моделі відповідно до алгоритмічної схеми рис. 2.7 (показана на рис. 2.9), що дозволяє швидше оцінювати вплив параметрів регуляторів на динаміку системи.

На рисунках 2.10 та 2.11 наведено графіки перехідних процесів зміни швидкості ω_d та струму I_d двигуна у відповідь на задавальні та збурюючі дії. Налаштування за симетричним оптимумом забезпечує високу швидкодію, проте супроводжується суттєвим перерегулюванням швидкості – близько 43,3%. Для його зменшення доцільним є введення фільтра з постійною часу $4T_{\mu e}$, що дозволяє знизити перерегулювання до 8,1%. У результаті перерегулювання знижується до 8,1%, проте при цьому збільшується час першого узгодження системи. Графіки $\omega_d = f(t)$ і $I_d = f(t)$ з фільтром та без нього позначені як «1» і «2» відповідно.

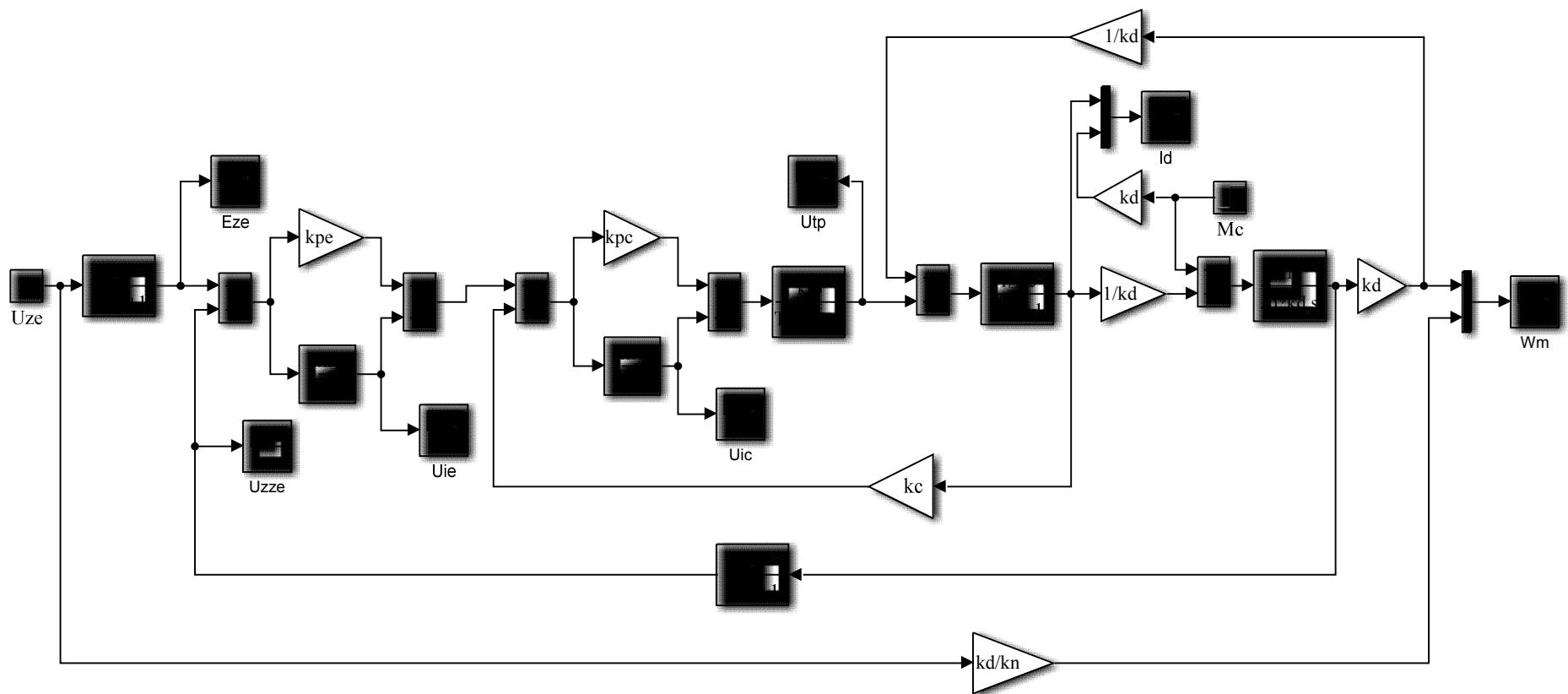


Рисунок 2.8 – Модель системи, реалізована у середовищі MATLAB/SIMULINK

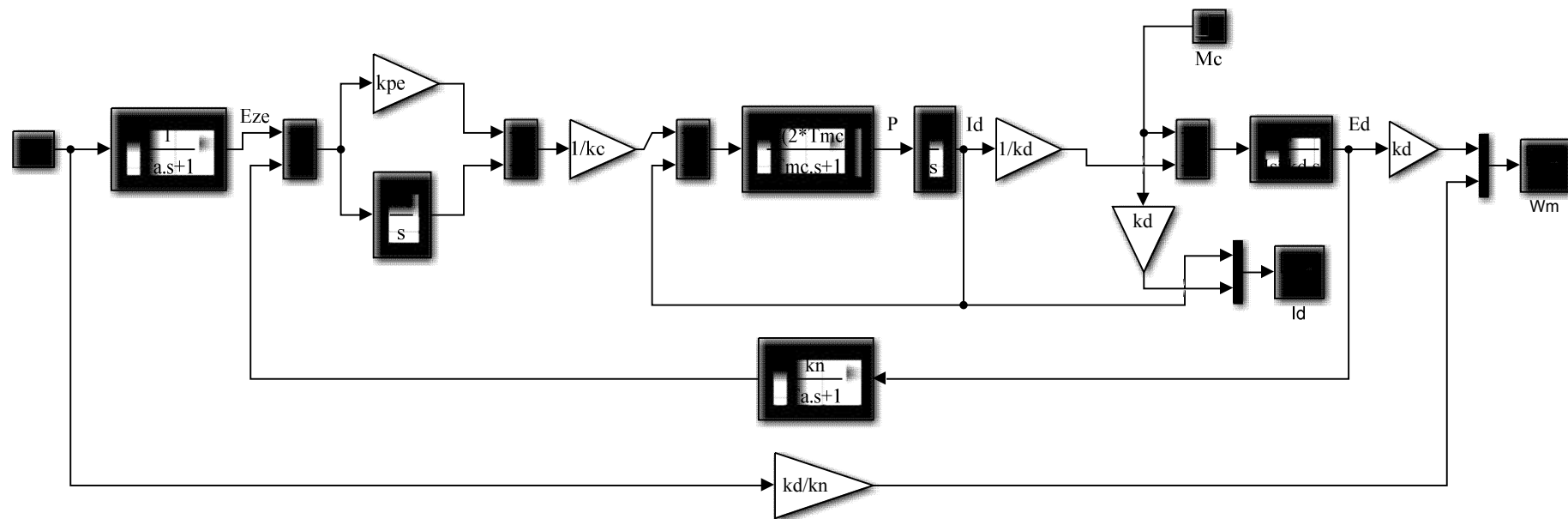


Рисунок 2.9 – Спрощена версія моделі системи у MATLAB/SIMULINK

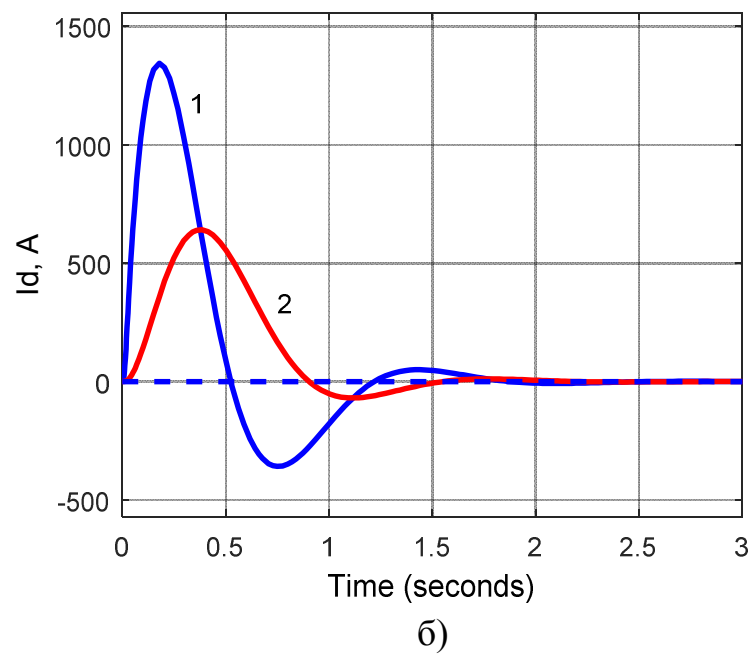
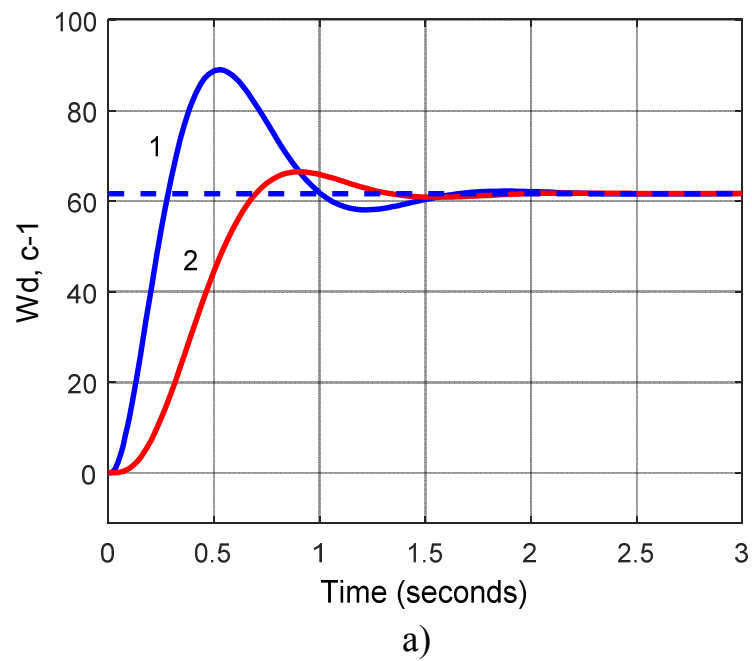


Рисунок 2.10– Перехідні процеси швидкості $\omega_d = f(t)$ а)
і струму $I_d = f(t)$ б) по задаючій дії

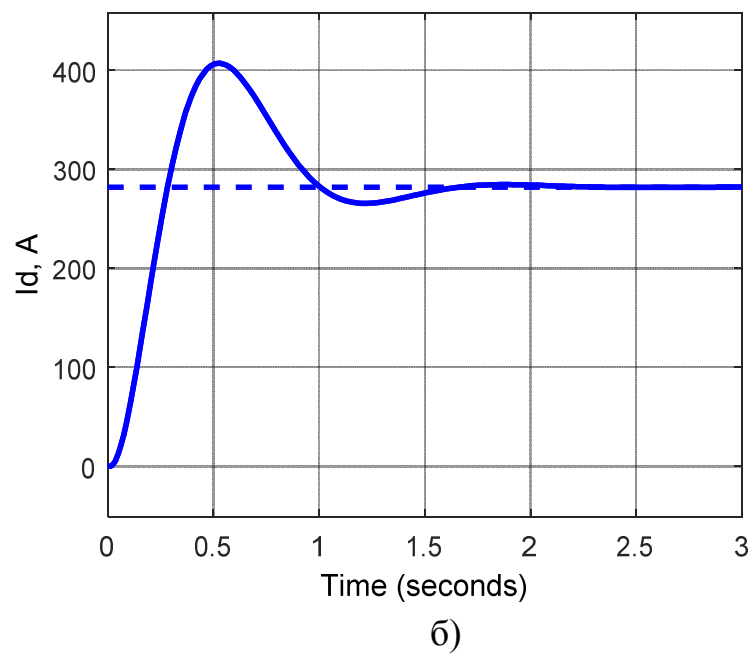
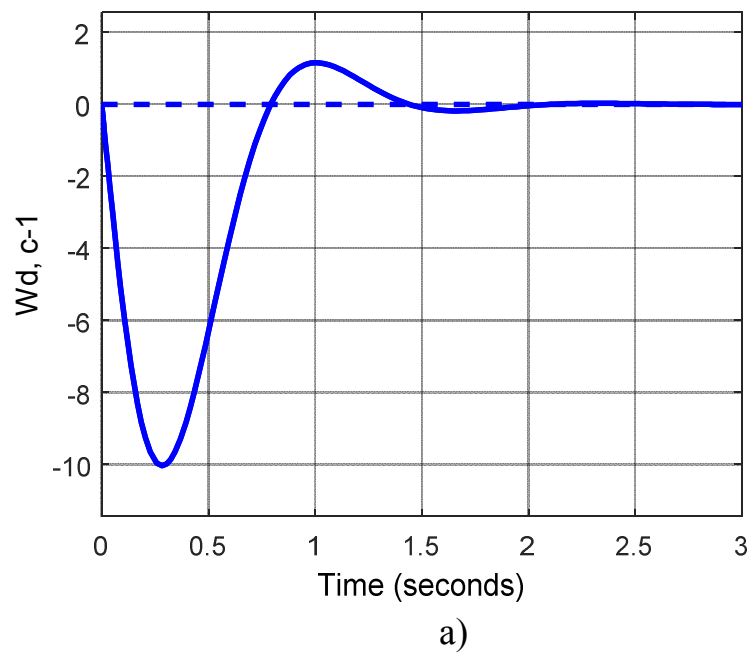


Рисунок 2.11 – Перехідні процеси швидкості $\omega_d = f(t)$ а)
і струму $I_d = f(t)$ б) по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 2

1. На основі аналізу технічних параметрів допоміжних механізмів прокатних станів та умов роботи електроприводів рольгангів обґрунтовано вибір електроприводу постійного струму, реалізованого за схемою ТП–Д. Для забезпечення необхідної керованості та стабільності параметрів електроприводу впроваджено систему підлеглого регулювання, яка найкраще відповідає вимогам даного технологічного обладнання.

2. Виконано перевірку електродвигуна на відповідність тепловим умовам та здатність витримувати перевантаження під час роботи у різних режимах, передбачених технологічним процесом. Розраховане еквівалентне значення моменту свідчить, що як пускові, так і гальмівні моменти забезпечують достатню динамічність перехідних процесів без перевищення допустимих теплових навантажень. Двигун допускає короткочасне перевантаження до , тому за критерієм перевантажувальної здатності він повністю задовольняє вимоги механізму.

3. Проведено обчислення ключових параметрів силового кола електроприводу, які необхідні для подальшого синтезу та налаштування системи автоматичного регулювання. Отримані величини використовуються як вихідні дані під час побудови математичної моделі та алгоритмічних схем системи.

4. Визначено структуру системи автоматичного керування, яка повинна забезпечувати необхідні механічні характеристики електроприводу в усталених режимах та під час перехідних процесів. Для електроприводу рольганга обрано двоконтурну систему підлеглого регулювання: внутрішній контур контролює струм якоря, а зовнішній відповідає за швидкість, при цьому загальна система замкнута по ЕРС двигуна. Такий підхід дозволяє досягти більшої точності, стійкості та придатності до роботи в широкому діапазоні навантажень.

5. Виконано синтез послідовних коригувальних пристроїв для кожного з контурів регулювання. Оскільки динамічний відгук системи визначається співвідношеннями постійних часу її складових, для формування оптимальних перехідних процесів застосовано регулятори, що коригують ці співвідношення. У кон-

турі напруги реалізовано П-регулятор, а для контурів струму та ЕРС – ПІ-регулятори. У результаті контур струму налаштовано відповідно до модульного критерію оптимізації, а контур ЕРС – за симетричним критерієм, що забезпечує баланс швидкодії та стійкості системи.

6. Проведено комп'ютерне моделювання отриманої системи з використанням програмного середовища MATLAB. У середовищі SIMULINK створено структурну модель, яка дозволяє детально аналізувати перехідні процеси при задавальних і збурюючих впливах. Моделювання виконано також на основі передавальних функцій та векторно-матричного опису системи. У процесі налаштування системи за симетричним оптимумом виявлено значне перерегулювання швидкості (43,3%). Для його зниження доцільно застосовувати додатковий фільтр з постійною часу, що дозволяє зменшити перерегулювання до при прийнятному збільшенні тривалості перехідного процесу.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ РОБОТИ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ВЛАСТИВОСТЕЙ МЕХАНІЧНИХ ЕЛЕМЕНТІВ

3.1 Математична модель механічної частини електроприводу з урахуванням пружних зв'язків

У попередніх розділах механічна частина електроприводу розглядалася у спрощеному вигляді – як абсолютно жорстка приведена ланка, в якій деформації механічних елементів не впливали на динаміку системи. Однак у реальних електромеханічних приводах пружність валів, муфт, зубчастих передач та інших елементів суттєво впливає на поведінку приводу, особливо в перехідних режимах.

Насправді кожен з елементів кінематичного ланцюга має кінцеву жорсткість, тому під дією моментів керування або зміни навантаження механічна система поводить себе як пружний об'єкт. У таких системах виникають відносні кутові переміщення та коливання між масами, інерційно пов'язаними між собою. Ці коливання можуть спричинити підвищені динамічні зусилля, що призводить до зниження ресурсу вузлів, погіршення точності регулювання та зростання часу перехідних процесів.

Для коректного описання таких явищ механічну частину електроприводу подають у вигляді системи з декількома масами, що з'єднані між собою пружними та демпфуючими ланками. Найбільш поширеною є двомасова модель (рис. 3.1), яка відображає основні динамічні властивості більш складних структур. У ній одна маса (перший момент інерції) відповідає ротору двигуна та жорстко зв'язаним з ним елементам, а друга – моменту інерції механізму чи робочого органа машини. Між масами вводиться пружний зв'язок із заданою жорсткістю та внутрішнім тертям.

У разі складніших машин із декількома валами та проміжними ланками модель може бути розширена до багатомасової. Такі моделі дають змогу точні-

ше описувати резонансні явища, складні коливальні процеси та взаємодію окремих частин механізму. Однак вони потребують значно більших обчислювальних витрат і тому здебільшого використовуються під час комп'ютерного моделювання.

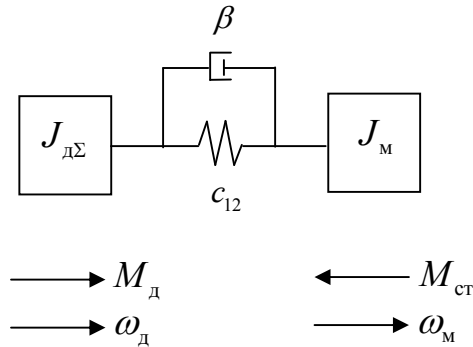


Рисунок 3.1 – Двомасова розрахункова схема

У системах автоматичного керування електроприводами машин прокатного виробництва, зокрема рольгангових механізмів, урахування пружних властивостей є важливою умовою забезпечення надійності й точності роботи. Нехтування пружними зв'язками часто призводить до появи осциляцій під час пуску, реверсу або різких змін навантаження, що, своєю чергою, може викликати коливання швидкості, погіршення якості транспортування заготовок та зростання динамічних навантажень у елементах редуктора чи муфти.

Особливо небезпечними є резонансні режими, коли частота діючих моментів збігається або близька до власних частот системи. У таких режимах амплітуда відносних переміщень між масами різко зростає, що може спричинити аварійні перевантаження, додаткове нагрівання вузлів або навіть їх механічне руйнування. Тому при синтезі системи регулювання важливо визначити власні частоти механічної частини і забезпечити віддалення робочих частот приводу від резонансних зон.

Основні параметри моделі двомасової механічної системи включають:

- приведений момент інерції якоря електродвигуна та жорстко зв'язаних елементів – $J_{дΣ}$;

- момент інерції виконавчого механізму, приведений до вала двигуна – $J_{\text{м}}$;
- еквівалентну жорсткість пружного з'єднання – c_{12} ;
- коефіцієнт внутрішнього в'язкого тертя в пружному елементі – β .

Такий підхід дозволяє побудувати адекватну математичну модель для подальшого аналізу впливу пружних ланок на стабільність, точність і швидкодію системи автоматичного керування електроприводом.

3.2 Рівняння стану одномасової системи

Для подальшого переходу до математичної моделі двомасового електропривода спочатку розглянемо одномасову модель і сформуємо її рівняння стану. Рівняння, що описують динаміку одномасової системи можуть бути записані у вигляді:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dI_{\text{д}}}{dt} = -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma}T_{\text{е}}} U_{\text{тп}}; \\ \frac{dU_{\text{тп}}}{dt} = -\frac{k_{\text{с}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мс}}} I_{\text{д}} - \frac{1}{T_{\text{мс}}} U_{\text{тп}} - \\ \quad + \frac{k_{\text{тп}}}{T_{\text{мс}}} U_{\text{іс}} + \frac{k_{\text{тп}}k_{\text{пт}}}{T_{\text{мс}}} U_{\text{іе}} + \frac{k_{\text{пе}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мс}}} E_{\text{зе}} - \frac{k_{\text{пе}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мс}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іс}}}{dt} = -k_{\text{іс}}k_{\text{с}}I_{\text{д}} + k_{\text{іс}}U_{\text{іе}} + k_{\text{пе}}k_{\text{іс}}E_{\text{зе}} - k_{\text{пе}}k_{\text{іс}}U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}}E_{\text{зе}} - k_{\text{іе}}U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} = -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{k_{\text{д}}T_{\text{а}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{array} \right. \quad (3.1)$$

Для більш компактного й універсального подання перепишемо систему у векторно-матричній формі та доповнимо її рівнянням виходу. Введемо вектор стану $\vec{X}(t)$ розмірності n та вектор керуючих дій $\vec{U}(t)$ розмірності m :

$$\vec{X}(t) = [\omega_d(t), I_d(t), U_{\text{тп}}(t), U_{\text{іс}}(t), U_{\text{іе}}(t), E_{\text{зе}}(t), U_{\text{ззе}}(t)]^T,$$

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

де символ T означає транспонування.

Тоді система рівнянь набуває стандартного вигляду:

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (3.2)$$

де $\mathbf{A}$ – матриця стану системи розмірності $n \times n$;

$\mathbf{B}$ – матриця керування розмірності $n \times m$;

$\mathbf{C}$ – матриця виходу розмірності $r \times n$;

$\vec{Y}(t)$ – вектор вихідних координат системи розмірності r .

Матриці $\mathbf{A}$ і $\mathbf{B}$ для одномасової системи мають вигляд:

$$\mathbf{A} = \begin{array}{c|cccccc} & \omega_d & I_d & U_{\text{тп}} & U_{\text{іс}} & U_{\text{іе}} & E_{\text{зе}} & U_{\text{ззе}} \\ \hline & 0 & \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} & 0 & 0 & 0 & 0 & 0 \\ & -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} & -\frac{1}{T_{\text{е}}} & \frac{1}{R_{\Sigma}T_{\text{е}}} & 0 & 0 & 0 & 0 \\ & 0 & -\frac{k_{\text{с}}k_{\text{пт}}k_{\text{тп}}}{T_{\mu\text{с}}} & -\frac{1}{T_{\mu\text{с}}} & \frac{k_{\text{тп}}}{T_{\mu\text{с}}} & \frac{k_{\text{тп}}k_{\text{пс}}}{T_{\mu\text{с}}} & \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} & -\frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} \\ & 0 & -k_{\text{іс}}k & 0 & 0 & k_{\text{іс}} & k_{\text{пе}}k_{\text{іс}} & -k_{\text{пе}}k_{\text{іс}} \\ & 0 & 0 & 0 & 0 & 0 & k_{\text{іе}} & -k_{\text{іе}} \\ & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & 0 \\ & \frac{k_{\text{н}}}{k_{\text{д}}T_a} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} \end{array} \quad \mathbf{B} = \begin{array}{c|cc} & U_{\text{з}} & M_{\text{ст}} \\ \hline & 0 & -\frac{1}{J_{\text{д}\Sigma}} \\ & 0 & 0 \\ & 0 & 0 \\ & 0 & 0 \\ & 0 & 0 \\ & \frac{1}{T_a} & 0 \\ & 0 & 0 \end{array}$$

Як зазначалося вище, одномасову модель можна замінити спрощеною алгоритмічною схемою, у якій оптимізований за модульним критерієм струмовий контур представлений послідовним з'єднанням аперіодичної та інтегруючої ланок, об'єднаних одиничним зворотним зв'язком. Вихід першої ланки є величиною, пропорційною ривку електропривода P .

У цьому випадку рівняння стану набувають вигляду:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = -\frac{k_{\text{д}}}{2T_{\text{мс}}^2} M_{\text{д}} - \frac{1}{T_{\text{мс}}} P + \frac{k_{\text{пе}}}{2T_{\text{мс}}^2 k_{\text{с}}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\text{мс}}^2 k_{\text{с}}} U_{\text{ззе}} + \frac{1}{2T_{\text{мс}}^2 k_{\text{с}}} U_{\text{іе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} = -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{array} \right. \quad (3.3)$$

Вектор стану $\vec{X}(t)$ та вектор керування $\vec{U}(t)$ для спрощеної моделі:

$$\vec{X}(t) = [\omega_{\text{д}}(t), M_{\text{д}}(t), P(t), U_{\text{іе}}(t), E_{\text{зе}}(t), U_{\text{ззе}}(t)]^T,$$

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

Матриці $\mathbf{A}$ і $\mathbf{B}$ для даної структури також приймають відповідні спрощені форми.

$$\mathbf{A} = \begin{vmatrix} \omega_{\text{д}} & M_{\text{д}} & P & U_{\text{іе}} & E_{\text{зє}} & U_{\text{ззє}} \\ 0 & \frac{1}{J_{\text{д}\Sigma}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\ 0 & -\frac{k_{\text{д}}}{2T_{\text{мс}}^2} & -\frac{1}{T_{\text{мс}}} & \frac{k_{\text{рє}}}{2T_{\text{мс}}^2 k_{\text{с}}} & -\frac{k_{\text{рє}}}{2T_{\text{мс}}^2 k_{\text{с}}} & \frac{1}{2T_{\text{мс}}^2 k_{\text{с}}} \\ 0 & 0 & 0 & 0 & k_{\text{іе}} & -k_{\text{іе}} \\ 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} & 0 \\ \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} & 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} \end{vmatrix} \quad \mathbf{B} = \begin{vmatrix} U_{\text{зє}} & M_{\text{ст}} \\ 0 & -\frac{1}{J_{\text{д}\Sigma}} \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_{\text{а}}} & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}$$

3.3 Математичне моделювання двомасової системи

Механічна частина електропривода, приведена до моделі з двома пружнозв'язаними масами, описується системою рівнянь, у якій враховано як пружність, так і внутрішнє в'язке тертя. Загальний вигляд диференціальних рівнянь такий:

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

Сумарний момент, що передається через пружний елемент M_{Σ} , визначається виразом:

$$M_{\Sigma} = M_{\text{пр}} + \beta_{12}(\omega_{\text{д}} - \omega_{\text{м}}), \quad (3.5)$$

де $M_{\text{пр}}$ – пружний момент, пропорційний кутовій деформації;

$M_{\text{вт}} = \beta_{12}(\omega_{\text{д}} - \omega_{\text{м}})$ – момент в'язкого тертя, пропорційний різниці кутових швидкостей;

β_{12} – коефіцієнт внутрішнього тертя, який у розрахунках приймається рівним нулю.

З урахуванням (3.5), система рівнянь (3.4) може бути подана у формі Коші:

$$\begin{cases} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta_{12}}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta_{12}}{J_{\text{д}\Sigma}} \omega_{\text{м}}. \end{cases} \quad (3.6)$$

Розглянемо роботу системи підлеглого регулювання швидкості для двомасової моделі, в якій струмовий контур налаштовано за модульним критерієм, а контур ЕРС – за симетричним. Відповідна алгоритмічна схема наведена на рис. 3.3. Подібно до випадку з одномасовою моделлю, її можна перетворити у вигляд (рис. 3.4), що зручний для формування рівнянь стану.

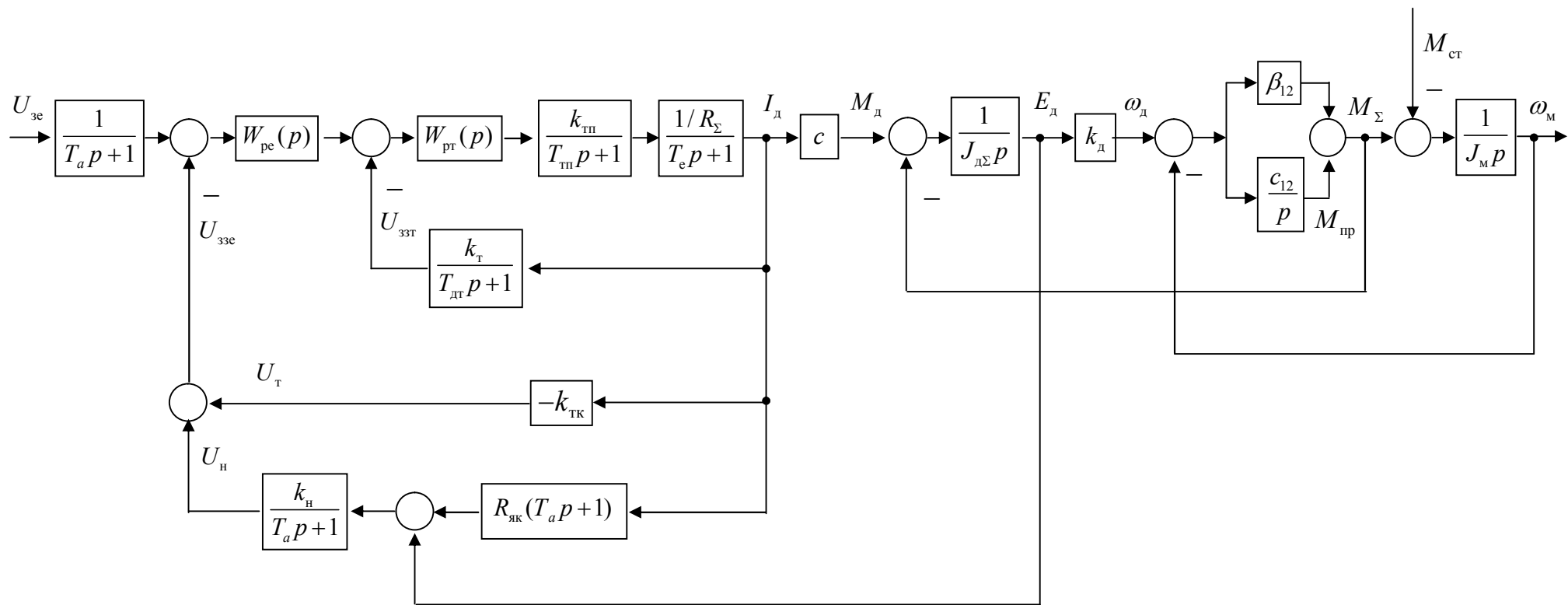


Рисунок 3.3 – Алгоритмічна схема двомасової системи

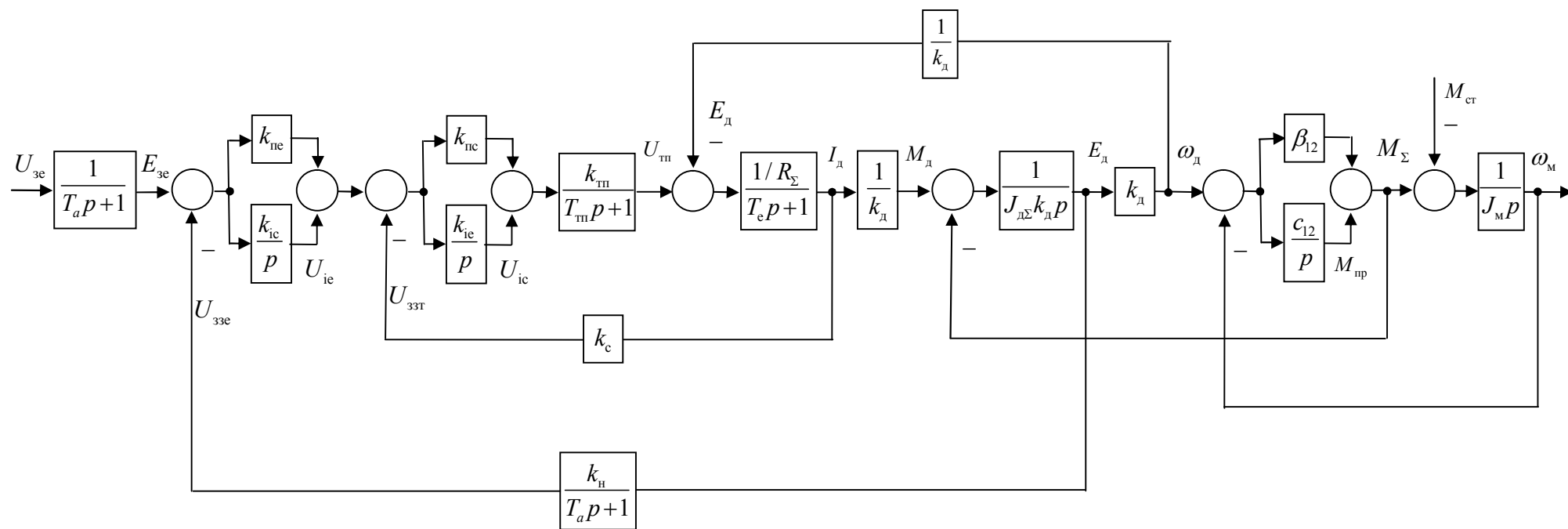


Рисунок 3.4 – Модифікована алгоритмічна схема двомасової системи

Одномасова система, описана рівняннями (3.1) та доповнена рівняннями (3.6), дає змогу сформувати повну систему рівнянь стану двомасової системи:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = -\frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{м}} + \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{д}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta_{12}}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta_{12}}{J_{\text{д}\Sigma}} \omega_{\text{м}}; \\ \frac{dI_{\text{д}}}{dt} = -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma}T_{\text{е}}} U_{\text{тп}}; \\ \frac{dU_{\text{тп}}}{dt} = -\frac{k_{\text{с}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} I_{\text{д}} - \frac{1}{T_{\mu\text{с}}} U_{\text{тп}} - \\ - \frac{k_{\text{тп}}}{T_{\mu\text{с}}} U_{\text{іс}} + \frac{k_{\text{тп}}k_{\text{пс}}}{T_{\mu\text{с}}} U_{\text{іе}} + \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} E_{\text{зе}} - \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іс}}}{dt} = -k_{\text{іс}}k_{\text{с}}I_{\text{д}} + k_{\text{іс}}U_{\text{іе}} + k_{\text{пе}}k_{\text{іс}}E_{\text{зе}} - k_{\text{пе}}k_{\text{іс}}U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}}E_{\text{зе}} - k_{\text{іе}}U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} = -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{k_{\text{д}}T_{\text{а}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{array} \right. \quad (3.7)$$

Перейдемо до векторно-матричної форми запису. Вектори стану $\vec{X}(t)$ та керування $\vec{U}(t)$ визначимо так:

$$\vec{X}(t) = [\omega_{\text{м}}(t), M_{\text{пр}}(t), \omega_{\text{д}}(t), I_{\text{д}}(t), U_{\text{тп}}(t), U_{\text{іс}}(t), U_{\text{іе}}(t), E_{\text{зе}}(t), U_{\text{ззе}}(t)]^T,$$

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

Структурний вигляд матриць **A** і **B** для двомасової системи:

$$\mathbf{A} = \begin{vmatrix} \omega_{\text{м}} & M_{\text{пп}} & \omega_{\text{д}} & I_{\text{д}} & U_{\text{тп}} & U_{\text{іс}} & U_{\text{іе}} & E_{\text{зе}} & U_{\text{ззе}} \\ -\frac{\beta_{12}}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta_{12}}{J_{\text{м}}} & 0 & 0 & 0 & 0 & 0 & 0 \\ -c_{12} & 0 & c_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta_{12}}{J_{\text{д}\Sigma}} & -\frac{1}{J_{\text{д}\Sigma}} & -\frac{\beta_{12}}{J_{\text{д}\Sigma}} & \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} & -\frac{1}{T_{\text{е}}} & \frac{1}{R_{\Sigma}T_{\text{е}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{k_{\text{с}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} & -\frac{1}{T_{\mu\text{с}}} & \frac{k_{\text{тп}}}{T_{\mu\text{с}}} & \frac{k_{\text{тп}}k_{\text{пс}}}{T_{\mu\text{с}}} & \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} & -\frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} \\ 0 & 0 & 0 & -k_{\text{іт}}k_{\text{т}} & 0 & 0 & k_{\text{іт}} & k_{\text{пе}}k_{\text{іт}} & -k_{\text{пе}}k_{\text{іт}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{\text{іе}} & -k_{\text{іе}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} & 0 \\ 0 & 0 & \frac{k_{\text{н}}}{k_{\text{д}}T_{\text{а}}} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} \end{vmatrix}$$

$$\mathbf{B} = \begin{vmatrix} U_{\text{зе}} & M_{\text{ст}} \\ 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_{\text{а}}} & 0 \\ 0 & 0 \end{vmatrix}$$

Так само, як і в одномасовому випадку, двомасову систему можна подати у спрощеній формі (рис. 3.5), використовуючи еквівалентне представлення струмового контуру.

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пп}} + \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пп}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пп}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = -\frac{k_{\text{д}}}{2T_{\mu\text{с}}^2} M_{\text{д}} - \frac{1}{T_{\mu\text{с}}} P + \frac{1}{2T_{\mu\text{с}}^2 k_{\text{с}}} U_{\text{іе}} + \frac{k_{\text{ре}}}{2T_{\mu\text{с}}^2 k_{\text{с}}} E_{\text{зе}} - \frac{k_{\text{ре}}}{2T_{\mu\text{с}}^2 k_{\text{с}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} = -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{array} \right. \quad (3.9)$$

У цьому випадку вектори стану $\vec{X}(t)$ та керування $\vec{U}(t)$ визначаються так:

$$\vec{X}(t) = \left[\omega_{\text{м}}(t), M_{\text{пп}}(t), \omega_{\text{д}}(t), M_{\text{д}}(t), P(t), U_{\text{іе}}(t), E_{\text{зе}}(t), U_{\text{ззе}}(t) \right]^T,$$

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

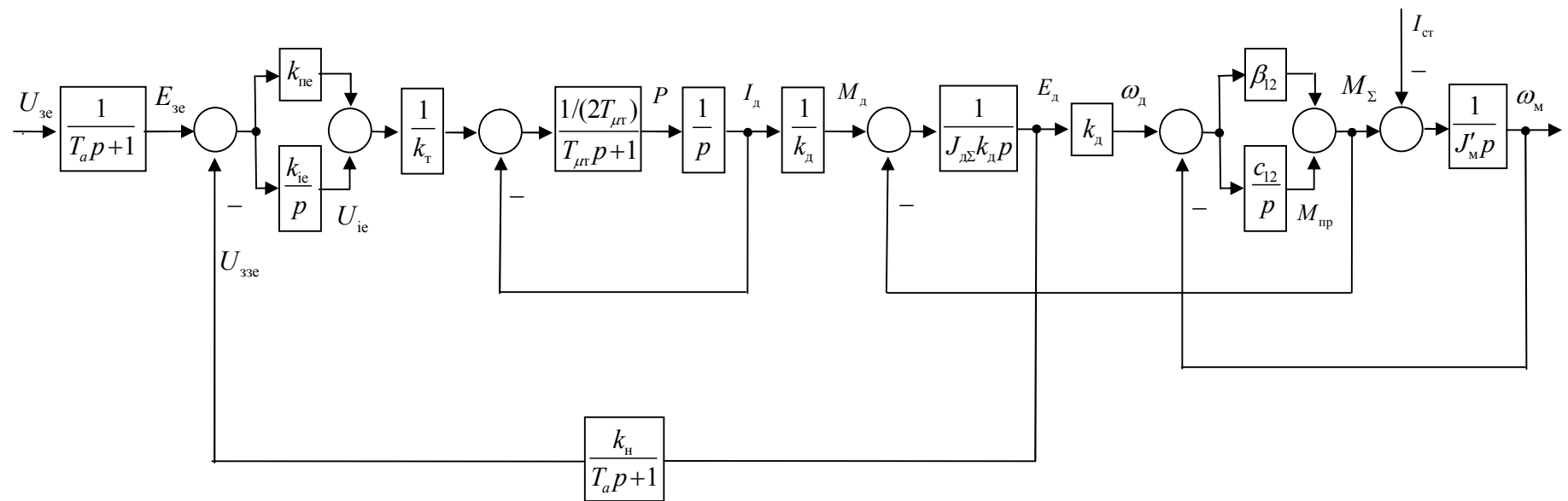


Рисунок 3.5 – Спрощена алгоритмічна схема двомасової системи

Матриці **A** та **B** для спрощеної двомасової моделі.

$$\mathbf{A} = \begin{matrix} & \omega_{\text{м}} & M_{\text{пр}} & \omega_{\text{д}} & M_{\text{д}} & P & U_{\text{іе}} & E_{\text{зє}} & U_{\text{ззє}} \\ \begin{pmatrix} -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 & 0 & 0 & 0 \\ -c & & c & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{k_{\text{д}}}{2T_{\text{мс}}^2} & -\frac{1}{T_{\text{мс}}} & \frac{1}{2T_{\text{мс}}^2 k_{\text{с}}} & \frac{k_{\text{ре}}}{2T_{\text{мт}}^2 k_{\text{с}}} & -\frac{k_{\text{ре}}}{2T_{\text{мс}}^2 k_{\text{с}}} \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{\text{іе}} & -k_{\text{іе}} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} & 0 \\ 0 & 0 & \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} & 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} \end{pmatrix} \end{matrix}$$

$$\mathbf{B} = \begin{matrix} U_{\text{зє}} & M_{\text{ст}} \\ \begin{pmatrix} 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_{\text{а}}} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \end{matrix}$$

3.4 Розрахунок перехідних процесів двомасової системи

Для проведення розрахунку перехідних процесів двомасової системи попередньо необхідно визначити всі параметри, що входять до математичної моделі. Момент інерції електродвигуна разом із жорстко з'єднаними з ним елементами був визначений у попередніх розділах і становить $J_{\text{дΣ}} = 1,1 \cdot J_{\text{дв}} = 1,1 \cdot 7 = 7,7 \text{ кг} \cdot \text{м}^2$. Приведений момент інерції виконавчого механізму (навантаженого рольганга) обчислюється за формулою:

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

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

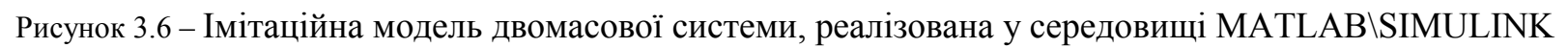
Інші параметри моделі також були встановлені раніше, тому на цьому етапі вони використовуються без додаткових уточнень. Усі необхідні числові значення, що потрібні для моделювання, наведено у відповідних підрозділах.

Аналіз динамічної поведінки двомасової системи здійснювався у програмному середовищі MATLAB. Для цього була розроблена імітаційна модель у SIMULINK, побудована на основі алгоритмічної схеми (рис. 3.4) та системи рівнянь стану (3.7). Структуру створеної моделі наведено на рис. 3.6. На рис. 3.7 подано спрощений варіант моделі, сформований згідно з алгоритмічною схемою, зображеною на рис. 3.5, та рівняннями стану (3.9).

Для моделювання двомасової системи також може застосовуватися схема SIMULINK, що включає блок State-Space (рис. 3.8). У цьому блоці задаються імена матриць стану системи. Кількість вихідних координат (розмірність вектора виходу $\vec{Y}(t)$) задається користувачем і визначає число вікон Score, у яких відображаються графіки перехідних процесів. Вікно введення матриць **A**, **B**, **C** і **D** показане на рис. 3.9; воно відкривається при подвійному натисканні на відповідний блок.

Матриця виходу **C** набуває вигляду одиничної, якщо необхідно спостерігати всі змінні стану. У випадках, коли потрібно аналізувати лише частину змінних, одиниці в матриці **C** розміщуються лише в тих рядках, що відповідають відповідним компонентам вектора виходу $\vec{Y}(t)$. Матриця **D** у цій задачі є нульовою.

Графічні залежності перехідних процесів подано на рис. 3.10–3.13. Для оптимізації виведення результатів на екран ЕОМ та подальшого друку було обрано лише перші чотири змінні стану. Аналіз отриманих графіків свідчить, що реакція системи на задаючу дію (зміну швидкості руху механізму та моменту пружного зв'язку), а також реакція на збурення, має характер слабозатухаючих коливальних процесів. Це узгоджується з фізичними властивостями двомасових систем із пружним зв'язком, у яких природним чином виникають коливання між масами при передаванні моменту.



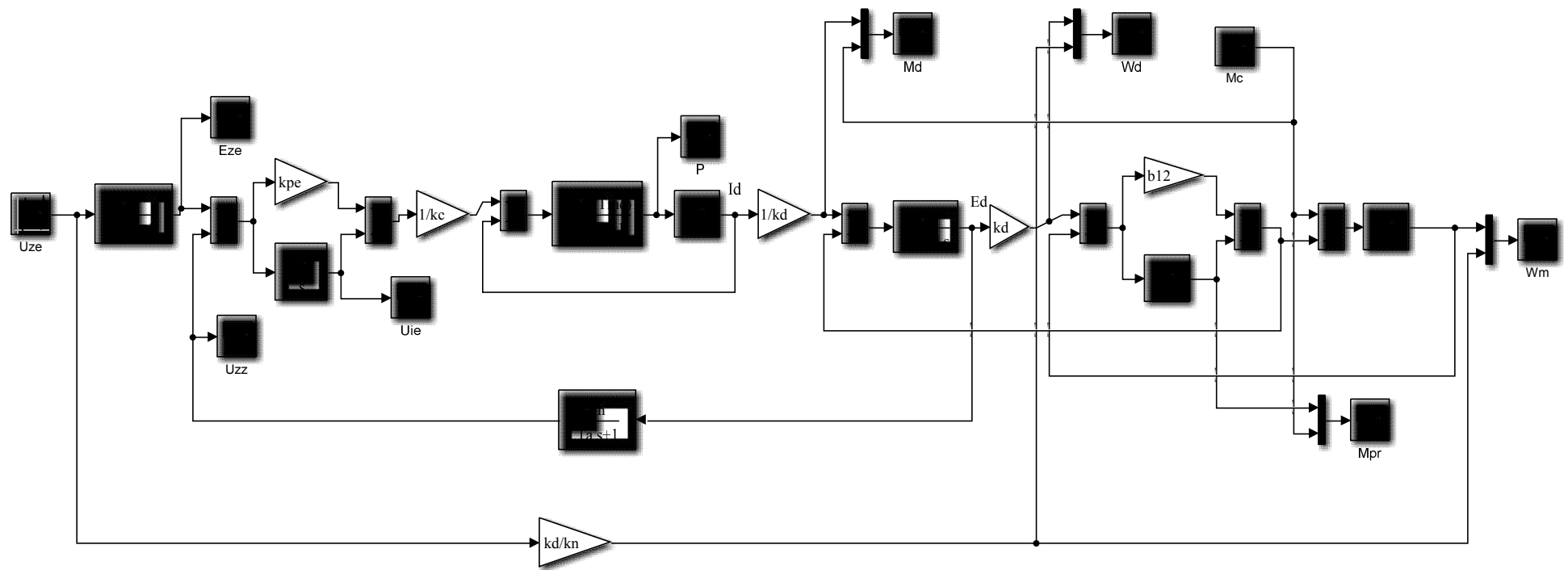


Рисунок 3.7 – Спрощений варіант моделі двомасової системи, сформований у MATLAB\SIMULINK

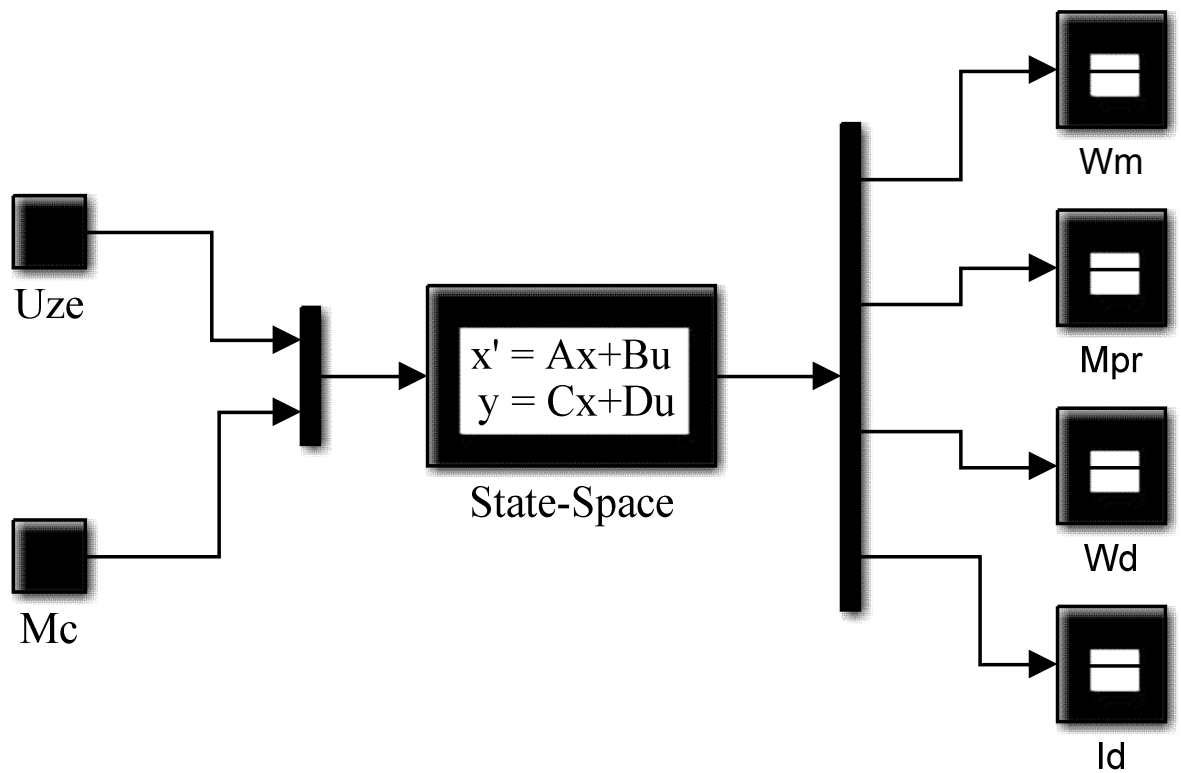


Рисунок 3.8 – Схема моделі системи з використанням блоку State-Space

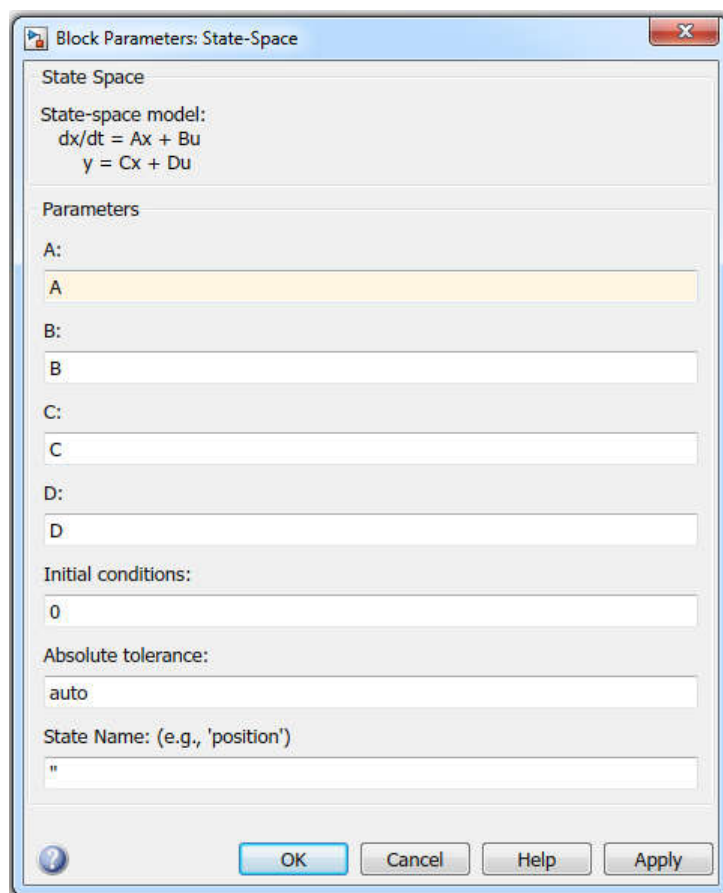
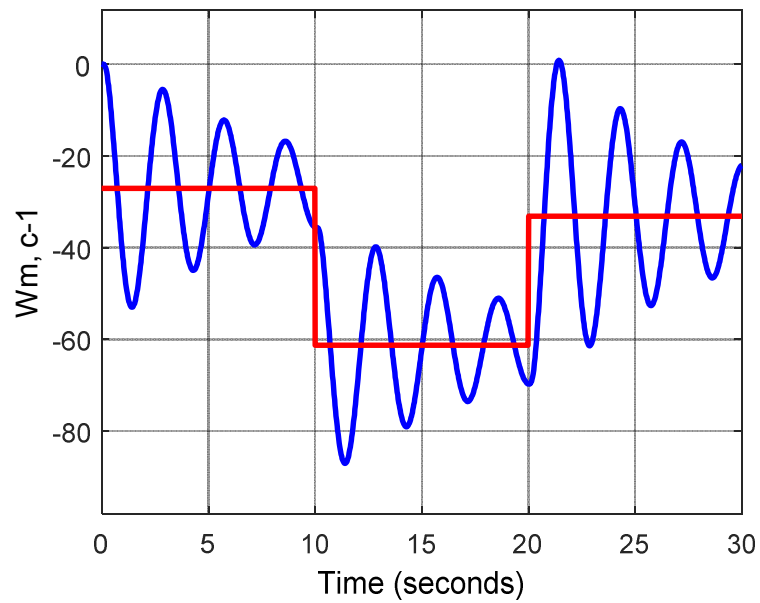
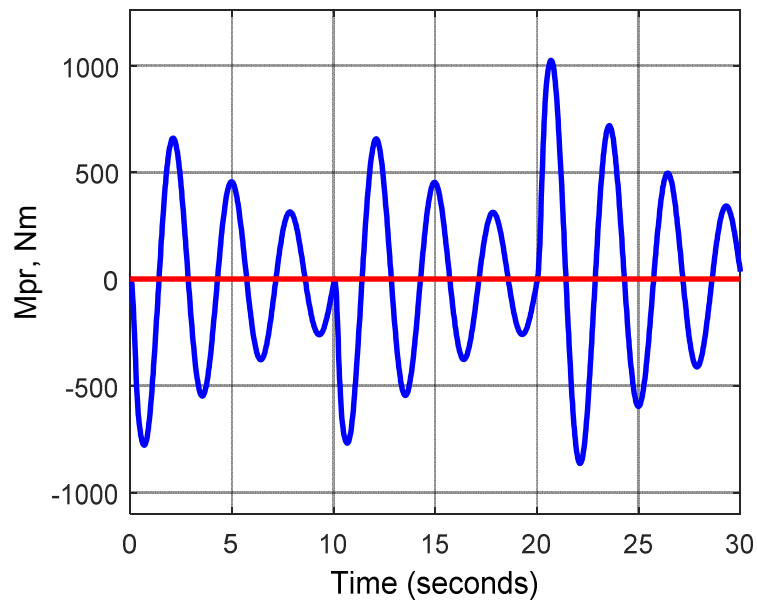


Рисунок 3.9 – Вікно завдання матриць **A**, **B**, **C** та **D** у блоці State-Space

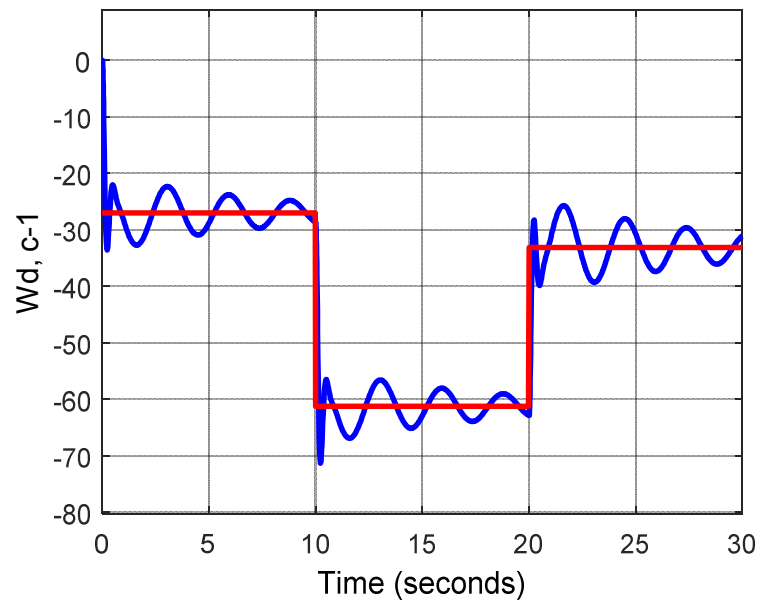


a)

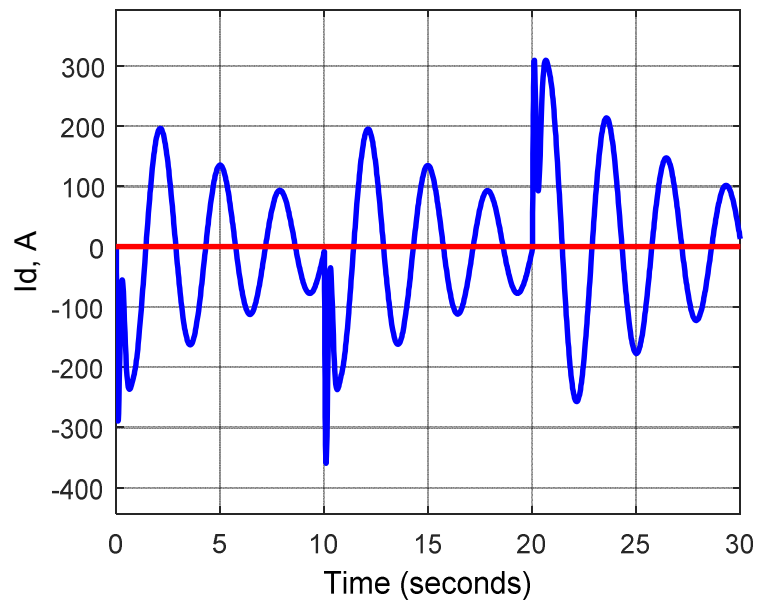


б)

Рисунок 3.5 – Перехідні процеси швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи при задаючому впливі



a)



б)

Рисунок 3.6 – Перехідні процеси швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи при задаючому впливі

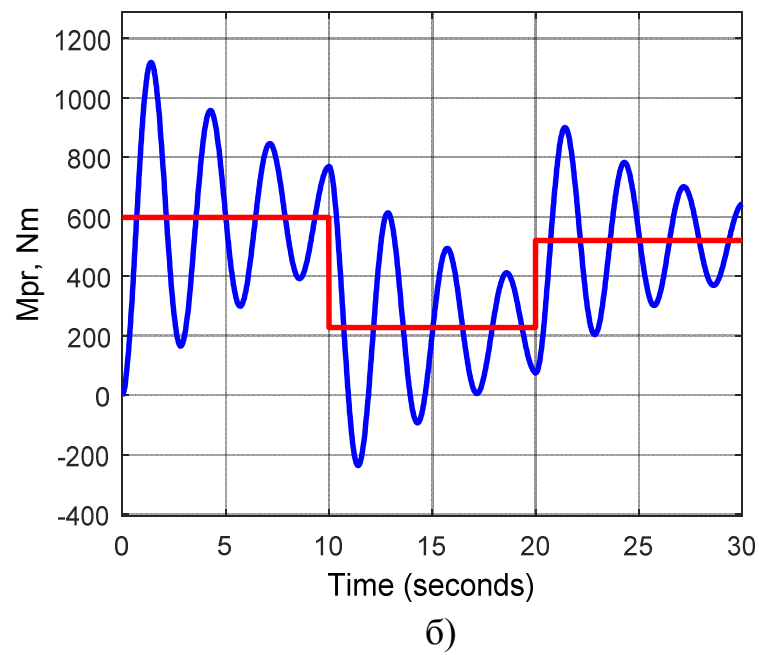
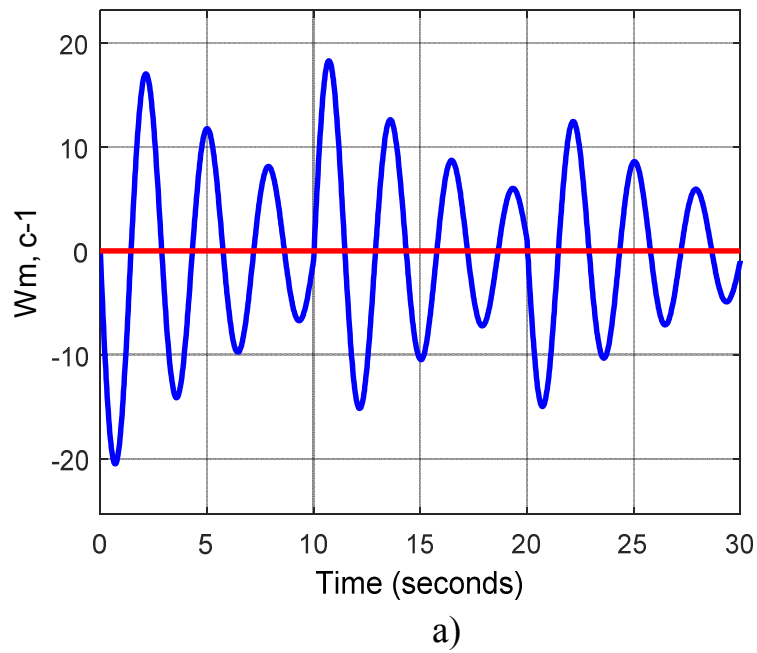


Рисунок 3.7 – Перехідні процеси швидкості механізму $\omega_m = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи при збурюючому впливі

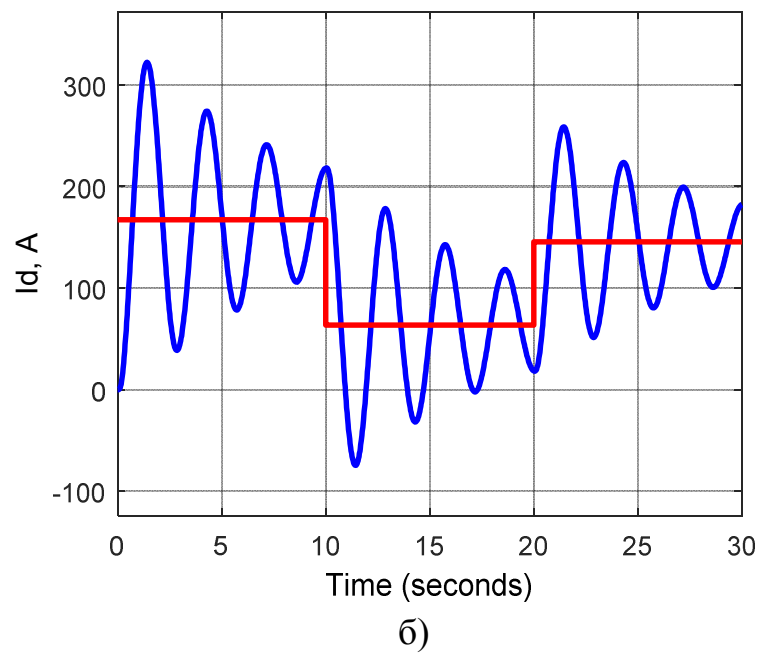
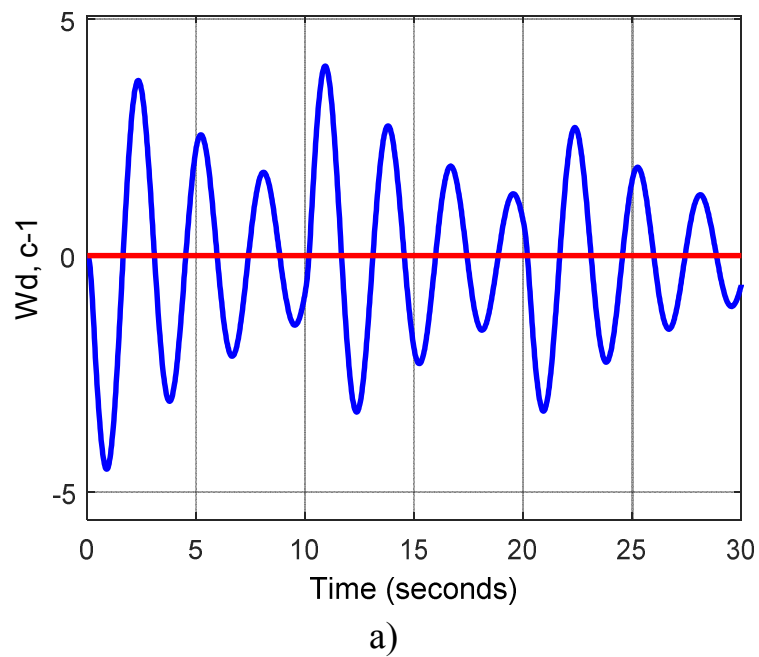


Рисунок 3.8 – Перехідні процеси швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи при збурюючому впливі

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму рольганга з урахуванням кінцевої жорсткості механічного зв'язку між електродвигуном і рольгангом. Показано, що моделювання механічної частини у вигляді жорсткої приведеної ланки лише наближено відображає реальну динаміку, оскільки не враховує вплив пружних властивостей елементів. Наявність кінцевої жорсткості з'єднань перетворює механічну частину приводу на пружну двомасову систему, у якій як керувальні дії, так і збурення (зокрема момент статичного навантаження) спричиняють коливання взаємозв'язаних мас. Це призводить до зростання динамічних навантажень у з'єднаннях і знижує точність реалізації заданих рухів. Установлено, що кінематична схема механізму рольганга з урахуванням податливості елементів може бути адекватно описана двомасовою моделлю.

2. Розроблено математичну модель двомасової системи. Отримано рівняння стану у диференціальній та векторно-матричній формах, визначено складові вектора стану, тобто основні змінні, що описують динаміку системи. Сформовано вектор зовнішніх впливів і побудовано матриці стану **A** та керування **B**. Розроблено повну алгоритмічну схему моделі, а також її спрощений варіант для системи з оптимізованим за модульним критерієм контуром струму.

3. Виконано комп'ютерне моделювання двомасової системи в середовищі MATLAB з використанням пакету SIMULINK. Побудовано структурну модель об'єкта, на вхід якої подавався тестовий ступінчастий сигнал випадкової амплітуди в межах ± 24 В, що відповідає зміні швидкості в діапазоні номінальних значень. Аналіз отриманих графіків показав, що перехідні процеси змінних стану мають виражений коливальний і недостатньо стабілізований характер, що свідчить про потребу подальшої оптимізації системи керування для підвищення її динамічних показників.

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

4.1 Регулятори, реалізовані засобами пакета **Neural Network Toolbox** у середовищі **MATLAB**

Процес побудови нейромережевої системи управління виконувався за допомогою інструментів пакета **Neural Network Toolbox**, що входить до складу **MATLAB**. Далі наведено короткий огляд можливостей цього пакета, описано послідовність створення нейронного регулятора та подано результати моделювання роботи нейромережевої системи керування.

У **MATLAB** розглянуто три основні архітектури нейронних мереж, реалізованих у зазначеному пакеті прикладних програм, які використовуються при побудові таких типів регуляторів:

- регулятор із прогнозуванням майбутніх станів системи (**NN Predictive Controller**);
- регулятор, що ґрунтується на моделі авторегресії з ковзним середнім (**NARMA-L2 Controller**);
- регулятор на основі еталонної моделі (**Model Reference Controller**).

Використання нейронних мереж у задачах автоматичного керування передбачає поділ процесу проектування на два основних етапи:

- етап ідентифікації моделі об'єкта;
- етап синтезу закону керування.

Під час ідентифікації будується нейромережева модель об'єкта керування, яка надалі використовується для розробки потрібного регулятора. Для всіх трьох згаданих архітектур застосовується однакова процедура побудови моделі, однак принципи синтезу регуляторів між собою істотно відрізняються.

У випадку регулятора, що працює з прогнозом, нейронна модель об'єкта призначена для оцінювання його майбутніх станів, а оптимізаційний алгоритм підбирає таке керуюче діяння, яке мінімізує відхилення між бажаною та реальною траєкторіями вихідної величини.

Для регулятора типу NARMA-L2 структура керування являє собою модифіковану версію моделі самого об'єкта і будується на основі авторегресії з ковзним середнім.

У випадку використання еталонної моделі регулятор являє собою нейронну мережу, що навчається таким чином, щоб змусити реальний об'єкт повторювати поведінку вибраної еталонної системи. При цьому модель об'єкта задіюється безпосередньо в процесі налаштування параметрів регулятора.

Динамічні моделі систем керування з нейромережевими регуляторами зібрані в окремому підрозділі *Control Systems* бібліотеки Neural Network Toolbox (див. рис. 4.1).

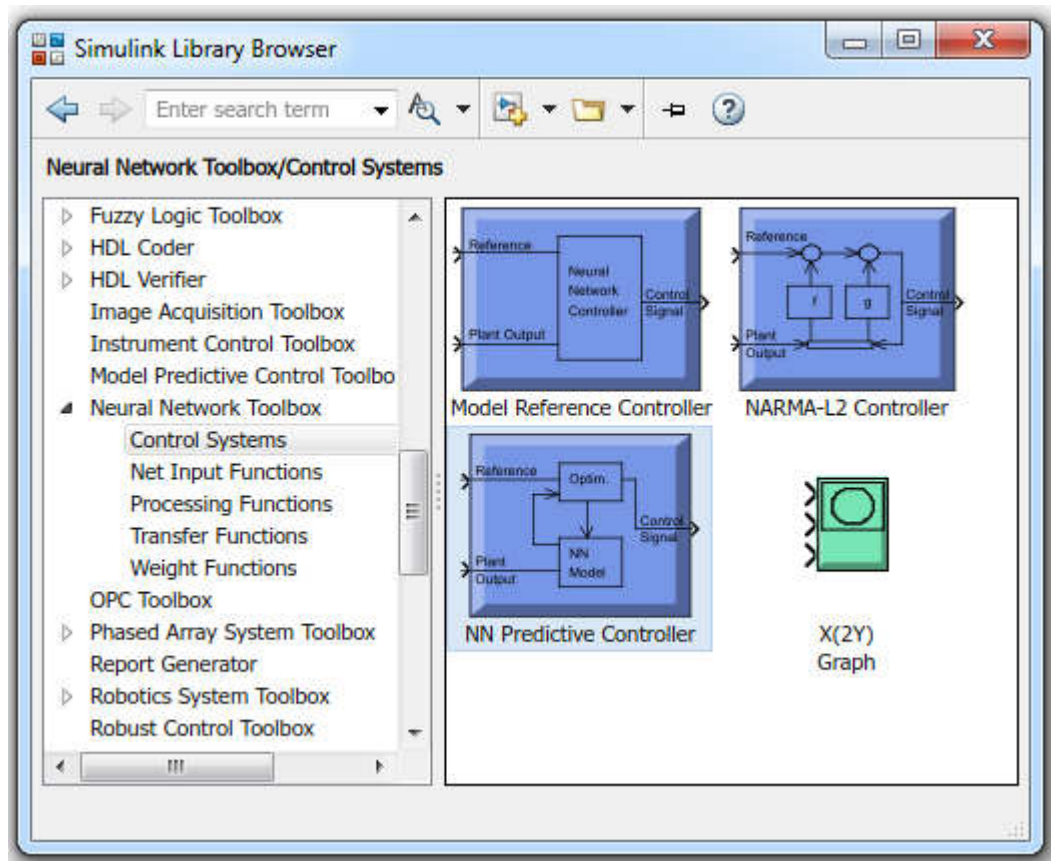


Рисунок 4.1 – Набір блоків Neural Network Blocksets підсистеми Control Systems середовища SIMULINK

Регулятор із прогнозуванням. У цьому типі регулятора нейромережева модель об'єкта використовується для прогнозу того, як система поводитиметься при різних можливих керуючих впливах. На основі цих прогнозів оптиміза-

ційний алгоритм вибирає такі сигнали керування, які забезпечують мінімальну похибку між заданою та фактичною траєкторією вихідного сигналу моделі. Формування нейромережевої моделі здійснюється окремо: мережу навчають пакетно, застосовуючи один із доступних навчальних алгоритмів. Недоліком цього підходу є значні обчислювальні витрати, оскільки оптимізація проводиться на кожному кроці формування керуючої дії.

Регулятор NARMA-L2 (на основі авторегресійної моделі з ковзним середнім). Серед трьох архітектур цей варіант є найбільш економним з точки зору обчислювальних ресурсів. По суті, регулятор є модифікацією отриманої на попередньому етапі нейромережевої моделі об'єкта. Основним недоліком методу є необхідність подавати модель у формі простору станів у канонічному вигляді зі строго визначеною структурою супровідної матриці. Це може спричинювати додаткові неточності в обчисленнях.

Регулятор з еталонною моделлю. Обсяг обчислень, необхідний для такого регулятора, приблизно відповідає обсягу, потрібному для NARMA-L2. Однак структура цього регулятора складніша: потрібно одночасно навчити нейромережу, що відтворює поведінку об'єкта, і нейромережу, яка виконує функції регулятора. Процес навчання складніший, оскільки використовує динамічний варіант алгоритму зворотного поширення помилки. Головною перевагою підходу є його універсальність – регулятор на базі еталонної моделі можна адаптувати до багатьох типів об'єктів керування.

Для електроприводу, що аналізується у цій роботі, було обрано саме регулятор із прогнозуванням. Варіанти на основі NARMA-L2 та еталонної моделі не забезпечили достатнього рівня якості керува

4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller

Регулятор типу NN Predictive Controller, реалізований у складі пакету Neural Network Toolbox системи MATLAB, ґрунтується на використанні ней-

ромережевої моделі нелінійного об'єкта керування. Така модель забезпечує можливість передбачити подальший розвиток процесів у системі, а також дозволяє сформувати оптимальний керуючий вплив на визначеному інтервалі прогнозування.

4.2.1 Процедура ідентифікації об'єкта керування

Схему модуля ідентифікації показано на рис. 4.2. До складу цього модуля входить нейромережева модель об'єкта, яка навчається у відокремленому (автономному) режимі. Метою навчання є мінімізація розбіжності між реакцією реального об'єкта та відповіддю моделі на подану послідовність тестових впливів.

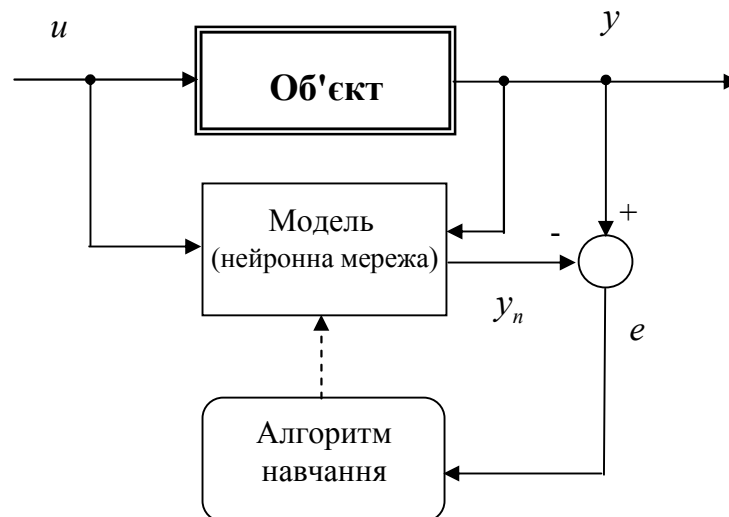


Рисунок 4.2 – Структурна схема підсистеми ідентифікації

Архітектуру нейромережі, що використовується для моделювання об'єкта, наведено на рис. 4.3. Вона є дворівневою (двошаровою) і включає лінії затримки, які дозволяють зберігати попередні значення сигналів на вході та виході об'єкта. Це забезпечує можливість прогнозувати майбутні значення вихідної координати.

Параметри мережі налаштовуються автономно, на основі набору експериментальних даних, отриманих у процесі тестування системи. Для навчання може застосовуватися будь-який зі стандартних алгоритмів нейромережевого

навчання. У даному дослідженні використано метод Левенберга–Марквардта, який в Neural Network Toolbox реалізований за допомогою М-функції **trainlm**.

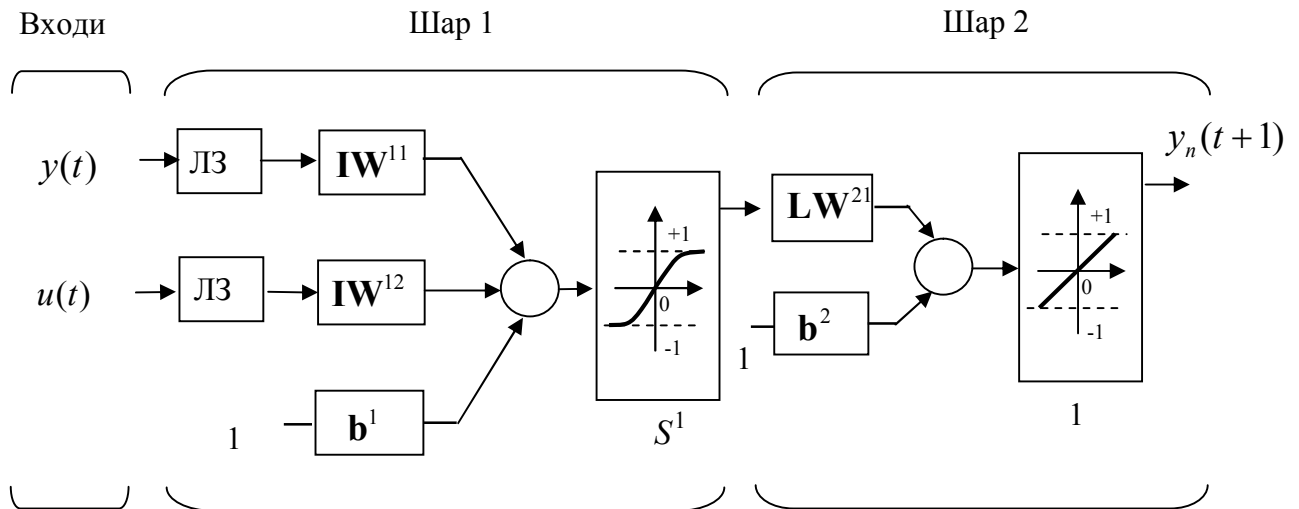


Рисунок 4.3 – Структура нейронної мережі, що моделює об’єкт керування

4.2.2 Принцип функціонування прогнозуючої системи керування

Алгоритм управління на основі прогнозу працює за ідеологією «ковзного горизонту». Це означає, що нейромережева модель об’єкта безперервно обчислює ймовірну реакцію системи на деякому майбутньому відрізку часу. Отриманий прогноз передається в оптимізаційний модуль, який обирає таку керуючу дію, що мінімізує критерій якості:

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

Тут коефіцієнти N_1 , N_2 і N_4 визначають межі інтервалу, на якому аналізуються відхилення вихідного сигналу та енергетичні витрати на формування управління. Змінна u' – пробний керуючий вплив, y_r – бажана траєкторія, а y_m – прогнозована реакція моделі. Параметр ρ регулює вагу енергетичної складової у функції якості.

Узагальнену блок-схему системи із застосуванням прогнозуючого принципу наведено на рис. 4.4. Вона складається з нейронної моделі об’єкта та оп-

тимізаційного модуля. Останній знаходить значення u' , які мінімізують критерій якості, а сформований сигнал керування надходить безпосередньо на об'єкт.

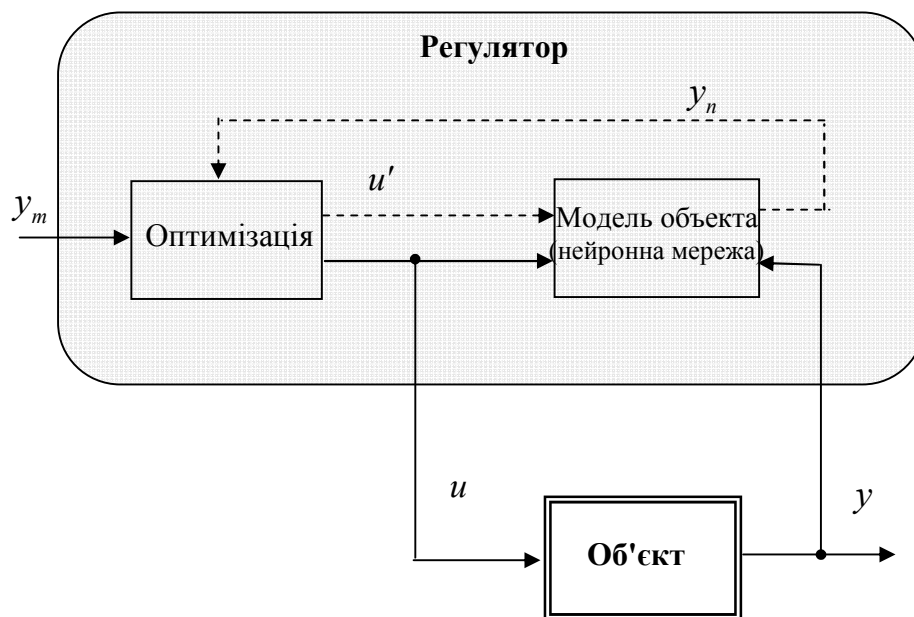


Рисунок 4.4 – Узагальнена схема керування з використанням прогнозуючого регулятора

4.3 Проектування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB

Під час побудови нейромережевого регулятора типу NN Predictive Controller використовується набір спеціалізованих функцій і моделей, що входять до складу каталогу *toolbox/nnet/nnetcontrol* пакету Simulink. До цього набору належать такі групи програмних засобів:

1. Функції одновимірної оптимізації

Ці алгоритми застосовуються під час пошуку оптимального керуючого впливу в процесі роботи регулятора:

- **Csrchbac** – алгоритм пошуку за схемою «зворотного проходу» (backtracking).
- **Csrchbre** – модифікований метод Брента, який поєднує техніку золотого перетину та квадратичну інтерполяцію.

- **Csrchcha** – метод кубічної інтерполяції Чараламбуса.
- **Csrchgol** – класичний метод золотого перетину.
- **Csrchhyb** – комбінований алгоритм, що використовує бісекцію та кубічну інтерполяцію.

2. Функції, що реалізують механізм керування з прогнозуванням

Ці М-функції забезпечують обчислення критеріїв якості та роботу оптимізаційних алгоритмів:

- **Calcjddjj** – оцінювання критерію якості та його градієнта;
- **Predopt** – основний модуль оптимізації прогнозуючого регулятора;
- **Dyduvar** – розрахунок часткових похідних вихідних сигналів за входами.

3. Моделі та графічні додатки Simulink

- **Predcstr** – графічний інтерфейс користувача (GUI) для роботи з регулятором типу Predictive Controller;
- **Prest3sim2** – нейромережева модель об'єкта керування, що застосовується функцією *deport* для формування прогнозу поведінки системи.

4. Допоміжні програмні модулі

- **Nncontrolutil** – набір службових функцій, що забезпечують взаємодію регулятора з внутрішніми компонентами Simulink;
- **Nnident.m** – ключова функція, яка використовується під час етапу ідентифікації об'єкта. Розташована у підкаталозі *private*, забезпечує створення навчальної вибірки, формування нейронної мережі та процес її навчання через зручний GUI.

Після підготовки необхідних даних у середовищі Simulink формується модель системи керування, структура якої наведена на рисунку 4.5.

У представленій конфігурації системи міститься підсистема, що моделює об'єкт керування (блок *Subsystem*), модуль нейромережевого регулятора (*NN Predictive Controller*), генератор випадкового еталонного ступінчастого сигналу (*Random Reference*), а також елементи для візуалізації результатів.

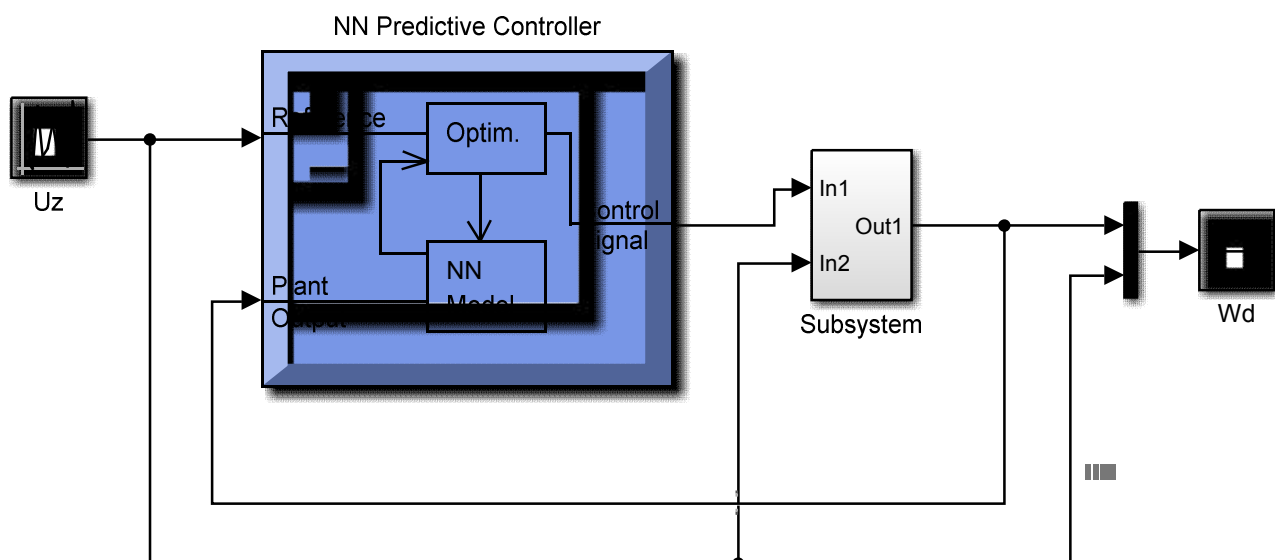


Рисунок 4.5 – Структурна модель системи керування з нейрорегулятором NN Predictive Controller

Важливо зазначити, що дана схема одночасно виконує роль стандартної моделі середовища SIMULINK і працює як графічний інтерфейс користувача (GUI), забезпечуючи безпосередню взаємодію з налаштуваннями системи.

Структура внутрішньої організації нейромережевого регулятора NN

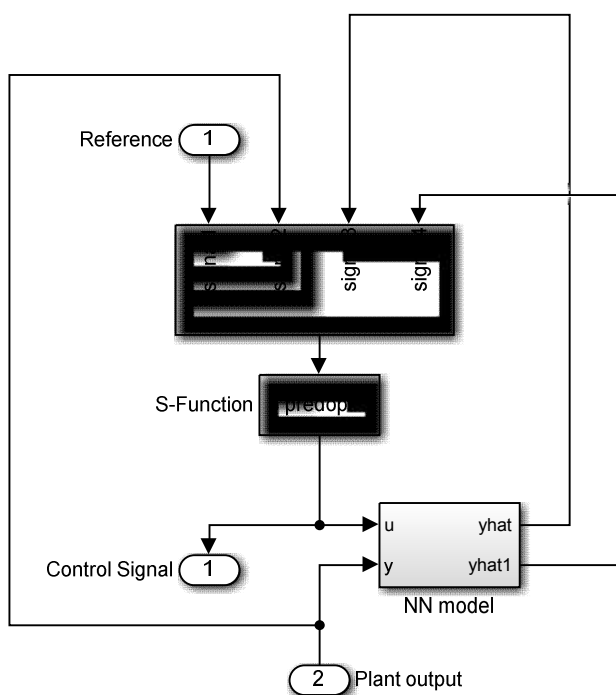


Рисунок 4.6 – Структурна схема нейрорегулятора

Predictive Controller подана на рис. 4.6. Вона відкривається у спеціальному вікні після вибору команди **Look Under Mask** для відповідного блоку.

Розробка регулятора розпочинається з активації блоку NN Predictive Controller. Після цього відкривається інтерфейс, зображений на рис. 4.7, який слугує панеллю налаштування параметрів. У верхній частині вікна відображається інформаційне повідомлення *«Perform plant*

identification before controller configuration», яке підказує, що перед подальшими налаштуваннями обов'язково потрібно провести етап ідентифікації – сформува-ти нейромережеву модель керованого об'єкта за допомогою процедури **Plant Identification**.

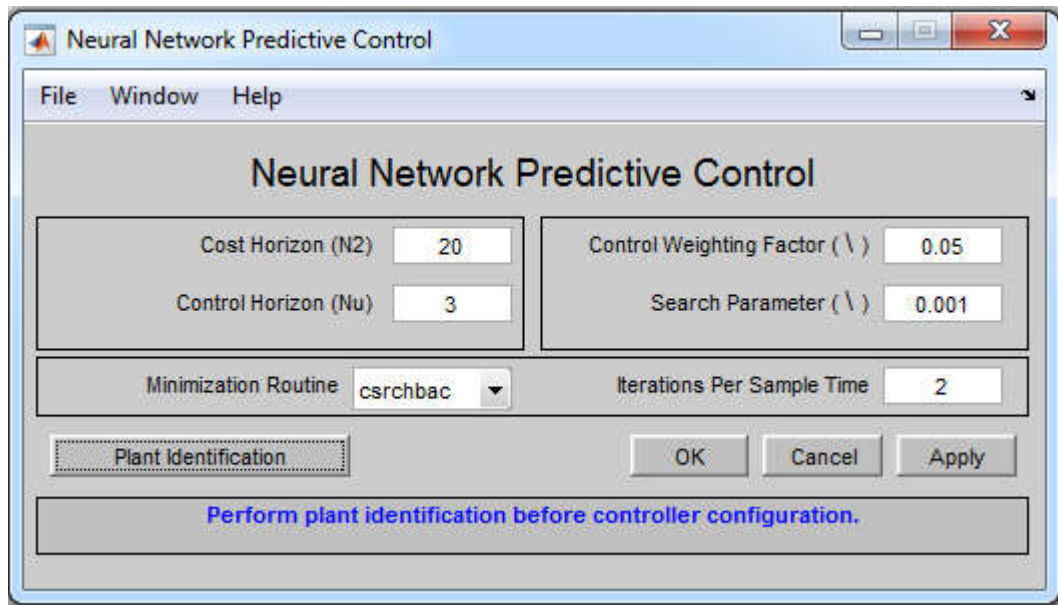


Рисунок 4.7 – Інтерфейс налаштування параметрів регулятора

На рис. 4.8 показано вікно модуля Plant Identification. Воно є універсальним інструментом, що дозволяє створювати нейромережеві моделі для будь-яких динамічних систем, описаних у SIMULINK. Для даної роботи таким об'єктом виступає модель двомасового механізму.

Під час ідентифікації формується нейронна мережа, яка повинна відтворювати поведінку реального об'єкта, оскільки саме ця модель буде використана в подальшому для налаштування та оптимізації регулятора. Тому етап створення моделі має виконуватися до розрахунків самого регулятора.

Процедура ідентифікації вимагає попереднього задання низки параметрів, які налаштовуються через вікно **Plant Identification**. Це вікно поділене на кілька логічних секцій, кожна з яких відповідає за певний аспект формування нейромережевої моделі.

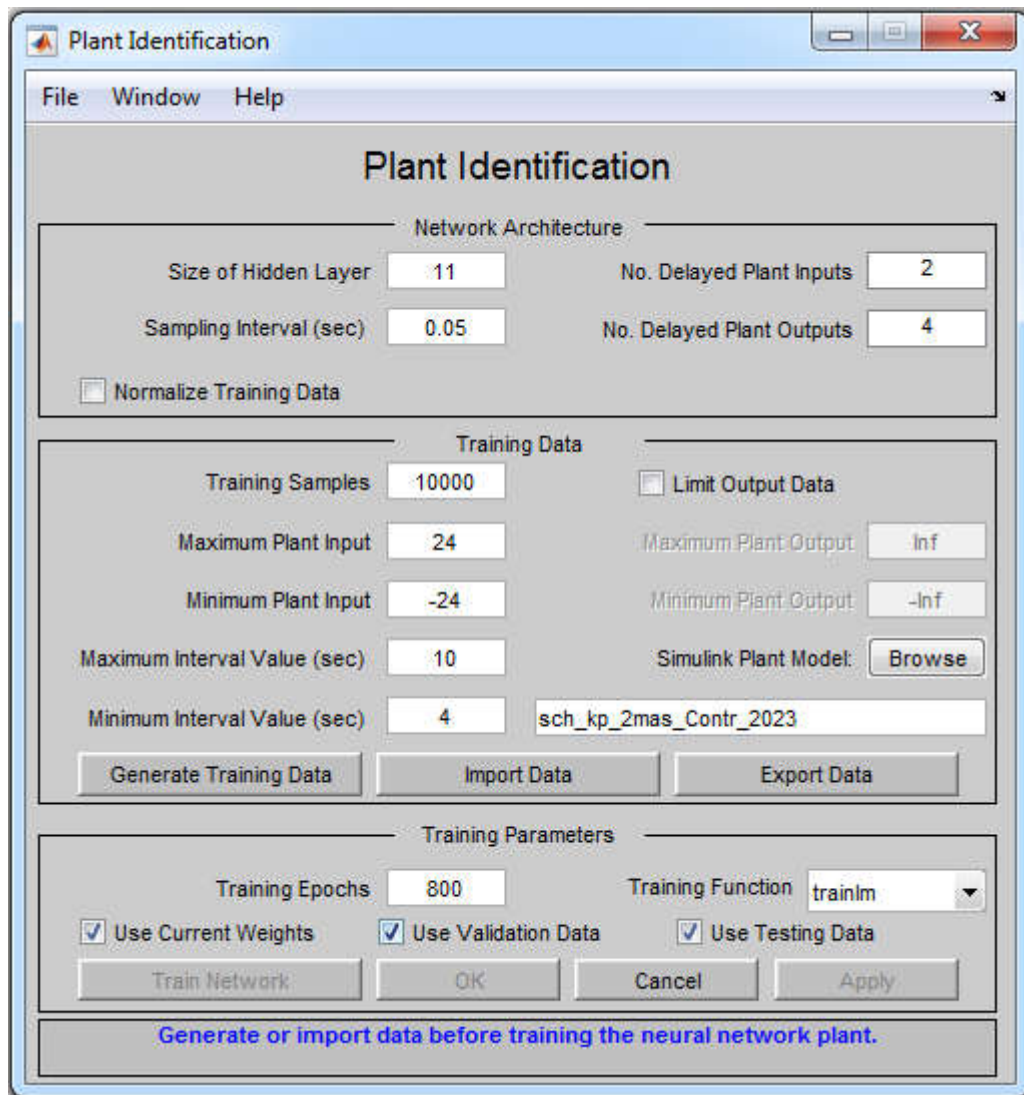


Рисунок 4.8 – Вікно процедури ідентифікації об'єкта керування

Опис вікна Plant Identification

Секція Network Architecture (Архітектура мережі).

У цій частині задаються параметри, які визначають структуру нейронної мережі, що використовується для ідентифікації та подальшого відтворення динаміки досліджуваної системи. Коректний вибір налаштувань у секції **Network Architecture** є принципово важливим, оскільки саме від них залежить, наскільки точно нейромережа здатна описати поведінку об'єкта.

Параметр Size of Hidden Layer (кількість нейронів прихованого шару). Цей параметр вказує, скільки нейронів міститиметься у прихованому шарі мережі, яка застосовується для моделювання або ідентифікації об'єкта. Вибір

розміру прихованого шару є одним із ключових рішень при проектуванні нейронмережі:

- збільшення кількості нейронів, як правило, дозволяє мережі краще відтворювати складні, нелінійні залежності в даних;
- однак занадто великий прихований шар може спричинити перенавчання (overfitting), коли мережа добре відтворює навчальні дані, але погано узагальнює поведінку на нових вхідних впливах.

Тому розмір прихованого шару зазвичай розглядають як гіперпараметр, який підбирають експериментально, орієнтуючись на якість моделювання для конкретної задачі.

Параметр Sampling Interval (sec) (інтервал дискретизації, с). Цей параметр задає часовий інтервал між двома послідовними вимірюваннями сигналів, що використовуються для ідентифікації. Іншими словами, це період, з яким знімаються дані з моделі або реального об'єкта.

Вибір інтервалу дискретизації має суттєвий вплив на:

- деталізацію та роздільну здатність отриманих даних;
- точність подальшого відтворення динаміки системи.

Якщо інтервал вибірки занадто великий, модель може «не побачити» швидких змін у процесі, а надто малий інтервал призводить до надлишку даних і зростання обчислювальних витрат. Тому при заданні Sampling Interval необхідно враховувати швидкодію об'єкта і вимоги до точності результатів ідентифікації.

No. Delayed Plant Inputs (кількість затриманих входів). Цей параметр визначає, скільки попередніх значень вхідних сигналів системи беруться до уваги під час ідентифікації. Урахування запізнених входів дає змогу моделі відобразити наявні у системі затримки та більш коректно відтворити її динаміку.

Такі затримки часто зустрічаються в реальних керованих процесах, де реакція на вхід проявляється із певним часовим відставанням. Тому коректне встановлення параметра **No. Delayed Plant Inputs** є важливим для отримання

точної моделі, особливо коли відомо, що між дією на об'єкт і його реакцією існує часовий лаг.

Правильне задання цього параметра дає можливість адаптувати модель до реальних умов роботи системи та підвищити якість процесу ідентифікації.

No. Delayed Plant Outputs (кількість затриманих виходів). Параметр указує кількість попередніх значень вихідних сигналів, які залучаються до формування моделі. Використання запізнених виходів дозволяє врахувати затримки у формуванні відгуку системи, що суттєво впливає на точність моделювання її поведінки.

Цей параметр визначає, наскільки глибоко модель аналізує історію вихідних значень. Якщо система має значні внутрішні затримки, то включення більшої кількості запізнених виходів може суттєво підвищити адекватність отриманої моделі.

Оптимальний вибір No. Delayed Plant Outputs залежить від властивостей конкретного об'єкта та рівня точності, якого потрібно досягти під час ідентифікації.

Normalize Training Data (нормалізація навчальних даних). Цей пункт відповідає за попереднє масштабування вхідних і вихідних сигналів перед їх використанням у процесі навчання нейромережевої моделі. Вмикання цього параметра означає, що всі дані будуть приведені до певного стандартного діапазону чи форми – наприклад, до інтервалу $[0;1]$, або ж віднормовані за середнім значенням і стандартним відхиленням.

Якщо нормалізація активована, система автоматично перетворює дані та подає на вхід мережі більш узгоджені значення. Якщо ж цей пункт вимкнено, навчання виконується на сирих даних без попередньої обробки.

Основне призначення нормалізації – забезпечити стабільніше та швидше навчання нейронної мережі, особливо у випадках, коли окремі змінні мають різні масштаби або різко відрізняються за величинами. Коректно виконане масштабування також позитивно впливає на точність і якість побудованої моделі.

Доцільність увімкнення **Normalize Training Data** визначається особливостями конкретної системи та вимогами до процесу її ідентифікації.

Секція Training Data (навчальні дані). У розділі *Training Data* задається набір сигналів, на основі яких формується нейромережева модель досліджуваної системи. До таких даних входять часові послідовності вхідних впливів та відповідних виходів об'єкта, що дозволяє мережі відтворити закономірності та динаміку реального процесу.

Головне призначення цього набору – забезпечити мережу достатньою кількістю інформації для коректного навчання, щоб побудована модель могла описувати поведінку системи в різних умовах. Як правило, до Training Data включають результати роботи об'єкта в різних режимах, що підвищує здатність моделі враховувати широке коло можливих станів та реакцій.

Під час ідентифікації саме на цій вибірці нейронна мережа шукає відповідність між вхідними і вихідними сигналами, що визначає точність отриманої моделі. Тому якість і повнота навчальних даних безпосередньо впливають на надійність побудованої моделі.

Training samples (кількість навчальних точок). Цей параметр визначає розмір навчальної вибірки – тобто число вимірів, використаних для побудови моделі. Від його значення залежить, наскільки глибоко та точно нейронна мережа зможе відтворити характеристики керованого об'єкта: надто мала вибірка може призвести до неточностей, тоді як достатній обсяг даних забезпечує більш адекватну ідентифікацію.

Maximum Plant Input (верхня межа вхідного сигналу). Цей параметр визначає найбільше допустиме значення сигналу, що подається на вхід системи під час процедури ідентифікації. Фактично він задає верхню границю амплітуди вхідного впливу, який використовуватиметься у навчальній вибірці.

Коректне встановлення цього обмеження дає змогу забезпечити, щоб модель навчалася лише в межах реальних або фізично обґрунтованих режимів роботи об'єкта. Якщо допустимий максимум буде вибрано занадто великим, то

мережа може навчитися на значеннях, які не відповідають реальним умовам експлуатації, що негативно позначиться на точності прогнозування.

Minimum Plant Input (нижня межа вхідного сигналу). Параметр визначає мінімально можливе значення керуючого сигналу, який може подаватися під час збирання даних для ідентифікації. Це нижня межа діапазону вхідних впливів, що враховується у процесі навчання нейромережевої моделі.

Разом із максимальним значенням цей параметр формує допустимий інтервал вхідних сигналів, у межах якого система вважається працездатною. Невдало вибрані межі можуть спричинити зниження якості моделі, оскільки мережа або не отримає достатньо різноманітних даних, або навпаки – опрацьовуватиме нереалістичні значення сигналів.

Правильно заданий діапазон входів забезпечує адекватність моделі та її здатність точно передбачати реакцію системи в реальних умовах.

Maximum Interval Value (sec) – верхня межа інтервалу часу. Цей параметр задає максимальну тривалість інтервалу у секундах, протягом якого використовуються вхідні та вихідні сигнали для ідентифікації динаміки системи. По суті, він визначає максимально допустиму довжину часу, на яку поширюється навчальний набір даних під час побудови моделі.

Встановлення цього значення дозволяє обмежити часовий діапазон аналізованої інформації. Це особливо корисно, коли деякі періоди роботи системи менш значущі для процесу ідентифікації, а дані в інших проміжках є ключовими. Вибір параметра Maximum Interval Value зазвичай враховує специфіку досліджуваного об'єкта та тривалість циклів його роботи, обмежуючи аналіз лише необхідним часовим відрізком.

Minimum Interval Value (sec) (нижня межа інтервалу часу). Цей параметр визначає мінімальну тривалість інтервалу у секундах, протягом якого враховуються вхідні та вихідні дані для ідентифікації системи. Іншими словами, він задає коротший допустимий часовий проміжок для навчальної вибірки.

Налаштування Minimum Interval Value дозволяє контролювати, які частини часового ряду вважати значущими для побудови моделі. Використання цьо-

го параметра допомагає виключати надто короткі або неінформативні інтервали, забезпечуючи, щоб у навчанні брали участь тільки дані з необхідною тривалістю або значущістю. Правильне встановлення мінімального інтервалу дозволяє оптимізувати ідентифікаційний процес відповідно до особливостей та вимог конкретної системи.

Limit Output Data (обмеження вихідних даних). Цей параметр активує вікно управління, яке дозволяє регулювати обсяг вихідних даних, що використовуються під час ідентифікації. При ввімкненні цього вікна стають доступними два додаткові поля для редагування:

- Maximum Plant Output – максимальне значення вихідного сигналу.
- Minimum Plant Output – мінімальне значення вихідного сигналу.

Параметри Maximum Plant Output та Minimum Plant Output визначають обмеження вихідних даних, які беруть участь у процесі навчання моделі та аналізу динаміки системи. Це особливо корисно, коли потрібно сконцентруватися на певному діапазоні вихідних величин або дослідити специфічну поведінку системи протягом конкретного проміжку часу.

Використання цих обмежень дозволяє зменшити обсяг оброблюваних даних, спростити процес навчання та зробити ідентифікацію моделі більш ефективною.

Simulink Plant Model (модель об'єкта в Simulink). Цей параметр задає модель керованого об'єкта в середовищі Simulink, визначаючи її вхідні та вихідні порти, які використовуються для побудови нейромережевої моделі. Вибирається конкретна модель об'єкта або системи для ідентифікації. За допомогою кнопки Browser здійснюється вибір моделі Simulink; у цій роботі використовується схема моделі, наведена на рис. 4.9.

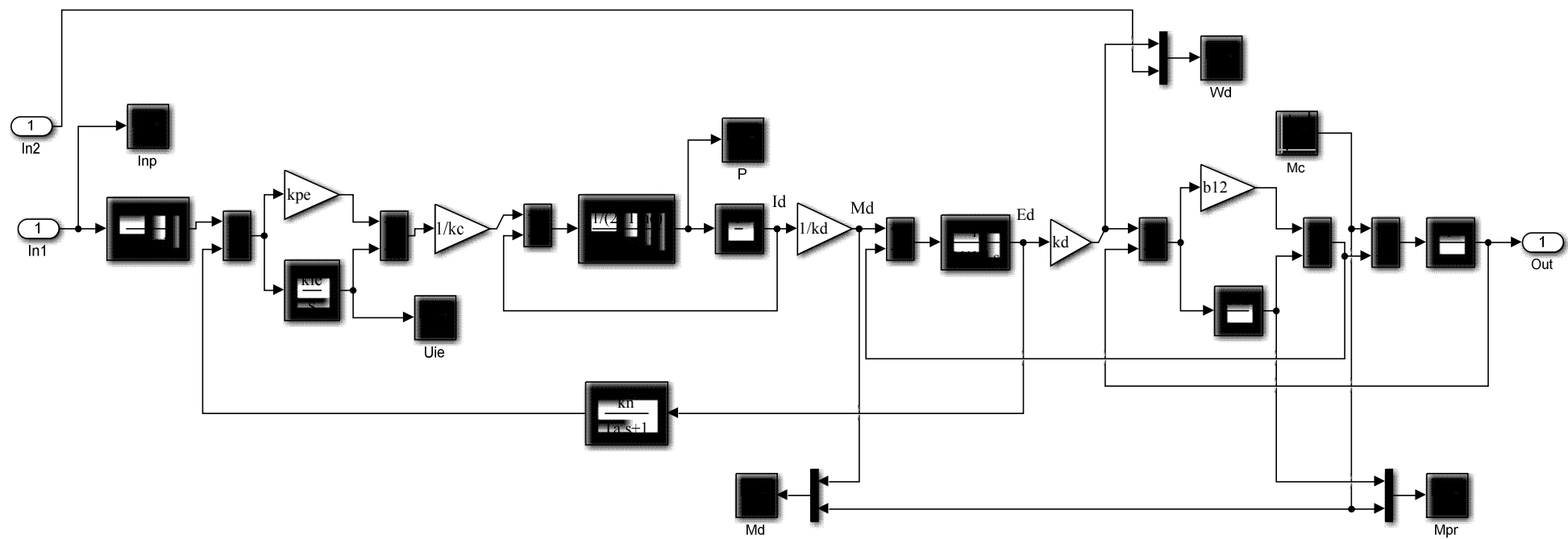


Рисунок 4.9 – Модель керованого об'єкта нейрорегулятора

Generate Training Data (генерація навчальних даних). Ця функція запускає процес створення навчальної послідовності для ідентифікації динаміки системи на основі заданих параметрів та моделі об'єкта. Вона дозволяє формувати симульовані дані для навчання, коли реальні виміряні дані недоступні.

Import Data (імпорт даних). Дозволяє завантажити навчальну послідовність із робочої області або зовнішнього файлу для використання у процесі ідентифікації.

Export Data (експорт даних). Надає можливість зберегти згенеровані дані в робочу область Simulink або у файл формату MAT для подальшого використання.

Секція Training Parameters (параметри навчання). Параметри навчання (Training Parameters) визначають, яким чином буде організовано процес навчання моделі системи на основі навчальних даних. Вони забезпечують керування процесом ідентифікації та впливають на точність і ефективність отриманої моделі.

Training Epochs (кількість епох навчання). Параметр Training Epochs задає число ітерацій або повних циклів навчання нейромережі. Кожна епоха передбачає, що модель проходить через весь набір навчальних даних, оновлюючи внутрішні параметри для поліпшення точності відтворення динаміки системи.

Правильне визначення кількості епох є критично важливим: якщо їх занадто мало, модель може залишитися недостатньо навченою і не здатною точно відтворювати поведінку системи; надмірна кількість епох може призвести до перенавчання (overfitting), коли модель запам'ятовує навчальні дані, але погано узагальнює нові випадки. Оптимальна кількість епох залежить від конкретної задачі ідентифікації та характеристик навчальних даних.

Training Function (навчальна функція). Параметр Training Function визначає алгоритм, за допомогою якого модель нейромережі навчається на підготовлених даних та оптимізує свої параметри. Від правильного вибору функції навчання залежить ефективність процесу та точність отриманої моделі.

У MATLAB і пакеті Neural Network Toolbox доступні різні функції навчання, такі як:

- `traingd` – градієнтний спуск;
- `traingdm` – градієнтний спуск з моментом;
- `trainlm` – метод найменших квадратів Левенберга-Марквардта;
- `trainbr` – байєсівський регулятор.

Кожна з цих функцій має свої особливості та параметри, які можна налаштовувати для досягнення максимальної точності моделі та здатності адекватно відтворювати динаміку системи. Training Function визначає, як нейромережа коригує внутрішні ваги та параметри під час навчання для підвищення якості прогнозу та стабільності моделі.

Use Current Weights (використовувати поточні ваги). Це елемент керування, який дозволяє підтвердити, що під час подальшої ідентифікації та навчання нейронної мережі слід використовувати вже наявні, попередньо сформовані вагові коефіцієнти.

Параметр *Use Current Weights* визначає, чи застосовуватимуться актуальні значення ваг як стартова точка для нового процесу навчання моделі. Він безпосередньо впливає на ініціалізацію мережі:

- Якщо *Use Current Weights* встановлено на «Так» (активовано), тоді мережа починає тренування зі значеннями ваг, які сформувалися під час попередніх навчальних процедур. Тобто навчання продовжує вдосконалення вже наявної моделі.
- Якщо параметр вимкнений («Ні»), навчання розпочинається знову – ваги ініціалізуються випадковим чином або відповідно до обраного алгоритму початкової ініціалізації.

Використання вже наявних ваг доцільне, коли потрібно досягти покращення або точнішої корекції існуючої моделі, а також коли попередні ваги вже певною мірою відображають динаміку об'єкта. Важливо розуміти, що цей параметр визначає стартові умови процесу навчання та може суттєво впливати на кінцеву точність ідентифікації системи.

Use Validation Data (використовувати дані валідації). Цей параметр задає, чи потрібно залучати окрему вибірку даних для етапу валідації під час навчання нейронної мережі. Валідація є важливою процедурою, що дозволяє оцінити здатність моделі узагальнювати інформацію, а не просто запам'ятовувати навчальні приклади.

- Якщо опцію *Use Validation Data* активовано («Так»), навчальний процес використовує додатковий набір даних, який не входить до навчальної вибірки. Це дає змогу оцінити продуктивність моделі на даних, які вона не бачила під час навчання. Такий підхід допомагає виявити перенавчання та визначити рівень узагальнення.

- Якщо параметр вимкнено («Ні»), валідація не проводиться, і результати оцінки моделі ґрунтуються лише на навчальній вибірці. Такий варіант може застосовуватися у випадках, коли немає можливості сформувати додатковий набір даних.

Використання валідаційної вибірки зазвичай рекомендується, оскільки вона забезпечує більш точну оцінку якості моделі та допомагає уникнути деградації її узагальнюючих властивостей.

Use Testing Data (використовувати тестові дані). Це налаштування визначає, чи буде застосовано спеціальний набір тестових даних для підсумкової оцінки роботи моделі після завершення процесу навчання та валідації. Тестові дані є незалежною вибіркою, яку модель ніколи не бачить під час навчання.

- Якщо параметр *Use Testing Data* встановлено на «Так», то після навчання на тренувальних даних і перевірки на валідаційних, модель проходить додаткову оцінку на тестовому наборі. Це дозволяє отримати об'єктивну картину того, наскільки якісно модель зможе працювати з абсолютно новими сигналами та чи здатна вона коректно відтворювати реальну динаміку системи.

- Якщо параметр вимкнено («Ні»), тестування не використовується, і оцінювання моделі обмежується навчальною та, за потреби, валідаційною вибірками.

Використання тестових даних є важливим етапом для перевірки здатності моделі до узагальнення, запобігання перенавчанню та оцінки її практичної придатності.

Процедура Generate Training Data (генерувати навчальні дані). Після вибору цієї опції запускається програмний модуль, який формує навчальну послідовність. Генерація відбувається шляхом подачі на модель керованого об'єкта в середовищі *Simulink* набору випадкових ступінчастих сигналів. У процесі формування навчальних даних на екран виводяться графіки вхідних та вихідних сигналів об'єкта (рис. 4.10), що дозволяє візуально контролювати характер реагування системи.

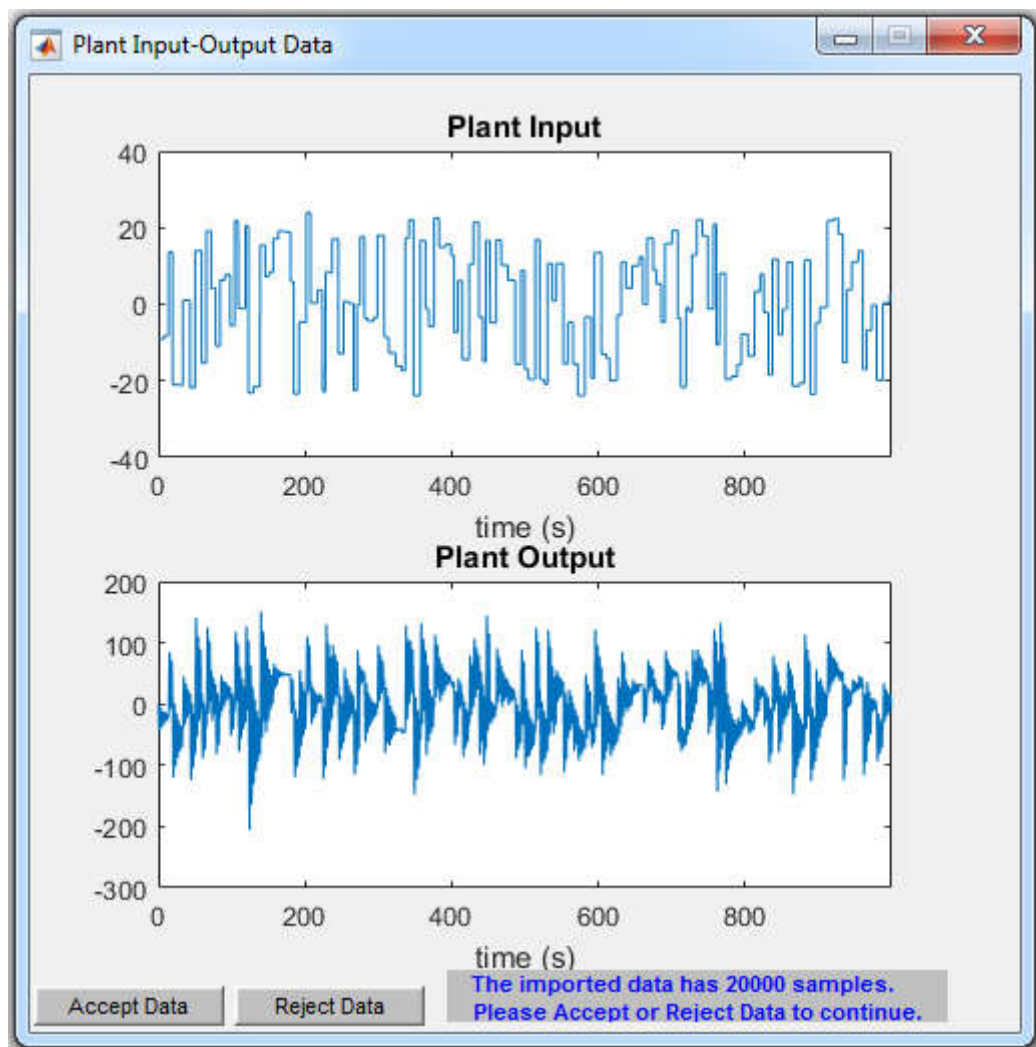


Рисунок 4.10 – Графіки вхідного та вихідного сигналів під час генерації навчальної послідовності

Після завершення генерації користувачу пропонується прийняти отримані дані (*Accept Data*) або відхилити їх (*Reject Data*), якщо вони не відповідають вимогам щодо структури чи якості.

Якщо дані прийняті користувачем, програма автоматично відкриває модифіковане вікно **Plant Identification** (рис. 4.11). У цьому вікні деякі елементи інтерфейсу стають недоступними для редагування, а кнопка **Generate Training Data** змінюється на кнопку **Erase Generate Data**, що надає можливість видалити раніше згенеровані навчальні дані.

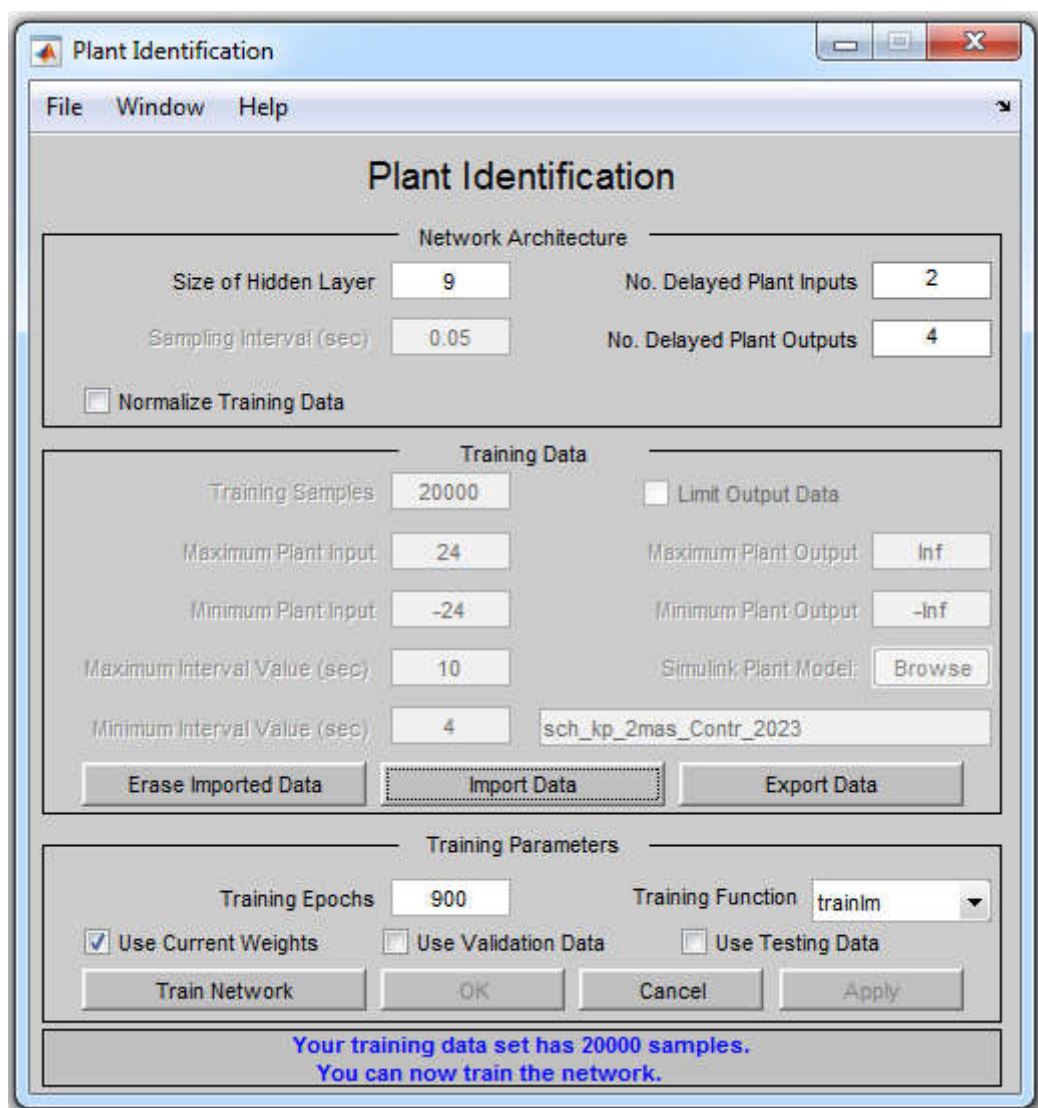


Рисунок 4.11 – Вікно Plant Identification після генерації навчальної послідовності

Натискання кнопки **Train Network** ініціює створення нейронної мережі з прямим розподілом сигналу за допомогою М-функції **newff**. Ця функція не тільки формує мережу з ім'ям **netn**, але й автоматично ініціалізує її ваги та зсуви, готуючи мережу до навчання. Створену нейронну мережу можна відобразити в середовищі Simulink за допомогою оператора **gensim(netn)** (див. рис. 4.12).

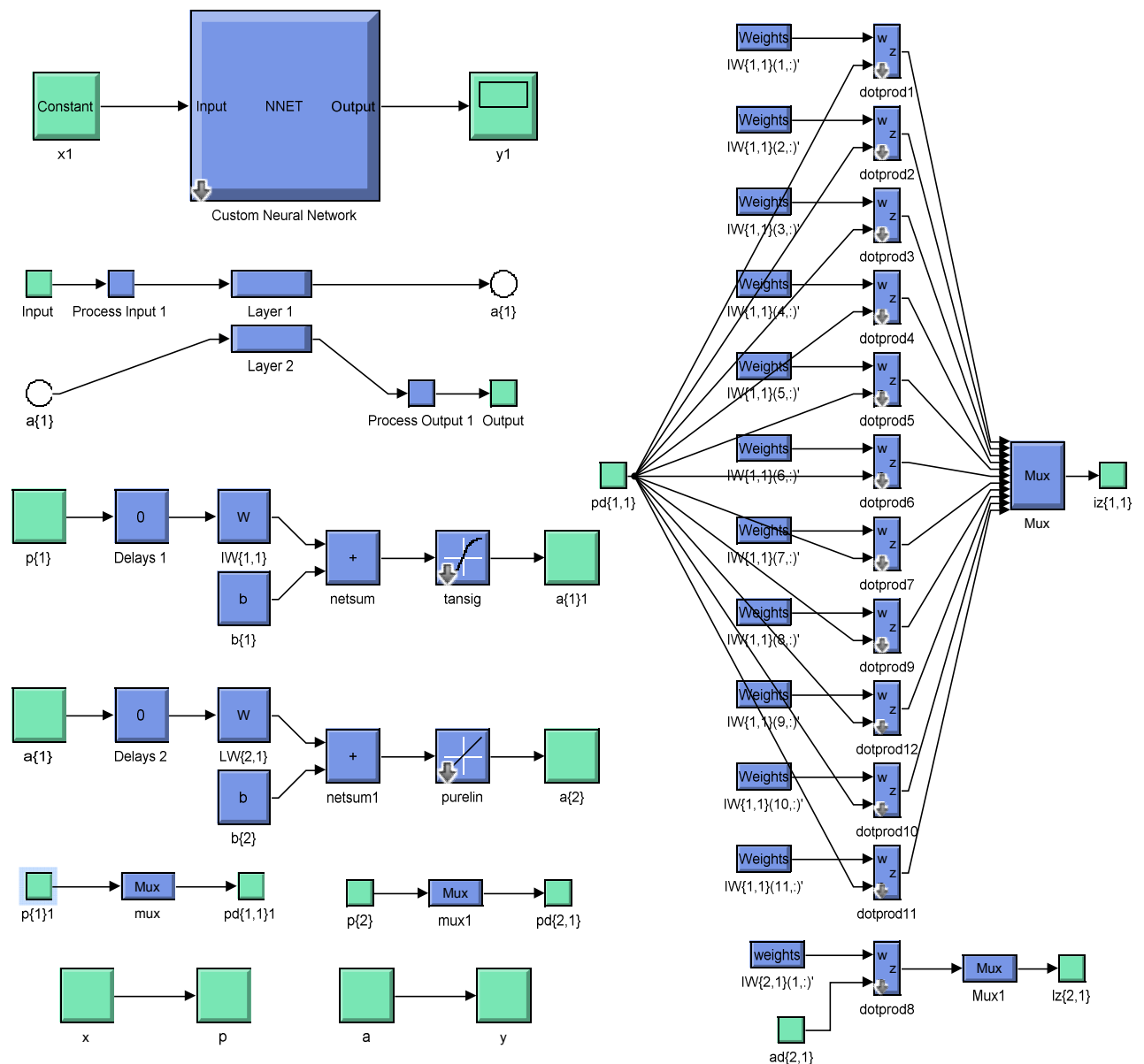


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

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

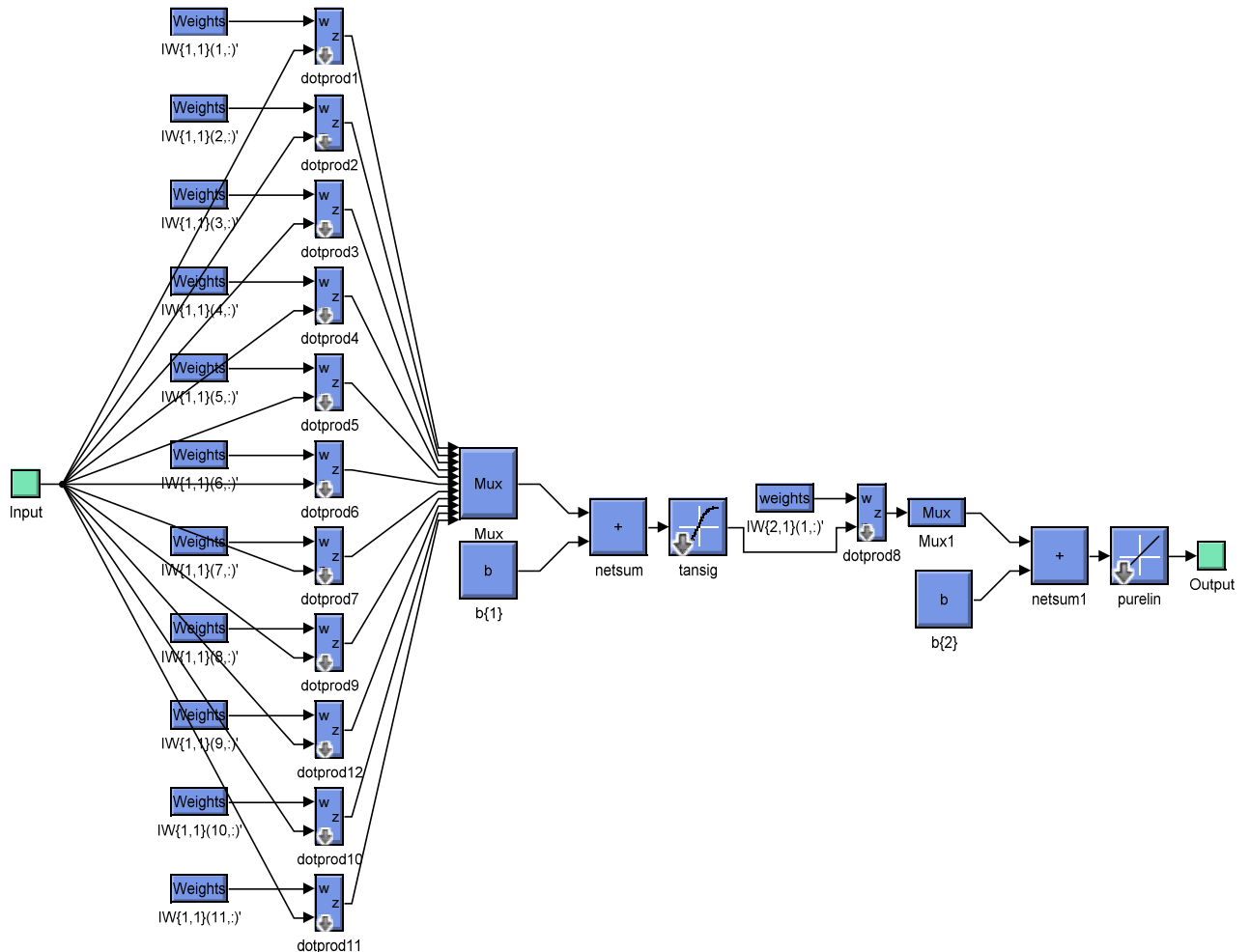


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

Створена мережа є статичною, тобто вона не містить елементів затримки або зворотних зв'язків. Мережа приймає один вектор входу з шістьма елементами, має два шари: перший – прихований, з визначеною кількістю нейронів, другий – вихідний, що складається з одного нейрона. Активаційні функції обрані такі: у прихованому шарі використовується гіперболічний тангенс (**tansig**), а у вихідному шарі – лінійна функція (**purelin**).

Після створення мережі розпочинається процес її навчання. Процес навчання та його результати наочно демонструються через діалогове вікно **Neural Network Training (nntraintool)**, яке показано на рис. 4.14. Це вікно є основним інтерфейсом для навчання нейронних мереж у **Neural Network Toolbox** системи MATLAB. Воно забезпечує потужні можливості для налаштування мережі, візуалізації процесу навчання та моніторингу динаміки зміни параметрів мережі під час тренування.

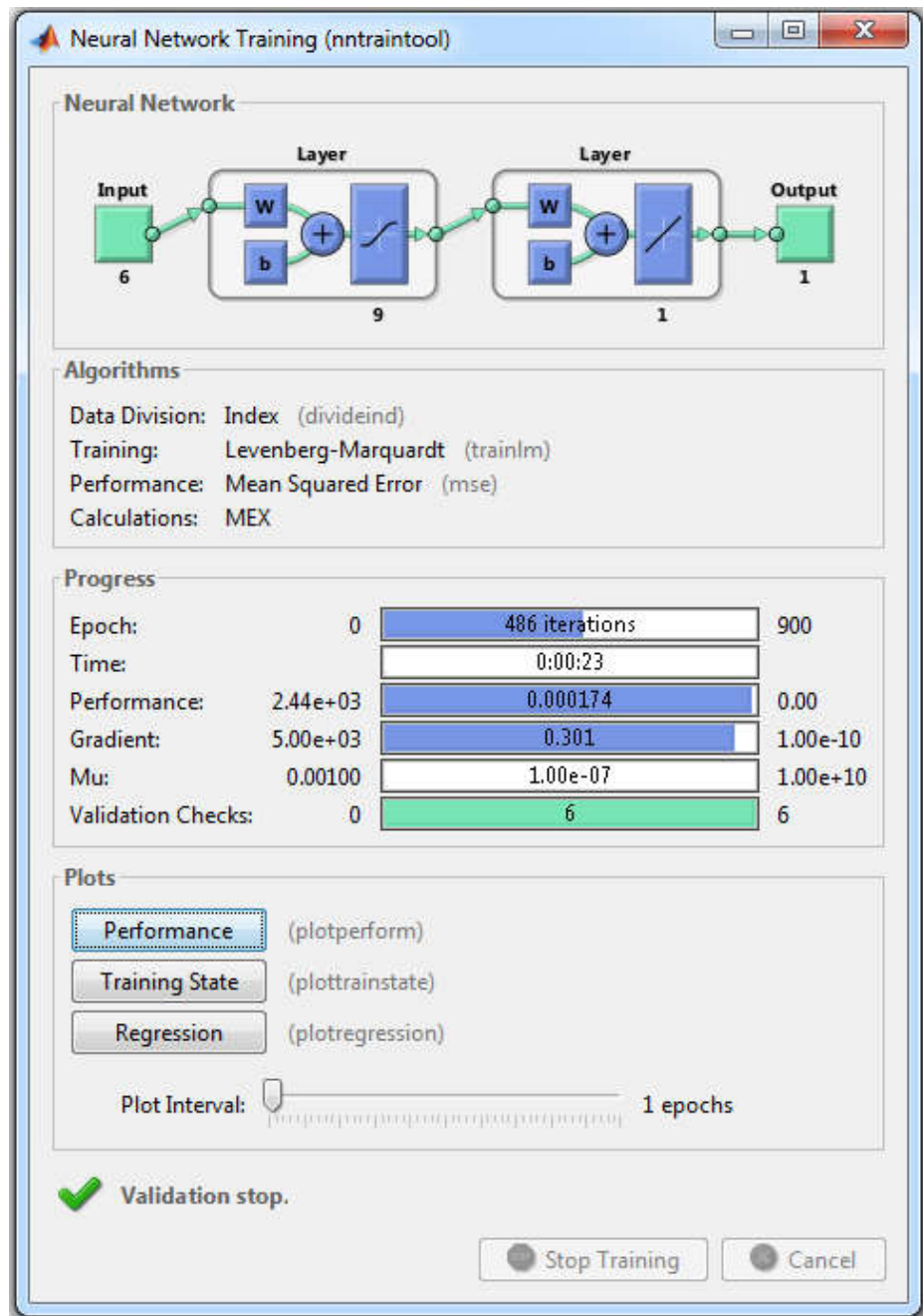


Рисунок 4.14 - Головне вікно для відображення процесу та результатів навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

Вікно **Neural Network Training (nntraintool)** у MATLAB складається з декількох функціональних секцій, кожна з яких відповідає за окремий аспект навчання нейронної мережі.

Секція Neural Network (нейронна мережа). У цій секції відображається структура згенерованої мережі, включаючи кількість шарів, нейронів у кожному шарі, а також типи активаційних функцій. Користувач може наочно бачити топологію мережі та взаємозв'язки між шарами, що допомагає оцінити її складність та потенційну здатність до моделювання системи.

Секція Algorithms (алгоритми навчання). Тут відображаються алгоритми, які використовуються для тренування мережі, а також параметри цих алгоритмів. Це дозволяє вибирати відповідний метод оптимізації для конкретного завдання та регулювати швидкість навчання, величину кроку, критерії зупинки та інші параметри, що впливають на процес адаптації ваг і зсувів мережі.

Data Division (поділ даних). Параметр Data Division: Index визначає, які приклади з набору даних будуть використані для навчання, валідації та тестування нейронної мережі. Цей механізм дозволяє контролювати, яка частина даних йде на формування навчальної моделі, а яка використовується для перевірки її здатності до узагальнення. Якщо дані не розподіляються на підмножини, рядок Data Division у вікні nntraintool не відображається.

Training (навчання). Ця секція відповідає за процес оновлення ваг і зсувів нейронної мережі під час тренування. Серед популярних алгоритмів навчання доступні Backpropagation, Levenberg-Marquardt, Bayesian Regularization та інші.

Вибір Training: Levenberg-Marquardt (trainlm) означає, що мережа буде навчатися за допомогою алгоритму Левенберга-Марквардта. Цей метод поєднує властивості методу найменших квадратів та методу Ньютона для ефективної оптимізації функції втрат. Його основна перевага полягає в швидкій мінімізації функції втрат і стабільному пошуку локального мінімуму через адаптивне онов-

влення вагових коефіцієнтів мережі. Алгоритм добре підходить для складних навчальних задач, де стандартні методи можуть показувати гірші результати.

Performance: Mean Squared Error (MSE). Параметр відображає показники точності навчання нейронної мережі на основі обраної функції втрат та даних для валідації. Цей показник дає змогу оцінити, наскільки ефективно мережа відтворює динаміку системи під час навчання, а також визначає момент, коли досягнуто оптимальної точності та процес тренування можна завершити. Моніторинг Performance допомагає уникнути перенавчання та забезпечує належну здатність моделі до узагальнення на нових даних.

Performance: Mean Squared Error (MSE) визначає використання середньоквадратичної помилки як критерію оцінки точності нейронної мережі під час її навчання. Середньоквадратична помилка є однією з найпоширеніших метрик у задачах регресії та моделювання. Вона розраховується як середнє значення квадратів різниці між прогнозованими виходами моделі (нейронної мережі) та відомими або фактичними значеннями з навчального набору даних.

Математично MSE можна записати так:

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

де N – кількість прикладів у наборі даних, y_{actual_i} – фактичне значення виходу, а $y_{predicted_i}$ – прогнозоване значення, яке видає нейронна мережа.

MSE показує середнє квадратичне відхилення між фактичними та передбаченими результатами. Чим менше значення MSE, тим точніше модель повторює поведінку системи. Під час навчання мережі оптимізаційний алгоритм прагне зменшити MSE, тобто підбирати ваги мережі так, щоб прогнозовані значення максимально відповідали реальним даним.

Calculations (обчислення). Секція Calculations відповідає за контроль кількості обчислень, виконаних під час процесу навчання. Вона важлива для оці-

нки продуктивності та часу, необхідного для завершення тренування нейронної мережі.

Параметр **Calculations: MEX** визначає, чи слід використовувати MEX-функції для прискорення обчислень у MATLAB. MEX (MATLAB Executable) дозволяє запускати оптимізований код, написаний на C, C++ або Fortran, безпосередньо у MATLAB, що забезпечує значно більшу швидкість виконання порівняно зі стандартним інтерпретованим кодом MATLAB.

У контексті навчання нейронних мереж використання MEX-функцій дозволяє швидше виконувати складні математичні обчислення, що особливо корисно при роботі з великими наборами даних або складними моделями. Завдяки MEX-файлам тренування мережі стає ефективнішим, зменшується час очікування результатів і підвищується продуктивність процесу.

Секція Progress (прогрес навчання). У цій секції відображається поточний стан навчання нейронної мережі в реальному часі. Вона надає користувачу можливість спостерігати за ходом навчання та оцінювати ефективність процесу. Інформація, що міститься у секції, допомагає контролювати навчання моделі, виявляти проблеми на ранніх стадіях та своєчасно вносити необхідні коригування для підвищення точності та стабільності моделі.

Epoch (епоха). Параметр Epoch означає повний прохід усіх прикладів навчального набору даних через нейронну мережу. Під час кожної епохи мережа отримує всі вхідні дані один раз, після чого відбувається оновлення її вагових коефіцієнтів та зсувів (biases) відповідно до обчислених помилок. Кількість епох визначає, скільки разів модель "пройде" через дані, що дозволяє поступово покращувати її здатність відтворювати динаміку системи. Відстеження епох дозволяє оцінювати, наскільки ефективно мережа навчається, і приймати рішення про зупинку або продовження тренування.

Time (час). Параметр Time показує фактичний час, витрачений на процес навчання нейронної мережі. Це допомагає контролювати тривалість навчання,

планувати ресурси та оцінювати продуктивність навчальної сесії, особливо при роботі з великими наборами даних або складними моделями.

Performance (ефективність/якість навчання). Показник Performance відображає, наскільки успішно нейронна мережа навчається на даних, використовуючи обрану функцію втрат (loss function). Для задач регресії найчастіше застосовується середньоквадратична помилка (Mean Squared Error, MSE), яка обчислює середнє значення квадратів різниці між прогнозованими виходами та реальними значеннями цільової змінної. Чим менше значення MSE, тим точніше мережа відтворює поведінку системи. Показник Performance дозволяє оцінити ефективність навчання на тренувальних та валідаційних даних і допомагає приймати рішення про необхідність коригування параметрів навчання.

Gradient (градієнт). Параметр Gradient демонструє вектор градієнта функції втрат відносно всіх параметрів нейронної мережі, тобто ваг і зсувів. Градієнт показує напрямок і величину змін, які слід застосувати до параметрів мережі, щоб мінімізувати функцію втрат. Під час навчання мережі використовується метод зворотного поширення помилки (backpropagation), і градієнт є ключовим елементом цього процесу. Він дозволяє алгоритму навчання ефективно коригувати ваги та зсуви, забезпечуючи поступове зменшення помилки і підвищення точності моделі.

Таким чином, секція Progress у nntraintool дозволяє детально контролювати процес навчання нейронної мережі, оцінювати якість навчання на кожному етапі та своєчасно реагувати на будь-які відхилення або проблеми, що виникають під час тренування.

Mu (тау-параметр). Параметр Mu відноситься до так званого тау-параметра, який застосовується в алгоритмах оптимізації для регулювання швидкості навчання нейронної мережі. Він особливо актуальний при використанні адаптивних методів оптимізації, таких як зворотне поширення помилки (backpropagation) або алгоритми на кшталт BFGS (Broyden-Fletcher-Goldfarb-Shanno). Основна функція Mu полягає у корекції величини кроку навчання відповідно до поточної динаміки процесу оптимізації. Це дозволяє алгоритму ефе-

ктивно знаходити мінімум функції втрат і забезпечує стабільну збіжність навіть при складних поверхнях втрат або великих градієнтах.

Значення μ зазвичай обчислюється автоматично алгоритмом на кожній ітерації навчання та може змінюватися в процесі, підлаштовуючись під поведінку мережі. Від правильного регулювання цього параметра залежить швидкість та стабільність навчання, а також здатність мережі до ефективного узагальнення даних.

Validation Checks (перевірки на валідаційному наборі). Параметр Validation Checks визначає кількість перевірок точності або якості нейронної мережі під час процесу навчання за допомогою окремого набору даних, який не бере участі у безпосередньому навчанні – так званого validation dataset. Основна мета цих перевірок – контроль за процесом навчання і запобігання перенавчанню (overfitting), коли модель «запам'ятовує» навчальні дані і втрачає здатність правильно реагувати на нові, невідомі приклади.

Наприклад, якщо встановлено 6 перевірок, нейронна мережа після кожних шести епох (де епоха – це повний прохід через весь навчальний набір даних) перевірятиме свою продуктивність на валідаційному наборі. Якщо точність моделі на цих даних перестає покращуватися або починає погіршуватися, це сигнал для зупинки навчання, оскільки подальші ітерації можуть призвести до перенавчання.

Таким чином, Validation Checks дозволяє автоматично відслідковувати ефективність мережі на даних, які вона раніше не бачила, і забезпечує своєчасне завершення навчання для досягнення оптимального балансу між підгонкою під тренувальні дані та здатністю узагальнювати на нові дані.

Секція Plots (графіки). Ця секція призначена для візуалізації ходу навчання нейронної мережі і дозволяє відслідковувати зміни її продуктивності у реальному часі. Візуальні графіки допомагають аналізувати ефективність навчання, виявляти проблеми, такі як перенавчання, та оцінювати динаміку зменшення помилки на різних етапах тренування.

Кнопка **Performance** відкриває вікно **Neural Network Training Performance** (рис. 4.15), в якому представлені графіки середньоквадратичної помилки (**Mean Squared Error, MSE**) для навчального, валідаційного та тестового наборів даних. Ці графіки демонструють, як змінюється точність мережі під час кожної ітерації навчання.

Графіки **MSE** дозволяють відстежувати, чи відбувається зменшення помилки на всіх наборах даних і чи немає ознак перенавчання, коли помилка на валідаційному та тестовому наборі починає збільшуватися. Основна мета – досягти мінімальної помилки на всіх наборах даних для отримання максимально точної та узагальнювальної моделі.

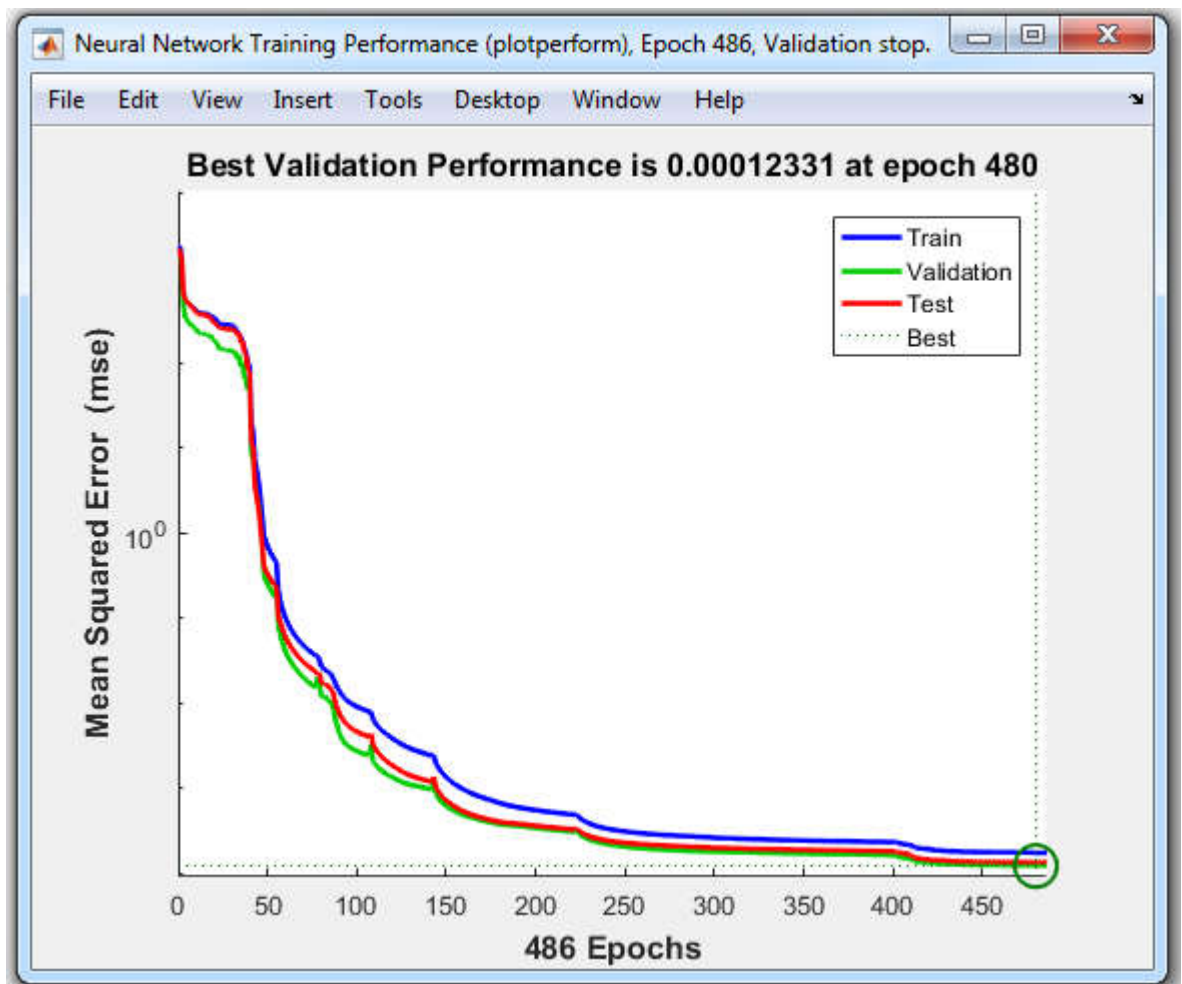


Рисунок 4.15 – Вікно контролю процесу навчання

Кнопка **Training State** відкриває вікно **Neural Network Training State** (рис. 4.16), яке відображає стан навчання нейронної мережі на конкретний момент часу. У цьому вікні можна переглядати актуальні значення таких параметрів, як кількість епох, поточна середньоквадратична помилка, градієнти, тау-параметр (μ) та інші показники. Це дозволяє детально відстежувати процес навчання та приймати рішення про його коригування, наприклад, про зупинку навчання у разі виникнення перенавчання або про необхідність зміни налаштувань алгоритму.

Таким чином, секція **Plots** є ключовим інструментом для аналізу навчального процесу, що забезпечує як візуальний контроль, так і можливість оперативно реагувати на будь-які відхилення у процесі тренування нейронної мережі.

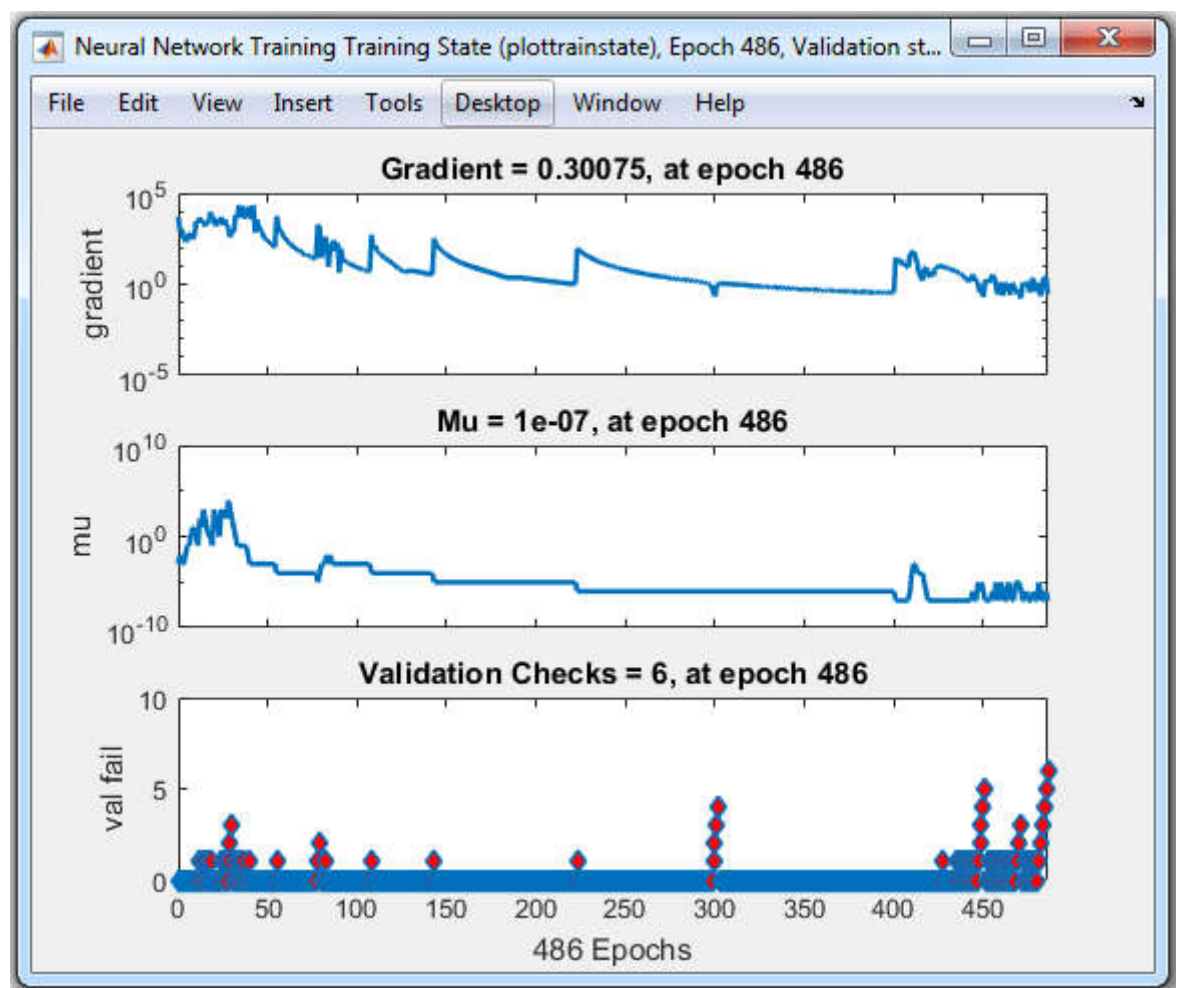


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні **Neural Network Training State** відображаються основні графіки, які дозволяють відстежувати динаміку навчання нейронної мережі в реальному часі. Це вікно дає змогу детально контролювати процес тренування та своєчасно реагувати на можливі проблеми.

Графік Gradient (градієнт). Цей графік демонструє зміну величини градієнта функції втрат під час навчання мережі. У контексті нейронних мереж градієнт є вектором, який вказує напрямок та швидкість найшвидшого зростання функції помилки. Використовуючи цю інформацію, оптимізаційний алгоритм коригує вагові коефіцієнти мережі в протилежному напрямку, щоб мінімізувати значення функції втрат.

На графіку Gradient можна побачити, як змінюється величина градієнта від початкових ітерацій до моменту, коли мережа наближається до оптимальних вагових коефіцієнтів. Спочатку градієнти можуть бути значними, але з часом вони зазвичай зменшуються, що свідчить про поступове зближення мережі до стабільного стану.

Моніторинг цього графіка допомагає виявляти проблеми gradient vanishing або gradient exploding та контролювати збіжність навчання. Надто великі або малі значення градієнта можуть сигналізувати про необхідність коригування параметрів навчання, таких як швидкість навчання або структура мережі.

Графік Mu (параметр навчання). Графік μ відображає зміну значення тау-параметра, який регулює швидкість навчання та рівень регуляризації мережі. У процесі тренування оптимізаційні алгоритми (наприклад, стохастичний градієнтний спуск або модифікації методу Бройдена-Флетчера-Гольдфарба-Шанно) використовують μ для адаптації швидкості корекції вагових коефіцієнтів.

Зміни параметра μ показують, як адаптується швидкість навчання до поточного стану мережі. Наприклад, збільшення μ може прискорити збіж-

ність, але водночас збільшити ризик нестабільності, тоді як зменшення ти може свідчити про уповільнення навчання або необхідність перегляду налаштувань оптимізатора. Регулярний контроль графіка ти дозволяє підтримувати баланс між швидкістю збіжності та стабільністю процесу навчання.

Графік Validation Failure (val fail). Цей графік демонструє зміну показника невдач на наборі валідаційних даних протягом навчання. Основне призначення валідаційного набору – оцінити здатність мережі узагальнювати знання на дані, які не використовувалися під час тренування, і запобігти перенавчанню.

Якщо кількість невдач на валідації зростає, це може вказувати на *overfitting*, тобто на надмірне підлаштування мережі під навчальні дані та втрату здатності правильно передбачати нові вхідні сигнали. У таких випадках доцільно застосувати ранню зупинку навчання, змінити гіперпараметри або додатково скоригувати структуру мережі. Мінімізація цього показника є однією з ключових цілей тренування, адже вона прямо впливає на точність та стабільність моделі в реальних умовах.

Таким чином, Neural Network Training State забезпечує комплексний моніторинг навчального процесу, дозволяючи спостерігати за градієнтами, швидкістю навчання та поведінкою мережі на валідаційних даних. Ця інформація є критично важливою для контролю якості навчання та своєчасного коригування параметрів для досягнення оптимальної моделі.

Продовження опису вікна Neural Network Training (nntraintool)

Кнопка **Regression (регресія)** призначена для налаштування параметрів навчання нейронної мережі у задачах регресії. Регресійне завдання – це один із типів задач машинного навчання, мета якого полягає у передбаченні числових значень або кількісних показників на основі вхідних даних. У таких задачах модель намагається відтворити функціональну залежність між вхідними ознаками (параметрами) та вихідними значеннями, які називаються цільовими або відгуком.

При натисканні на кнопку **Regression** відкривається вікно **Neural Network Training Regression** (рис. 4.17), яке надає графічне та числове представлення продуктивності моделі під час навчання. У цьому вікні користувач може спостерігати детальну інформацію щодо точності прогнозів мережі на різних наборах даних.

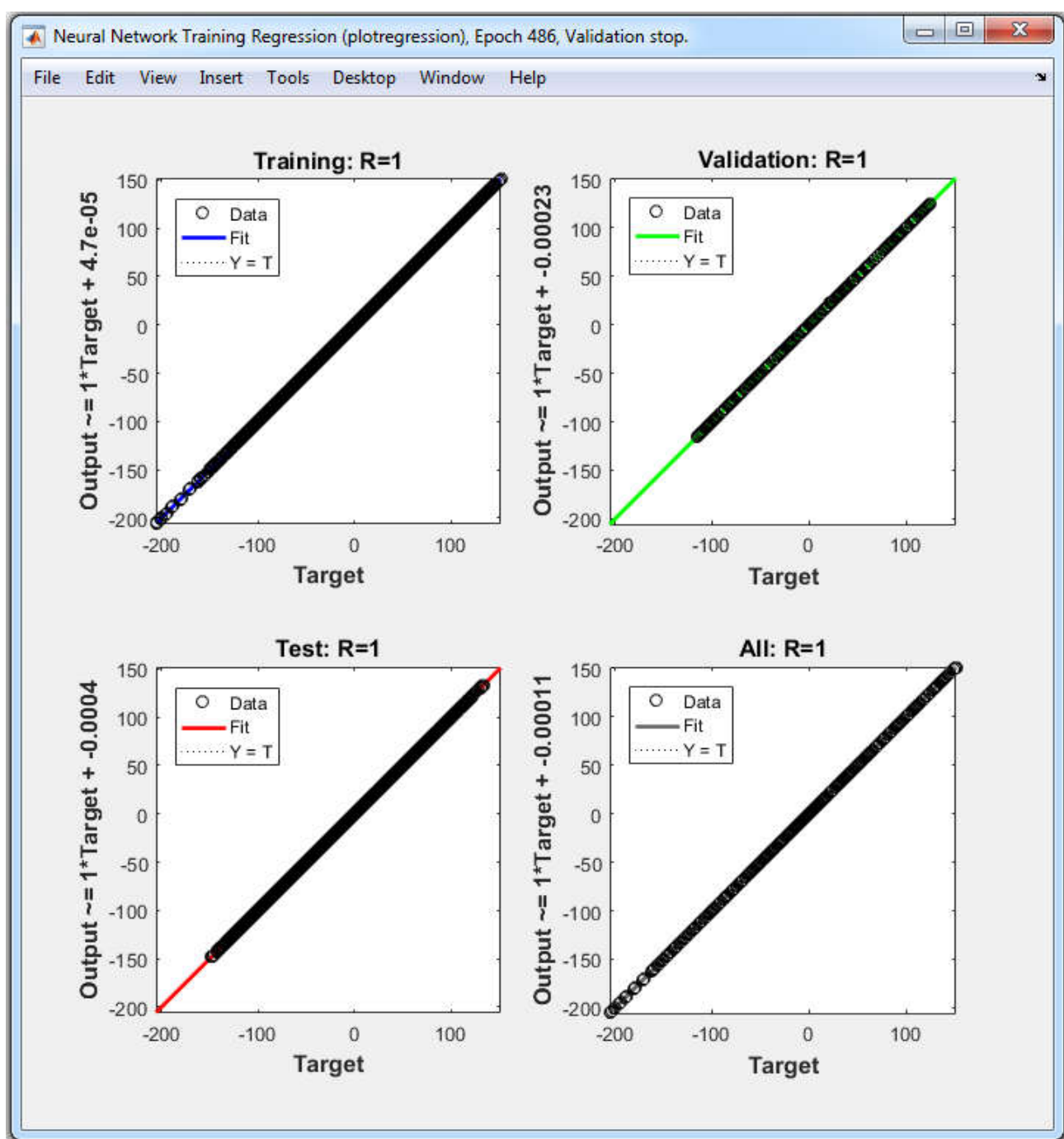


Рисунок 4.17 - Вікно Neural Network Training Regression

Опис секцій вікна Neural Neural Network Training Regression

Training (навчання). Ця секція відображає, наскільки добре нейронна мережа відтворює дані навчального набору, тобто дані, на яких модель проходила тренування. Графіки показують відповідність прогнозованих значень фактичним даним, що дозволяє оцінити ефективність навчання та ступінь узгодження мережі з вихідними даними.

Графіки Validation (валідація). Валідаційні графіки демонструють точність роботи моделі на окремому наборі даних, який не використовувався під час навчання. Валідація дозволяє оцінити, наскільки добре модель узагальнює знання на нові дані, і служить для своєчасного виявлення ознак перенавчання (overfitting).

Графіки Test (тестування). Тестові графіки включають незалежний набір даних, який не був задіяний ні під час навчання, ні під час валідації. Вони забезпечують об'єктивну оцінку продуктивності нейронної мережі на абсолютно нових даних. Аналіз графіків Test дозволяє перевірити, наскільки модель здатна передбачати результати для непередбачених ситуацій.

Графіки All (всі дані разом). Ця секція об'єднує результати прогнозування для всіх трьох наборів даних: Training, Validation та Test. Графіки **All** показують одночасно прогнозовані та фактичні значення для кожного набору, що дозволяє наочно порівнювати продуктивність мережі на різних етапах навчання. Такий підхід допомагає швидко визначити, чи існують розбіжності між поведінкою моделі на навчальних, перевіірочних і тестових даних, та робить оцінку її стабільності більш наочною.

Таким чином, вікно **Neural Network Training Regression** забезпечує комплексний аналіз регресійних результатів моделі, дозволяє відстежувати точність прогнозів на всіх основних наборах даних та вчасно коригувати параметри нейронної мережі для досягнення оптимальної продуктивності.

Графіки Data та Fit у вікні Neural Network Training Regression. У вікні Neural Network Training Regression графіки Data для наборів Training, Validation,

Test та All демонструють, наскільки прогнозовані значення моделі нейронної мережі узгоджуються з фактичними даними для кожного з наборів. На горизонтальній осі відкладаються фактичні значення з відповідного набору даних, тоді як на вертикальній осі – прогнозовані значення, отримані від моделі. Кожна точка на графіку відображає відповідність прогнозованого і справжнього значення: координата по осі X відповідає бажаному значенню, а координата по осі Y – вихідному значенню мережі. У середовищі MATLAB точки позначаються кружечками ().

Якщо всі точки розташовані близько до діагональної лінії $Y = T$, це свідчить про високу точність моделі. Велика розсіюваність точок може вказувати на наявність шуму в даних або недостатню адекватність моделі. Графіки Data дозволяють наочно оцінити відповідність моделі даним на різних наборах і виявити можливі аномалії чи області для поліпшення моделі.

Графіки **Fit** представляють результати лінійної регресії, що демонструють прогнозовані значення, які генерує нейронна мережа на основі вхідних даних. Вони відображають, наскільки модель узгоджується з фактичними значеннями, і служать для візуальної оцінки точності прогнозів. Якщо лінія Fit проходить близько до більшості точок, це вказує на високу продуктивність моделі. Відхилення лінії від фактичних значень може сигналізувати про потребу в додатковій оптимізації або корекції гіперпараметрів нейронної мережі.

У вікні Neural Network Training Regression також відображаються коефіцієнти кореляції та рівняння лінійної регресії для кожного набору даних (Training, Validation, Test та All). Вони використовуються для кількісної оцінки точності моделі.

Коефіцієнт кореляції R визначає ступінь лінійної залежності між прогнозованими та фактичними значеннями:

- $R = 1$ – ідеальна лінійна залежність, модель точно передбачає всі значення;
- $R = 0$ – відсутність лінійної залежності, модель не здатна передбачити дані;

- $R = -1$ – повна зворотна лінійна залежність, модель передбачає значення у протилежному напрямку.

Коефіцієнт кореляції, близький до 1, свідчить про високу точність регресійної моделі, тоді як значення близько до 0 або від’ємне сигналізує про низьку точність прогнозування. Використання R разом із графіками Data та Fit дозволяє комплексно оцінити ефективність моделі нейронної мережі на всіх етапах навчання та перевірки, а також визначити можливі напрями для її вдосконалення.

Продовження опису вікна Neural Network Training (nntraintool)

Plot Interval (інтервал графіка). Параметр Plot Interval визначає, як часто під час навчання нейронної мережі для задач регресії оновлюються графіки та відображаються проміжні результати. Іншими словами, він встановлює кількість ітерацій або епох, після яких відбувається оновлення інформації на графіках.

Правильне налаштування Plot Interval дозволяє ефективно контролювати обсяг відображуваної інформації під час навчання, особливо коли кількість епох велика. Менше значення Plot Interval забезпечує частіше оновлення графіків, що дає змогу уважніше аналізувати процес навчання, але одночасно збільшує обсяг візуальної інформації. Більше значення, навпаки, зменшує частоту оновлення, спрощуючи відстеження загальної тенденції навчання без перевантаження графіків.

Налаштування цього параметра є важливою частиною контролю процесу навчання, оскільки дозволяє оцінювати вплив змін параметрів мережі на результати моделі в режимі реального часу.

Кнопки Stop Training і Cancel. У вікні **Neural Network Training (nntraintool)** середовища MATLAB присутні дві кнопки для управління процесом навчання, які мають різні функції:

- **Stop Training (зупинка навчання)** – негайно припиняє процес навчання нейронної мережі без можливості його продовження. Ця кнопка викори-

стовується, коли навчання виявляється занадто тривалим, не дає задовільних результатів або потрібно терміново припинити тренування.

- **Cancel (скасувати)** – дозволяє зупинити навчання, але при цьому зазвичай передбачає підтвердження дії та збереження поточного стану мережі. Використання кнопки Cancel дає змогу зупинити навчання після перегляду проміжних результатів, внести необхідні зміни до параметрів та при необхідності продовжити навчання пізніше.

Після завершення навчання результати відображаються у вікні **Training Data for NN Predictive Control**, на якому показані графіки фактичних та прогнозованих значень, що дозволяє оцінити точність моделі та якість її навчання (рис. 4.18).

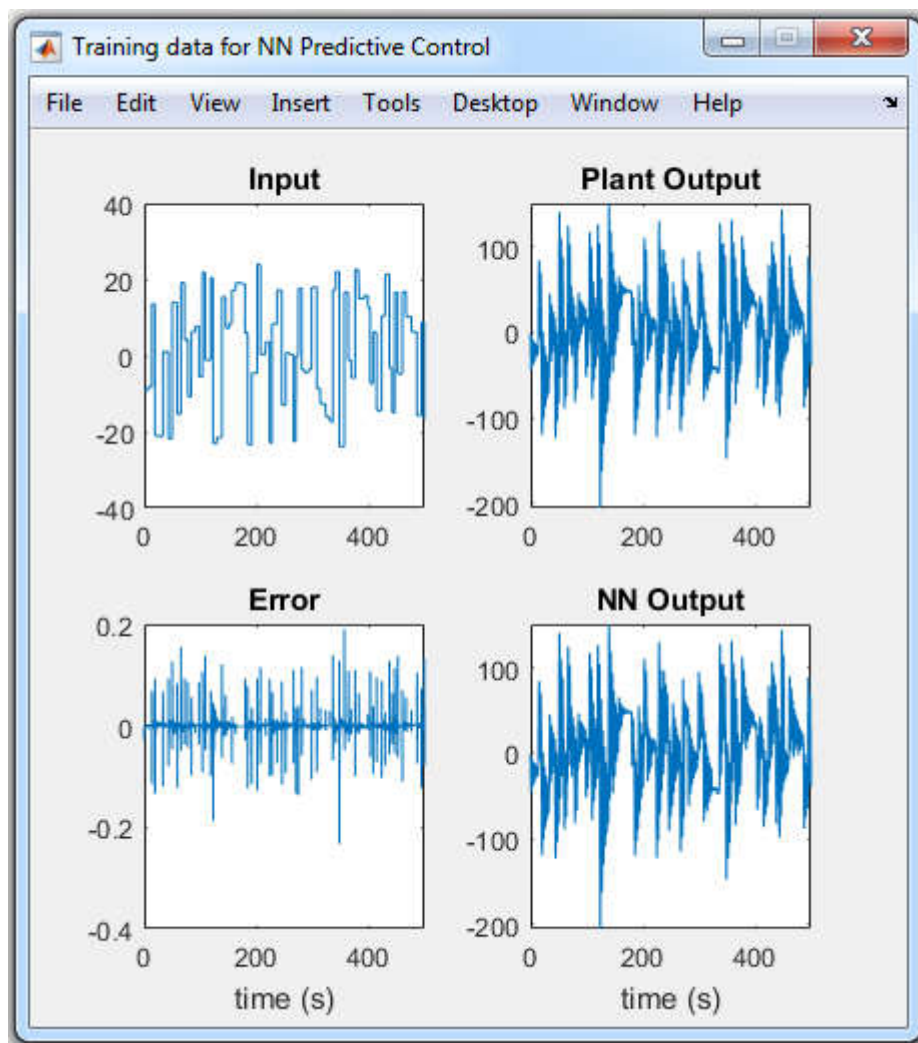


Рисунок 4.18 – Графічне представлення результатів навчання нейронної мережі

Опис вікна Training Data for NN Predictive Control

У вікні **Training Data for NN Predictive Controller** відображаються графіки, що ілюструють процес навчання нейронної мережі та взаємодію моделі з системою:

- **Input (вхідні сигнали)** – графік Input показує вхідні сигнали, які подаються на систему та нейронну мережу під час навчання. Ці сигнали стимулюють модель для генерації відповідей і дозволяють досліджувати її реакцію на різні умови.

- **Plant Output (вихід системи)** – графік Plant Output демонструє фактичні вихідні дані системи (Plant) під час навчання. Він відображає реальну динаміку системи у відповідь на вхідні сигнали, що дозволяє порівняти її поведінку з прогнозами моделі.

- **NN Output (вихід нейромережі)** – графік NN Output відображає прогнозовані вихідні сигнали, згенеровані нейронною мережею під час навчання. Ці дані ілюструють здатність моделі відтворювати поведінку системи на основі поданих вхідних сигналів.

- **Error (помилка)** – графік Error показує різницю між прогнозованими значеннями нейронної мережі (NN Output) та фактичними виходами системи (Plant Output). Цей показник відображає точність моделі та дозволяє оцінити, наскільки адекватно мережа відтворює динаміку системи.

Аналогічні графіки відображаються у вікнах **Validation Data for NN Predictive Control** та **Testing Data for NN Predictive Control** (рис. 4.19, 4.20), які ілюструють результати перевірки моделі на контрольній та тестовій множині даних. Ці графіки дозволяють оцінити здатність нейронної мережі узагальнювати знання, отримані під час навчання, на нові дані.

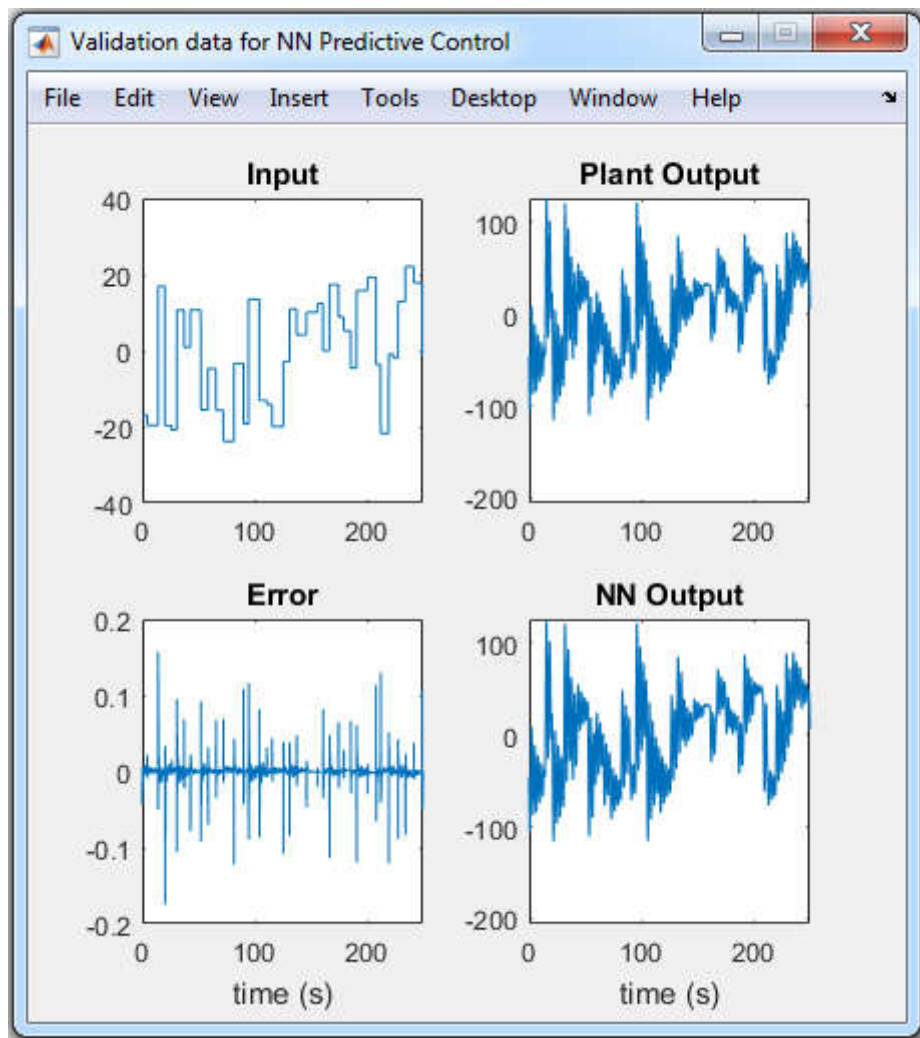


Рисунок 4.19 – Візуалізація результатів перевірки нейронної мережі на контрольній множині

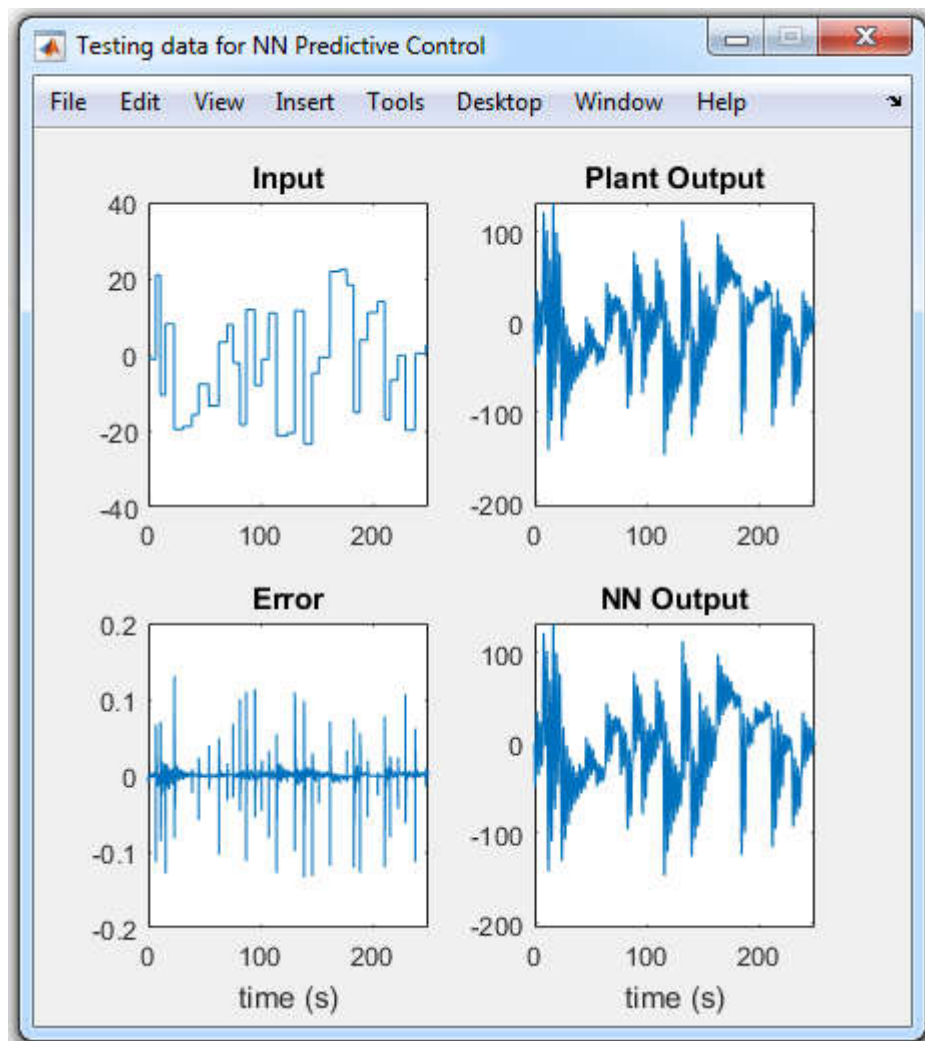


Рисунок 4.20 – Візуалізація результатів перевірки нейронної мережі на тестовій множині

Продовження опису вікна Plant Identification.

Після завершення процесу навчання у вікні **Plant Identification** з'являється повідомлення: *"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 результати, обравши кнопки **OK** або **Apply**).

На цьому етапі М-функція **Nnident** формує динамічну нейромережу **netn2** із заданою кількістю затримок на вході та виході моделі. При цьому попередньо набуті значення вагів і зміщень нейронів зберігаються. Елементи динамічної мережі показані на рис. 4.21.

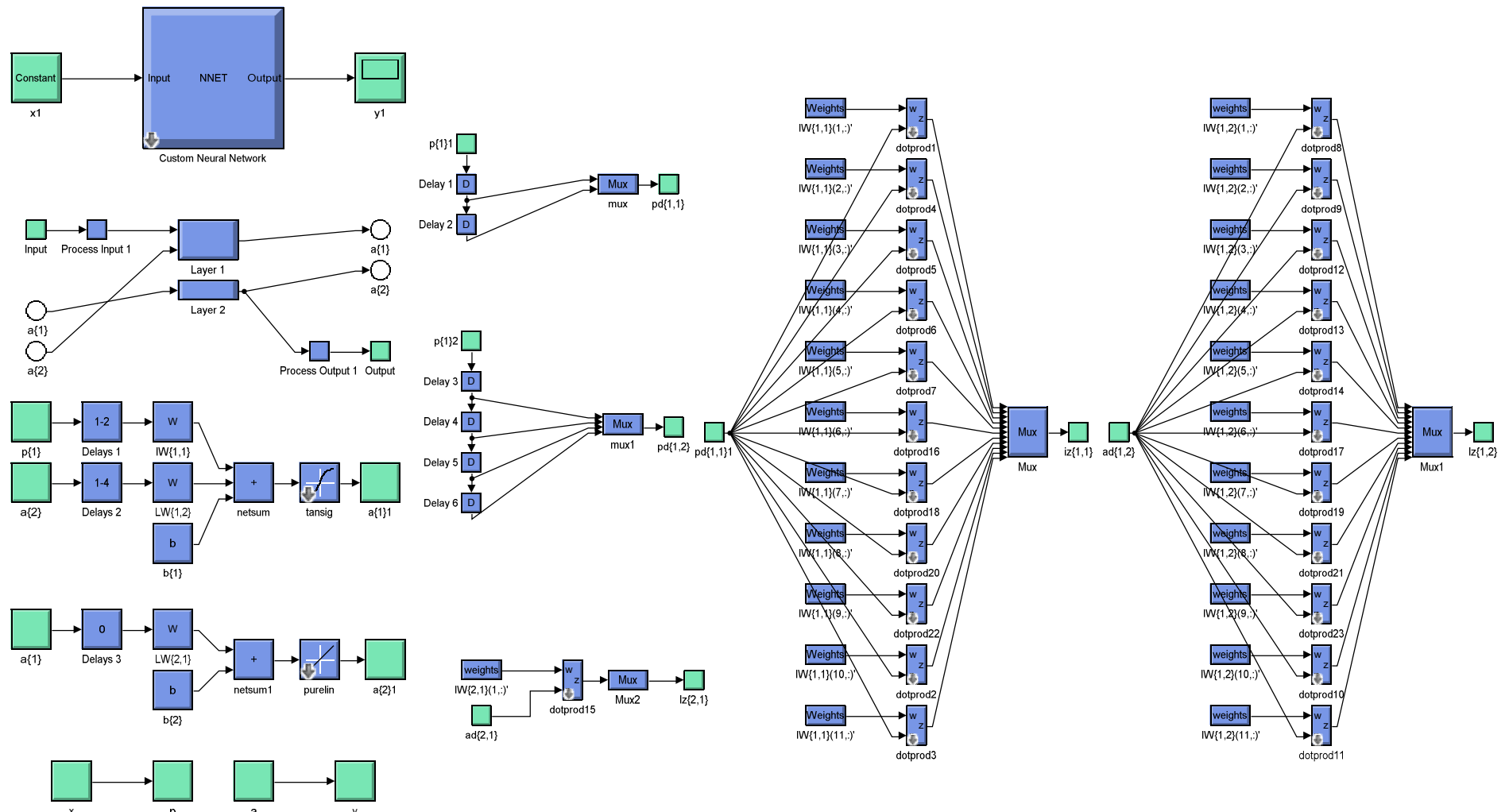


Рисунок 4.21 – Моделі елементів динамічної мережі з блоками затримок у Simulink

Модель динамічної мережі створюється у Simulink за допомогою команди **gensim(netn2)**. Кожен новий елемент стає доступним у окремому вікні після подвійного клацання на попередньому. На основі цих елементів формується повна схема динамічної мережі, показана на рис. 4.22.

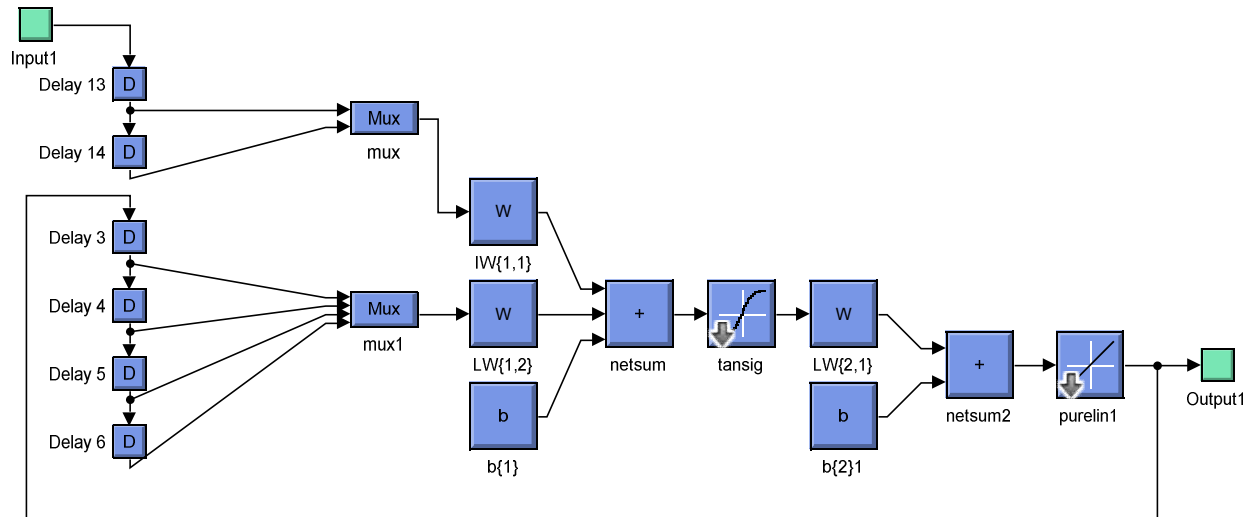


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

Для успішної побудови моделей **netn** і **netn2** необхідно встановити **Breakpoint** у відповідному місці M-функції **Nnident.m**, оскільки після завершення виконання цієї функції подальше формування мереж стає неможливим.

Параметри нейромережевої моделі керованого об'єкта передаються до блоку **NN Predictive Controller** у Simulink. Мережа відображається як структурна схема, показана на рис. 4.23.

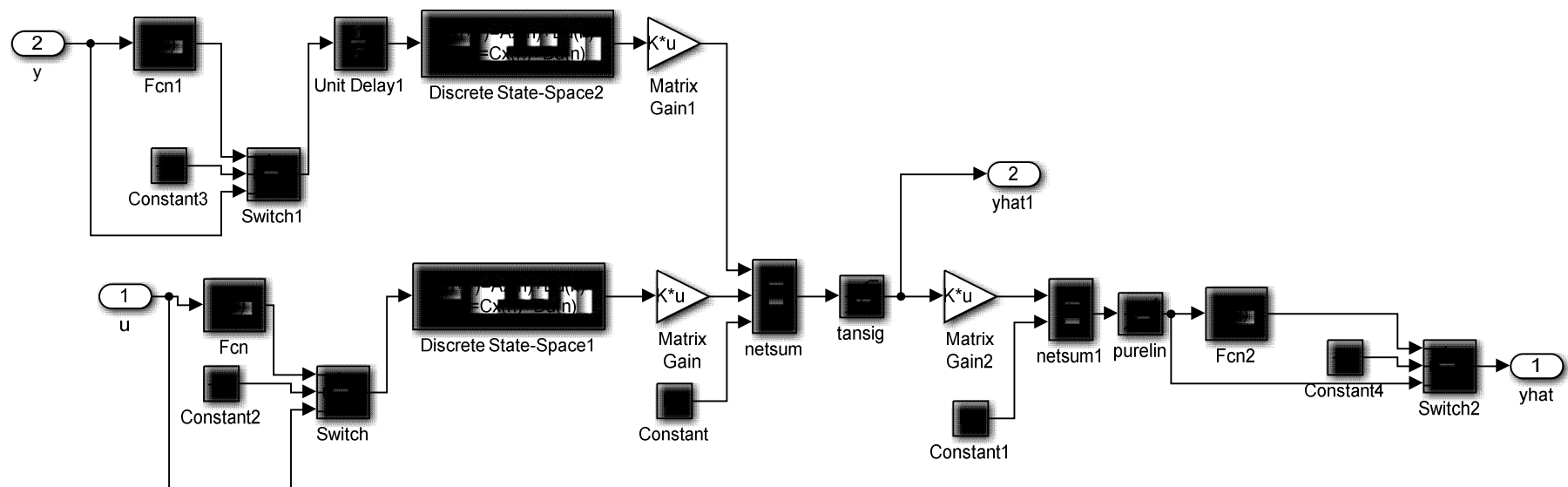


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

Блоки Matrix Gain і Matrix Gain1 відповідають матрицям $IW\{1,1\}$ та $LW\{1,2\}$; блоки Constant (B1) і Constant1 (B2) – це зсуви нейронів першого та другого шарів; елементи затримок реалізовані блоками Discrete State Space 1 та Discrete State Space 2.

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

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

Візьмемо, для прикладу, $N_i = 2$ і $N_j = 5$. МВ цьому випадку матриці $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ і $\mathbf{D}$ мають наступний вид.

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

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

Також у Simulink формується схема **pctest3sim2** (рис. 4.24), яка містить додаткові виходи та використовується М-функцією **predopt** для прогнозування процесу у майбутньому. При активації блоку **Subsystem** відкривається детальна схема, показана на рис. 4.25.

Об'єднана структурна схема нейрорегулятора, що поєднує моделі з рис. 4.6 і рис. 4.23, представлена на рис. 4.26.

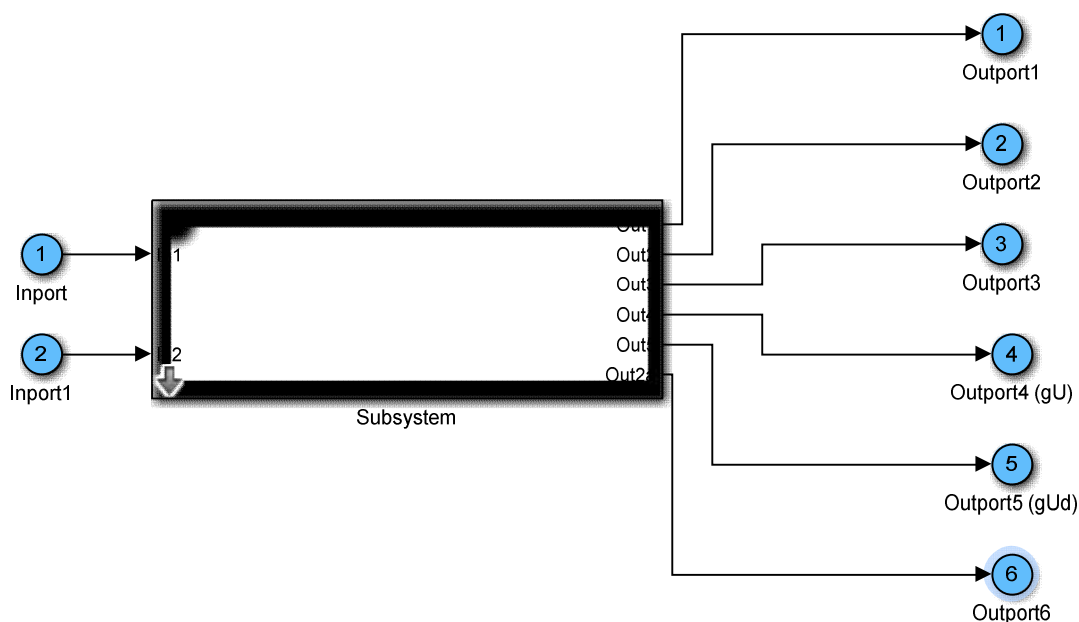


Рисунок 4.24 – Схема ptest3sim2, що використовується для прогнозування процесу функцією predopt

Після формування нейромережевої моделі керованого об'єкта користувач повертається до вікна **Neural Network Predictive Controller** (рис. 4.7), де здійснюються налаштування параметрів оптимізації.

Продовження опису вікна Neural Network Predictive Controller

Const Horizon (N2) – верхня межа підсумовування у функції якості (нижня межа фіксована і дорівнює 1). Цей параметр визначає горизонт прогнозу, тобто кількість кроків у майбутньому, на які алгоритм NNPC робить передбачення.

Збільшення Const Horizon (N2) дозволяє отримати більш далекі випереджальні прогнози, але водночас підвищує обчислювальну складність системи. Вибір оптимального значення цього параметра залежить від специфіки керованого об'єкта та вимог до точності прогнозу.

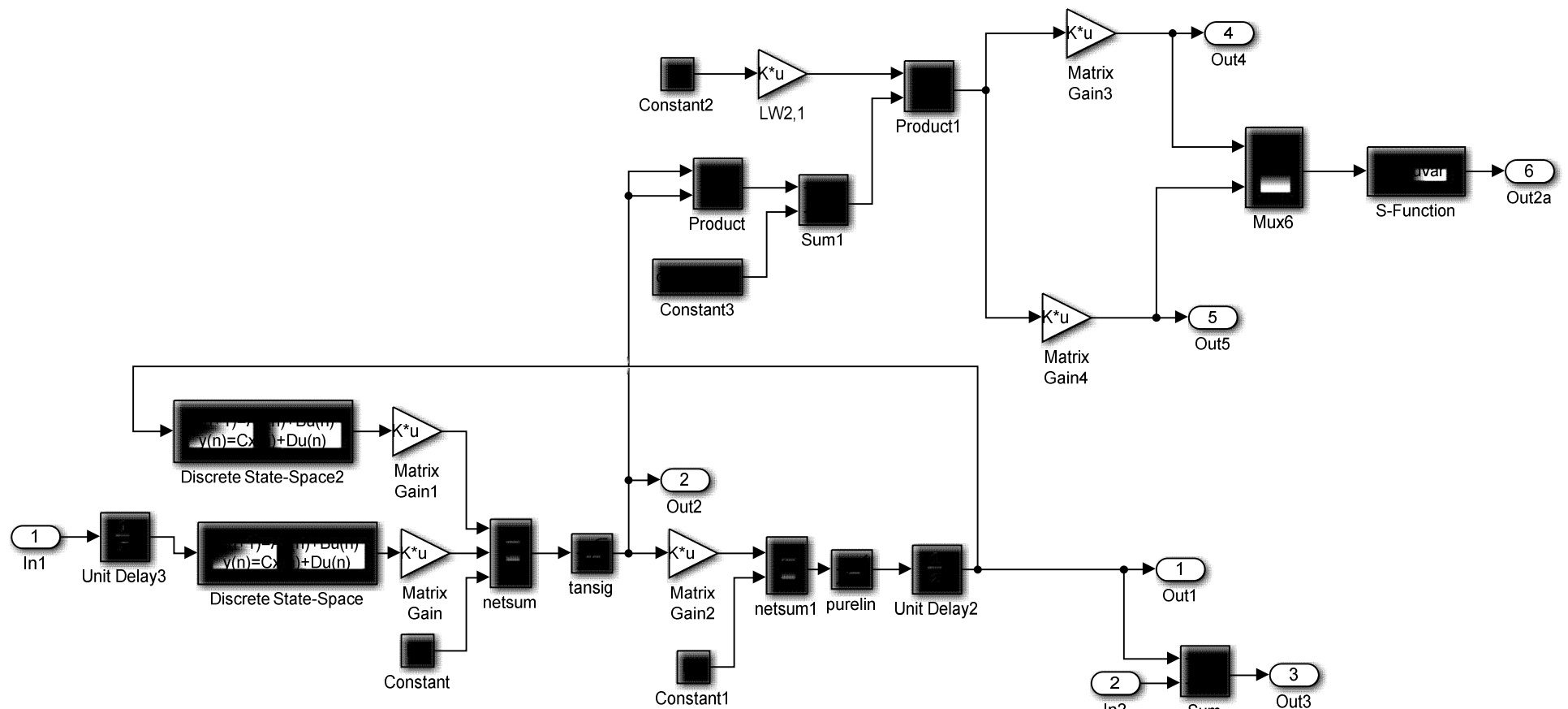


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

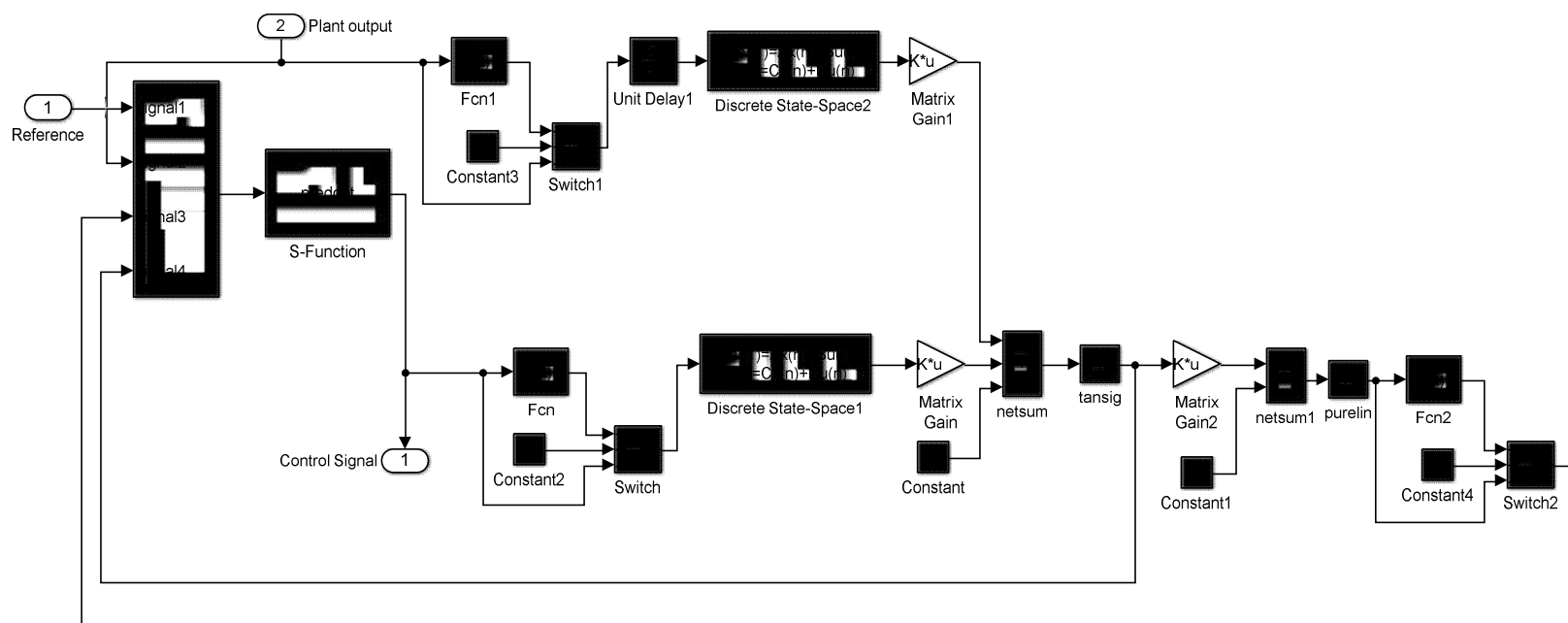


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

Control Horizon (Nu). Параметр **Control Horizon (Nu)** визначає верхню межу підсумовування при оцінці потужності керування. Він безпосередньо впливає на алгоритм прогнозного керування, визначаючи, на скільки кроків у майбутньому контролер буде розраховувати керуючі дії.

Основна ідея цього параметра полягає в тому, що збільшення **Control Horizon** дозволяє алгоритму оптимального керування враховувати більш віддалені майбутні стани системи, що сприяє точнішому досягненню цільових значень та підтриманню стабільності системи. Однак занадто велике значення **Nu** може призвести до підвищення обчислювального навантаження та збільшення вимог до ресурсів комп'ютера. Тому при виборі цього параметра слід враховувати характеристики системи, складність задачі та необхідний рівень прогнозу.

Control Weighting Factor ρ (Коефіцієнт ваги керування). **Control Weighting Factor** визначає вагу або важливість мінімізації керуючих сигналів у функції витрат контролера. Іншими словами, цей параметр дозволяє налаштувати, наскільки сильно контролер буде прагнути робити керуючі дії економічними або «дешевими» щодо витрат на управління.

Якщо коефіцієнт ρ встановлено великим, контролер надає пріоритет зменшенню амплітуди керуючих сигналів, що може бути корисно у випадках, коли управління дороге коштує або є фізичні обмеження. Зменшення значення коефіцієнта ρ призводить до того, що контролер більше концентрується на досягненні цільових значень системи, а економія керуючих сигналів стає менш пріоритетною.

Налаштування **Control Weighting Factor** дозволяє знайти оптимальний баланс між ефективністю керування та вартістю керуючих дій, що особливо важливо при застосуванні нейромережевого прогнозного контролю, де одночасно враховуються точність прогнозу і обмеження на ресурси управління.

Search Parameter α (Параметр пошуку). **Search parameter** – це параметр одновимірного пошуку, який задає поріг допустимого зменшення показника якості. Він використовується для контролю процесу оптимізації та визначає момент завершення пошуку, коли подальше покращення якості стає незначним.

Одновимірний пошук – це метод оптимізації, при якому оптимізується лише один параметр, тоді як інші залишаються фіксованими. У контексті нейромережевого прогнозного контролера цей метод допомагає налаштувати окремі параметри контрольної стратегії, що безпосередньо впливають на ефективність керування.

Поріг зменшення показника якості визначає мінімальне значення, яке повинен мати показник покращення, щоб вважати процес оптимізації завершеним. Коли приріст або зменшення показника якості стає меншим за цей поріг, алгоритм припиняє пошук, а отримані параметри вважаються оптимальними. Такий підхід дозволяє заощадити обчислювальні ресурси та прискорити збіжність оптимізації.

Minimization Routine (Процедура мінімізації). Minimization Routine визначає алгоритм оптимізації, який використовується для налаштування параметрів контрольної стратегії. Вибір конкретного методу оптимізації має значний вплив на точність і ефективність роботи контролера.

При виборі процедури враховуються особливості завдання керування, наявні обмеження, а також доступні обчислювальні ресурси. Ретельний підбір алгоритму дозволяє досягти оптимального балансу між точністю розрахунку керуючих сигналів і швидкістю обчислень.

Iterations Per Sample Time (Ітерації на крок часу). Параметр Iterations Per Sample Time визначає кількість ітерацій оптимізації, які виконуються протягом одного кроку часу системи (Sample Time). Він впливає на точність розрахунку оптимального керування та швидкість реакції контролера на зміни стану системи.

Збільшення кількості ітерацій дозволяє отримати більш точний розрахунок керуючих дій, проте підвищує обчислювальне навантаження. Навпаки, занадто мала кількість ітерацій може призвести до недостатньої точності оптимізації та втрати ефективності керування.

Оптимальне значення цього параметра зазвичай підбирається експериментально, виходячи з характеристик конкретної системи керування та обмежень

обчислювальних ресурсів. Воно забезпечує баланс між точністю оптимізації та швидкістю реакції контролера.

Після налаштування всіх параметрів оптимізації вони передаються в блок **NN Predictive Controller** у середовищі Simulink, де використовуються для розрахунку оптимальних керуючих сигналів під час роботи системи.

4.4 Вибір параметрів нейрорегулятора NN Predictive Controller

При проектуванні нейрорегулятора відбувається варіювання параметрів Const Horizon (N_2), Control Horizon (N_u) та Control Weighting Factor (ρ) (за умови, що нижня межа Const Horizon залишається зафіксованою і дорівнює 1), а також параметра одновимірного пошуку Search Parameter α , який визначає поріг допустимого зменшення показника якості, та кількість ітерацій оптимізації на один крок дискретності. Додатково обирається метод проведення одновимірного пошуку.

Дослідження показали, що параметри Const Horizon, Control Weighting Factor та обрана процедура пошуку мають незначний вплив на кінцеві результати синтезу регулятора. Тому для даної задачі були обрані значення: $N_u=3$, $\rho=0,05$, $\alpha=0,001$, а метод одновимірного пошуку – `csrchbas`.

Водночас значення Control Horizon (N_u) і кількість ітерацій на такт дискретності γ значно впливають на продуктивність контролера. Збільшення цих параметрів дозволяє підвищити точність керування, проте істотно зростає обчислювальне навантаження на кожному кроці. Для розв'язуваної задачі оптимальні значення знаходяться в межах $N_2 = 20 \div 30$, $\gamma = 2 \div 4$

При ідентифікації об'єкта керування критично важливим є вибір кількості нейронів у прихованому шарі N . Недостатня кількість нейронів унеможливило виконання поставлених завдань, тоді як надмірна – призводить до перенавчання та збільшення часу обчислень. Для розглядуваної системи оптимальне значення $N = 7 \div 12$, при цьому середньоквадратична помилка на навчальних, валідацій-

них (перевірочних) та тестових наборах даних залишалася в межах $10^{-3} \div 10^{-5}$, а миттєві помилки не перевищували 0,2.

Якість навчання мережі безпосередньо залежить від довжини навчальної вибірки N_b та дискретного кроку часу Δt , який визначає інтервал між двома послідовними точками збору даних. Оптимальні значення для даної системи: $N_b = 20000$, $\Delta t = 0,05 \div 0,1$ с. Збільшення Δt зменшує точність обчислень і скорочує різницю між помилкою на навчальному наборі та помилками на контрольній і тестовій множинах. Зменшення Δt потребує подовження навчальної вибірки N_b , що суттєво збільшує час навчання без значного зниження похибки.

Для формування репрезентативної вибірки важливо правильно встановити мінімальні та максимальні межі інтервалу ідентифікації, що залежать від динамічних характеристик об'єкта Subsystem. У даній роботі вони обрані як $t_{\min} = 4 \div 8$ с, $t_{\max} = 10 \div 20$ с.

При побудові нейромережевої моделі задається кількість затримок на вході z_1 та виході z_2 моделі. Найкращі результати були досягнуті при $z_1 = 2$, $z_2 = 2 \div 5$.

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

Для навчання мережі використано алгоритм Levenberg-Marquardt (trainlm), який забезпечує ефективну оптимізацію вагових коефіцієнтів і швидке зближення до мінімуму функції втрат.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором **NN Predictive Controller**

Побудова графіків перехідних процесів двомасової системи, керованої нейрорегулятором **NN Predictive Controller**, виконувалась з використанням моделей, створених в середовищі Simulink, що представлені на рис. 4.5 та 4.27.

На рис. 4.28–4.31 наведено графіки перехідних процесів основних змінних стану синтезованої системи. Аналіз отриманих результатів показує, що система демонструє задовільну реакцію на ступінчасті впливи з випадковими амплітудами. Перехідні процеси мають коливальний характер із невеликим перерегулюванням, що свідчить про стабільність та адекватність регулятора.

Таким чином, синтезований нейромережевий регулятор з прогнозом **NN Predictive Controller** може бути ефективно використаний для керування двомасовою електромеханічною системою, розглянутою в даній роботі.

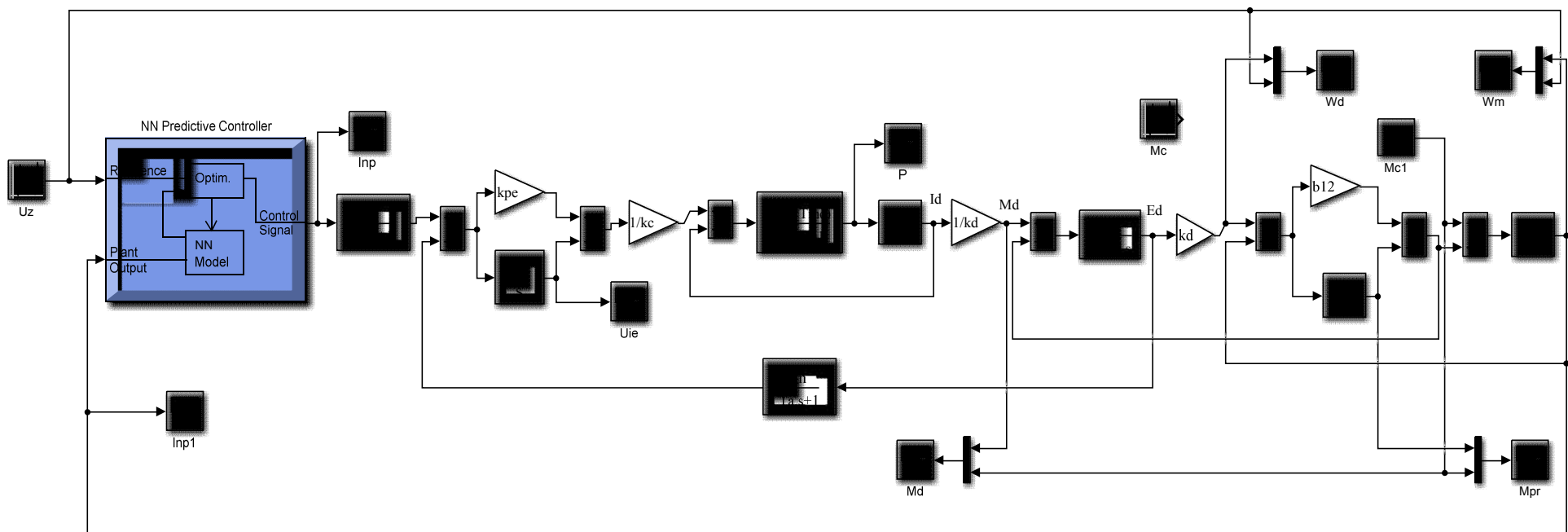
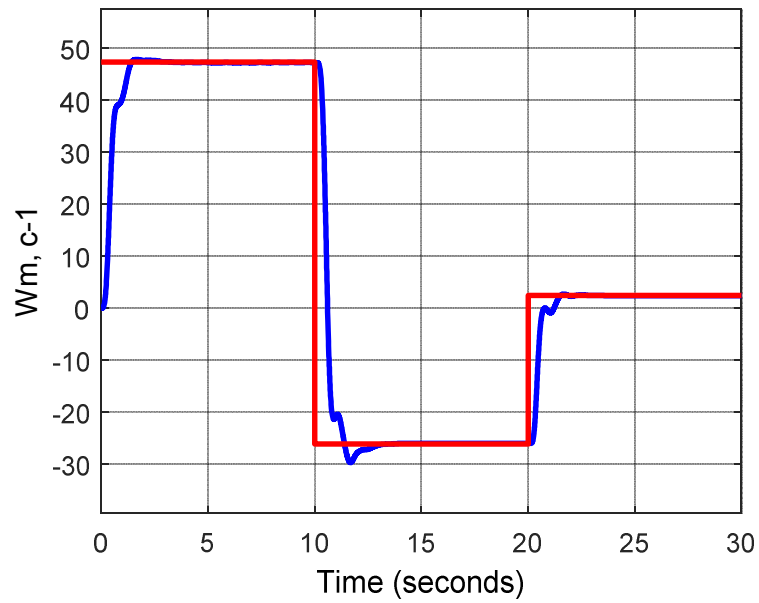
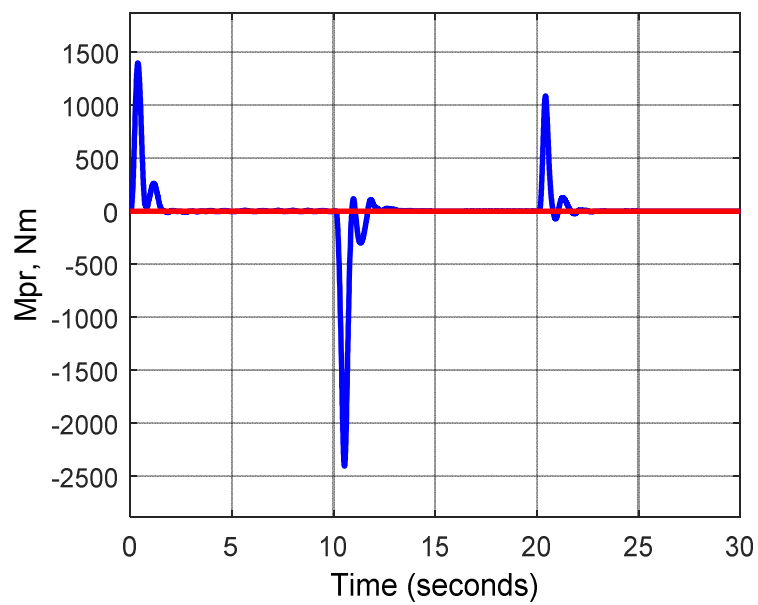


Рисунок 4.27 – Імітаційна модель двомасової системи з нейрорегулятором



a)



б)

Рисунок 4.28 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією задаючої величини:

а) швидкість механізму $\omega_m = f(t)$, б) момент пружності $M_{пр} = f(t)$

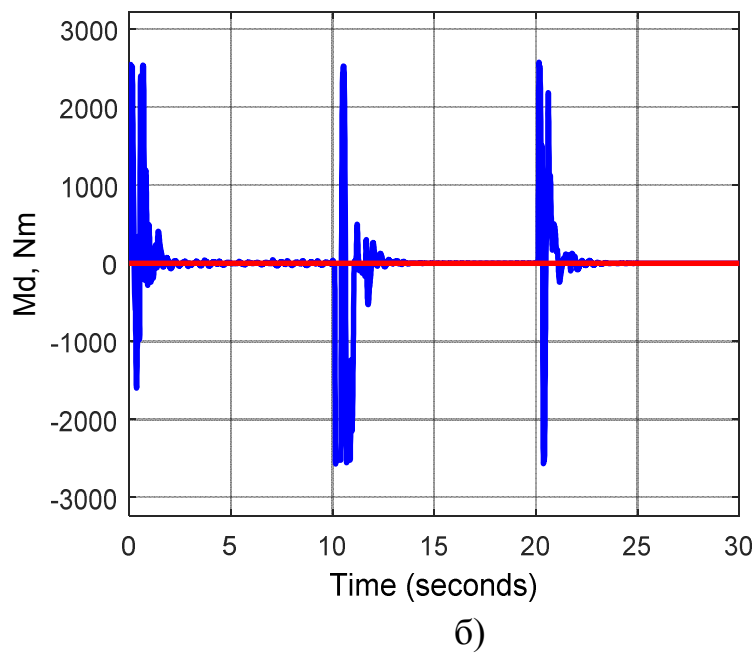
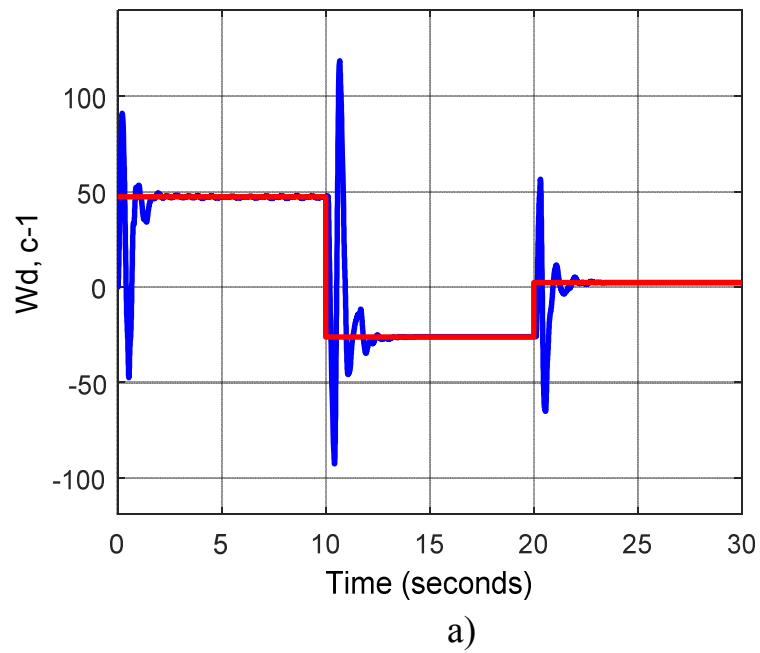
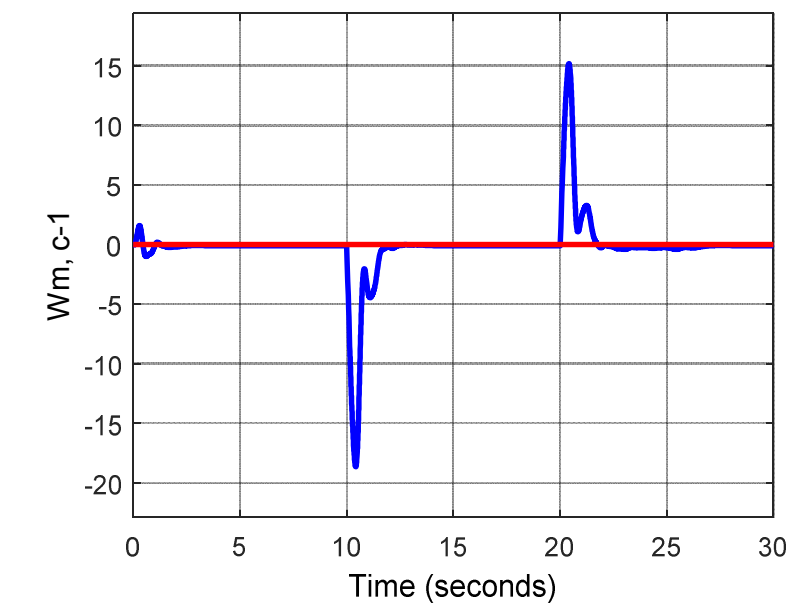
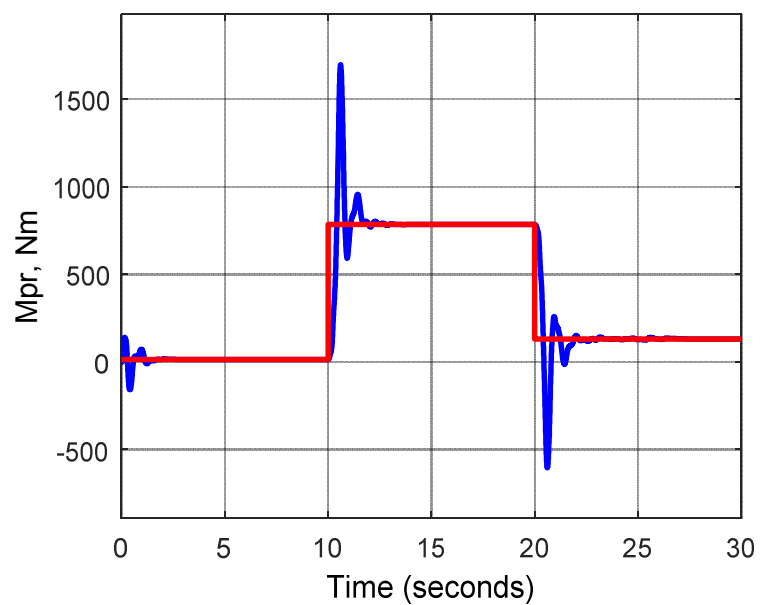


Рисунок 4.29 – Динаміка двигуна двомасової системи з нейрорегулятором під впливом задаючої величини:

а) швидкість обертання двигуна $\omega_d = f(t)$, б) момент двигуна $M_d = f(t)$



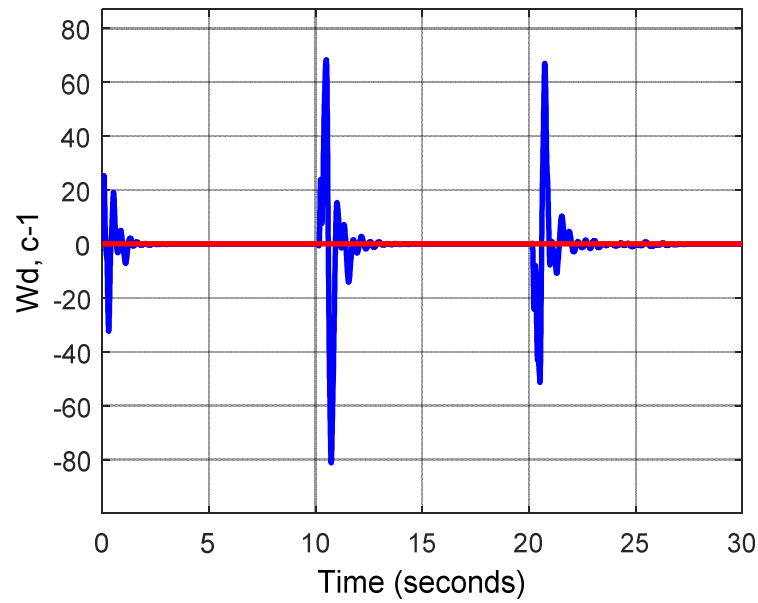
а)



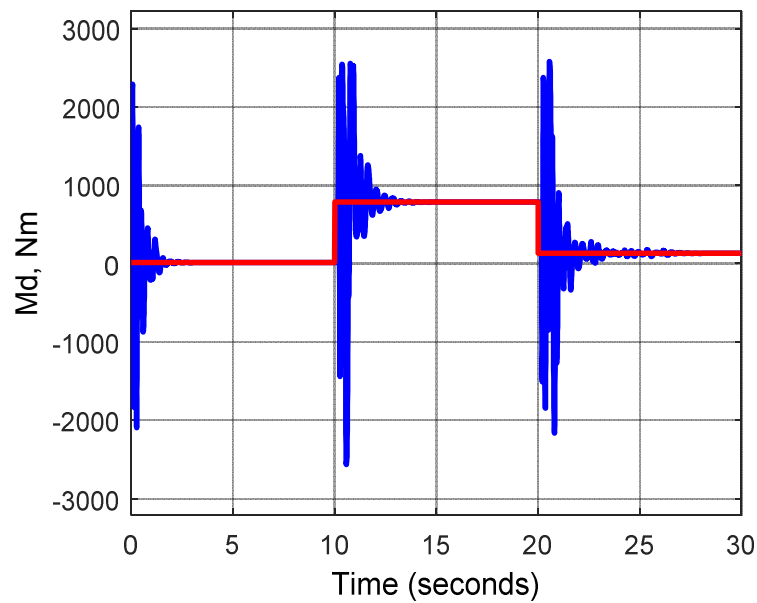
б)

Рисунок 4.30 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією зовнішнього збурення:

а) швидкість механізму $\omega_m = f(t)$, б) момент пружності $M_{пр} = f(t)$



а)



б)

Рисунок 4.31 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією зовнішнього збурення:

а) швидкість механізму $\omega_m = f(t)$, б) момент пружності $M_{np} = f(t)$

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проведено аналіз доступних у пакеті Neural Network Toolbox системи MATLAB методів синтезу нейрорегуляторів, зокрема регулятора з прогнозуванням (NN Predictive Controller), регулятора на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) та регулятора, що використовує еталонну модель (Model Reference Controller). Враховуючи специфіку задачі стабілізації роботи рольганга, у якості основного засобу керування було обрано нейрорегулятор з прогнозом – NN Predictive Controller. Проаналізовано структури зазначених регуляторів та послідовність їх синтезу.

2. У середовищі SIMULINK MATLAB побудовано систему керування з використанням нейрорегулятора NN Predictive Controller. До складу системи входить модель керованого об'єкта – двомасова модель приводу рольганга перед ножицями стана гарячої прокатки – та блок предиктивного нейронного регулятора.

3. На основі моделі виконавчого пристрою сформовано навчальну вибірку шляхом варіювання параметрів навчальної послідовності. Це дало змогу забезпечити якісне тренування нейронної мережі. Визначено оптимальні параметри навчання: довжина вибірки – 10000 точок; інтервал між сусідніми вибірками – 0,05 с; мінімальний та максимальний інтервали ідентифікації – 4 с і 10 с відповідно.

4. Розроблено модель двомасової системи у вигляді двошарової нейронної мережі прямого поширення сигналу та виконано її навчання за алгоритмом Левенберга–Марквардта. Встановлено оптимальну кількість нейронів прихованого шару. Помилка навчання має порядок 10^{-4} , а миттєві помилки на навчальній, контрольній і тестовій множинах не перевищують 0,2, що свідчить про високу точність ідентифікації.

5. Виконано синтез предиктивного нейрорегулятора (NN Predictive Controller). Завдяки широкому діапазону зміни параметрів регулятора визначено ті з них, які суттєво впливають на якість керування, та знайдено їх оптима-

льні значення. Використання високоточної нейромережевої моделі об'єкта забезпечило формування нейрорегулятора з відмінними показниками якості.

6. Проведено моделювання роботи нейромережевої системи керування з синтезованим регулятором. Аналіз результатів показав, що перехідні процеси у системі під час пускових режимів та при дії навантажень мають стабільний, задовільно згладжений характер. Система є астатичною, перерегулювання не перевищує 5 %, а час регулювання становить близько 3 с, що відповідає технічним вимогам задачі.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розв'язано актуальну науково-прикладну задачу створення нейромережевої системи керування багатомасовою електромеханічною системою механізму рольганга за ножицями стана гарячої прокатки. Розроблена система забезпечує підвищену якість регулювання та покращені динамічні властивості приводу. Основні результати дослідження зводяться до такого.

1. На основі аналізу існуючих систем керування рольгангами гарячої прокатки та вимог до сучасних електромеханічних систем доведено доцільність і перспективність використання нейромережевих технологій для керування багатомасовою механічною системою. Обґрунтовано, що застосування штучних нейронних мереж дозволяє суттєво підвищити точність та адаптивність процесів регулювання.

2. Зурахуванням характеристик допоміжних механізмів прокатних станів та умов роботи електроприводів рольгангів вибрано електропривод постійного струму системи ТП-Д. Виконано перевірку електродвигуна на відповідність умовам нагріву та перевантажувальної здатності в робочих режимах відповідно до вимог технологічного процесу.

3. Проведено вибір та розрахунок системи автоматичного регулювання параметрів електроприводу. Для керування координатами приводу застосовано систему підлеглого регулювання, що включає внутрішній струмовий контур та зовнішній контур ЕРС. Виконано розрахунок параметрів силового кола електроприводу та синтез послідовних коригуючих ланок: струмовий контур оптимізовано за модульним критерієм, а контур ЕРС – за симетричним критерієм. Ефективність розробленої структури підтверджено моделюванням у середовищі MATLAB/Simulink.

4. Створено математичну модель двомасової системи електроприводу, що враховує пружність механічних з'єднань. Результати комп'ютерного моделювання показали, що в такій системі ключові координати мають коливальний ха-

рактар перахiдних процесiв, що погiршує якiсть регулювання i пiдтверджує не-
обхiднiсть удосконалення структури керування.

5. Виконано синтез нейромережевого предиктивного регулятора NN Predictive Controller. В ходi параметричного аналізу визначено основнi параме-
три, що впливають на якiсть регулювання, та знайдено їх оптимальнi значення.
Завдяки використанню високоточної нейромережевої моделi об'єкта забезпече-
но можливiсть побудови нейрорегулятора, який значно покращує динамiчнi ха-
рактеристики системи.

6. Проведено моделювання роботи електроприводу з розробленим нейро-
регулятором. Аналiз результатiв пiдтвердив, що система iз застосуванням NN
Predictive Controller забезпечує якiснi перахiднi процеси: перерегулювання
швидкостi механiзму не перевищує 5 %, а час регулювання становить близько
3 с. Отриманi показники вiдповiдають технiчним вимогам та демонструють пе-
реваги нейромережевого пiдходу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Штучні нейронні мережі: навчальний посібник / С. В. Ткаліченко. – Кривий Ріг: Державний університет економіки і технологій, 2023. –150 с.
2. 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.
3. 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.
4. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах:підручник..– Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
5. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.
6. Штучні нейронні мережі: навчальний посібник для студентів вищих навчальних закладів / О.Г.Руденко, Є.В. Бодянський. – К: Компанія СМІТ, 2006, 404 с.
7. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.
8. Тимошук П.В. Штучні нейронні мережі; Навч. посібн. - Львів: Львівська політехніка, 2011. -444 с.
9. Кононюк, А. Ю. .Нейронні мережі і генетичні алгоритми. - Київ: Корнійчук, 2008. - 468 с.
10. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах:підручник..– Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
11. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.

12. Штучні нейронні мережі: базові положення Навчальний посібник Електронне мережне навчальне видання / І.А. Терейковський, Д.А. Бушуєв, Л.О. Терейковська, Київ: КПІ ім. Ігоря Сікорського, 2022. - 123 с.
13. Pyzer-Knapp E. O., Ziatdinov M. A. *Foundation models for materials discovery – current state and future prospects* : наук. стаття // *Nature Computational Materials*. – 2025. – Режим доступу: <https://www.nature.com/articles/s43588-025-00967-x>
14. Lan J. *Efficient model predictive control for nonlinear systems modelled by deep neural networks* : e-print (arXiv) / J. Lan. – 2024. – Режим доступу: <https://arxiv.org/abs/2405.10372>
15. *A Survey on State-of-the-art Deep Learning Applications* : e-print (arXiv). – 2024. – Режим доступу: <https://arxiv.org/abs/2403.17561>
16. China's "Darwin Monkey" is the world's largest brain-inspired supercomputer : онлайн-публікація / Live Science. – 2025. – Режим доступу: <https://www.livescience.com>
17. Doshi-Velez F., Kim B. *Towards a Rigorous Science of Interpretable Machine Learning* : e-print (arXiv). – 2017. – Режим доступу: <https://arxiv.org/abs/1702.08608>.
18. Thirunavukarasu R. *A comprehensive review on transformer network for natural language processing and vision tasks* : журн. стаття / R. Thirunavukarasu. – ScienceDirect, 2024. – Режим доступу: <https://doi.org/10.1016/j.inffus.2024.05.012>
19. *Neuromorphic Computing: Innovations and Future Prospects* : оглядова стаття / авт. кол. – ResearchGate, 2024. – Режим доступу: <https://www.researchgate.net/publication/380674321>
20. Camacho E. F., Bordons C. *Model Predictive Control* : монографія. – Springer, 2013. – 408 с
21. Narendra K. S., Parthasarathy K. *Identification and Control of Dynamical Systems Using Neural Networks* // *IEEE Transactions on Neural Networks*. – 1990. – Vol. 1, No. 1. – P. 4–27.
22. Chen W., Jiang Y. *Neural Model Predictive Control: A Review of Recent Advances and Applications* // *Control Engineering Practice*. – 2023. – Vol. 134. – P. 105465
23. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction* : монографія. – 2nd ed. – MIT Press, 2018. – 552 с
24. Astrom K J , Wittenmark B *Adaptive control*, Reading MA Addison Wesley, 1989.

25. Parks P C Lyapunov redesign of model reference adaptive control systems, IEEE Journal Transactions of Automatic Control, 1966, vol 11, pp 362-367.
26. Hornik K , Stinchcombe M , White H Mylti-layer feed-forward networks are universal approximators Discussion paper, Department of Economics, University of California, San Diego, La Jolla, CA, 1988.
27. Haykin S , Neural Networks A comprehensive foundation, MacMillan College Publishing Co , New York, 1994.
28. Heermann P D Neural network techniques for stable learning control of nonlinear systems Dissertation D S University of Texas at Austin, 1992.
29. Narendra K S , Parthasarathy K Neural networks and dynamical systems Part 1 A gradient approach to Hopfield networks TR 8820, Center for systems science, Department of Electrical Engineering, Yale University, New Haven, CT, 1988.
30. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
31. Werbos P J , Beyond Regression New Tools for Prediction and Analysis in the Behavioral Sciences, Phd Thesis, Dept of Applied Mathematics, Harvard University, Cambridge, Mass , 1974.
32. Ровітхакіс Г. А.: «Stable adaptive neuro-control design via Lyapunov function derivative estimation» // Automatica. – 2001. – Vol. 37, No. 8. – P. 1213–1221.
[ResearchGate+3ResearchGate+3dl.acm.org+3](#)
33. Tenavi Nakamura-Zimmerer, Qi Gong, Wei Kang: «Neural network optimal feedback control with guaranteed local stability» // arXiv preprint. – 2022. – ID arXiv:2205.00394
34. Gao, Q., Li, H., & Sun, Y. Adaptive neural network-based control for a class of nonlinear pure-feedback systems with time-varying full state constraints // Journal of Applied Science & Engineering. – 2018. – Vol. 11, No. 3. – P. 203-218.
35. Park, J. W., Harley, R. G., Venayagamoorthy, G. K. Indirect adaptive control for synchronous generator: Comparison of MLP/RBF neural networks approach with Lyapunov stability analysis // IEEE Trans. Neural Networks. – 2004. – Vol. 15, No. 2. – P. 460-464. [Yonsei University](#)

36. Xiong, Z., Li, J., & Sun, C. Fault-tolerant control for nonlinear systems using sliding mode and adaptive neural network estimator // *Soft Computing*. – 2020. – Vol. 24. – P. 11535-11544. [SpringerLink](#)
37. Fu, X., Li, S., Wunsch, D. C., Alonso, E. Local stability and convergence analysis of neural network controllers with error integral inputs // *Elec. & Comp. Engineering Res. & Creative Works*. – 2018. – P. (выпуск)-(сторінки). [Scholars' Mine](#)
38. Garcia C.E. Model predictive control: theory and practice a survey / Garcia C.E., Prett D.M., Morari M. // *Automatica*. – 1989. – Vol.25. – P.335-348.
39. Recurrent Neural Network-Based Model Predictive Control for Continuous Pharmaceutical Manufacturing (Wong W.C., Chee E., Li J., Wang X.) – *Mathematics*, 2018, 6(11):242. Розглядається MPC із використанням рекурентної нейронної мережі (RNN) для ідентифікації динаміки системи у фармацевтичному безперервному виробництві. [MDPI](#)
40. Neural Network Based Model Predictive Control (Piche S., Keeler J.D., Martin G., Boe G., Johnson D., Gerules M.) – *Advances in Neural Information Processing Systems NIPS'99*, 1999. Застосування нейронної мережі для моделювання нелінійної динаміки та інтеграція з MPC-підходом. [NeurIPS Papers](#)
41. Review on model predictive control: an engineering perspective – *The International Journal of Advanced Manufacturing Technology*, 2021, Vol. 117, pp. 1327-1349. Оглядова стаття, яка досліджує MPC – теорію, практику, стійкість, обмеження та впровадження. [SpringerLink](#)
42. Efficient model predictive control for nonlinear systems modelled by deep neural networks – 2024, e-print arXiv/preprint. Розробка MPC для нелінійних систем, модельованих глибокими нейронними мережами; увага до оптимізації й обмежень вводу/стану.
43. Neural networks for control / eds. W. T. Miller, R. S. Sutton, P. J. Werbos. – Cambridge, MA : The MIT Press, 1990. – 537 p.
44. Solowey D. Neural Generalized Predictive Control / Solowey D., Haley P.J // *Proceeding of the 1996 IEEE Symposium on Intelligent Control*. – 1996. – P. 277–282.

45. Clarke D.W., Mohtadi C., Tuffs C. Generalized Predictive Control - Part 1: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 137-148.
46. Clarke D. W., Mohtadi C., Tuffs C. Generalized Predictive Control. Part 2: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 149-163.
47. Clarke D. W. Advances in model-based predictive control.- In *Advances in Model-Based Predictive Control* / D. W. Clarke: Oxford University Press. – 1994. – 274
48. Montague G. A., Willis M. J., Tham M. T., Morris A. J. Artificial Neural Network Based Control // *Int. Conference on Control*. – 1991. – Vol.1. – P. 266-271.
49. Takahashi Y. Adaptive Predictive Control of Nonlinear Time-Varying System using Neural Network // *IEEE International Conference on Neural Networks*, 1993. – Vol.3. – P.1464-1468
50. Jeong Jun Song, Sunwon Park. Neural Model-Predictive Control for Nonlinear Chemical Processes // *Journal of Chemical Engineering of Japan*. – 1993. – Vol.26. – №4. –P.347-354.
51. Psychogios D. C., Ungar L. H. Nonlinear Internal Model Control and Model Predictive Control using Neural Networks // *5th IEEE Int. Symposium on Intelligent Control*. – 1990. – P.1082-1087.
52. Press W. H., Flannery B. P., Teukolsky S. A., Vetterling V. T. *Numerical Recipes.- The Art of Scientific Computing*. Cambridge University Press, 1988. – 452.
53. Rumelhart D. E., McClelland J. L. and the PDP Research Group. *Parallel Distributed Processing*. The MIT Press. – 1986. – Vol.1. – Chapter 8. – 337 p.
54. Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.
55. Теорія автоматичного управління : навчальний посібник [Електронний ресурс] : навч. посіб. / О. Й. Штіфзон, П. В. Новіков, В. П. Бунь. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 144 с.
56. Теорія автоматичного управління. Нелінійні та дискретні системи [Електронний ресурс] : навчальний посібник / О. Й. Штіфзон, П. В. Новіков. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 98 с.

57. Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.
58. Теорія автоматичного керування : навчальний посібник / О. К. Аблесімов – К. : «Освіта України», 2019. – 270 с.
59. Навчальний посібник з дисципліни "Теорія автоматичного керування" : у 2 ч. Ч. 1 / А. П. Гуров, С. І. Ольшевський, О. О. Черно, Л. І. Бугрім. – Миколаїв : НУК, 2018. – 111 с.
60. Теорія автоматичного управління : курс лекцій / Т. М. Боровська. – Вінниця : ВНТУ, 2018. – 256 с.
61. Теорія автоматичного управління: Навчальний посібник [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології кібер-енергетичних систем»; уклад.: О. Й. Штіфзон, П. В. Новіков, В.П. Бунь. – Електронні текстові дані (1 файл: 2,2 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 144 с
62. Лістровий С. В., Мірошник М. А., Клименко Л. А. Теорія автоматичного керування, штучний інтелект і автоматизація процесу прийняття рішення: Навч. посібник. – Харків: УкрДУЗТ, 2019. – 120 с.
63. Автоматизований електропривод. Розділ 1. Теорія електропривода. Навчальний посібник. / Н.Д. Красношопка – К.: КПІ ім. Ігоря Сікорського. - 2023. – 101 с.
64. Возняк, О.М., Штуць. А.А., Колісник М.А. Сучасні системи електроприводів. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ТВОРИ, 2021. – 280 с.
65. А. А. Видмиш, Л. В. Ярошенко. Основи електропривода. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ВНАУ, 2020. – 387 с.
66. Панкратов А.І. Системи керування електроприводами. Видання 2: Навч. посібник з дисципліни «Системи керування електроприводами» (для студентів спеціальності 151 «Автоматизація та комп'ютерноінтегровані технології» ден-

ної і заочної форми навчання) – Краматорськ: ДДМА, 2018. – 225 с.

67. Колб Ант. А, Колб А. А. Теорія електроприводу: Навчальний посібник. – 2-е вид. перероб. і доп. – Д., Національний гірничий університет, 2011. – 540 с.

68. Панкратов А.І. Системи керування електроприводами. Видання 2: Навч. посібник з дисципліни «Системи керування електроприводами» (для студентів спеціальності 151 «Автоматизація та комп'ютерноінтегровані технології» денної і заочної форми навчання) – Краматорськ: ДДМА, 2018. – 225 с.

69. Василега П. О. Електропривод робочих машин : підручник / П. О. Василега. – Суми: Сумський державний університет, 2022. – 290 с. ISBN 978-966-657-900-6.

70. Шефер О.В. Автоматизований електропривод загальнопромислових механізмів: конспект лекцій. – Полтава: ПолтНТУ, 2020. – 154 с.

71. Електропривод виробничих машин і механізмів: Навчальний посібник / О.Ю. Синявський, В.В. Савченко, В.Я. Бунько, В.Ю. Рамш; За ред. О.Ю. Синявського. – К.: ФОП Ямчинський О.В., 2020. – 444 с. ISBN 978-617-7890-88-0.

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

73. 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

БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ

А.1 Класифікація штучних нейронних мереж

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

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

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

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

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

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

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

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

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

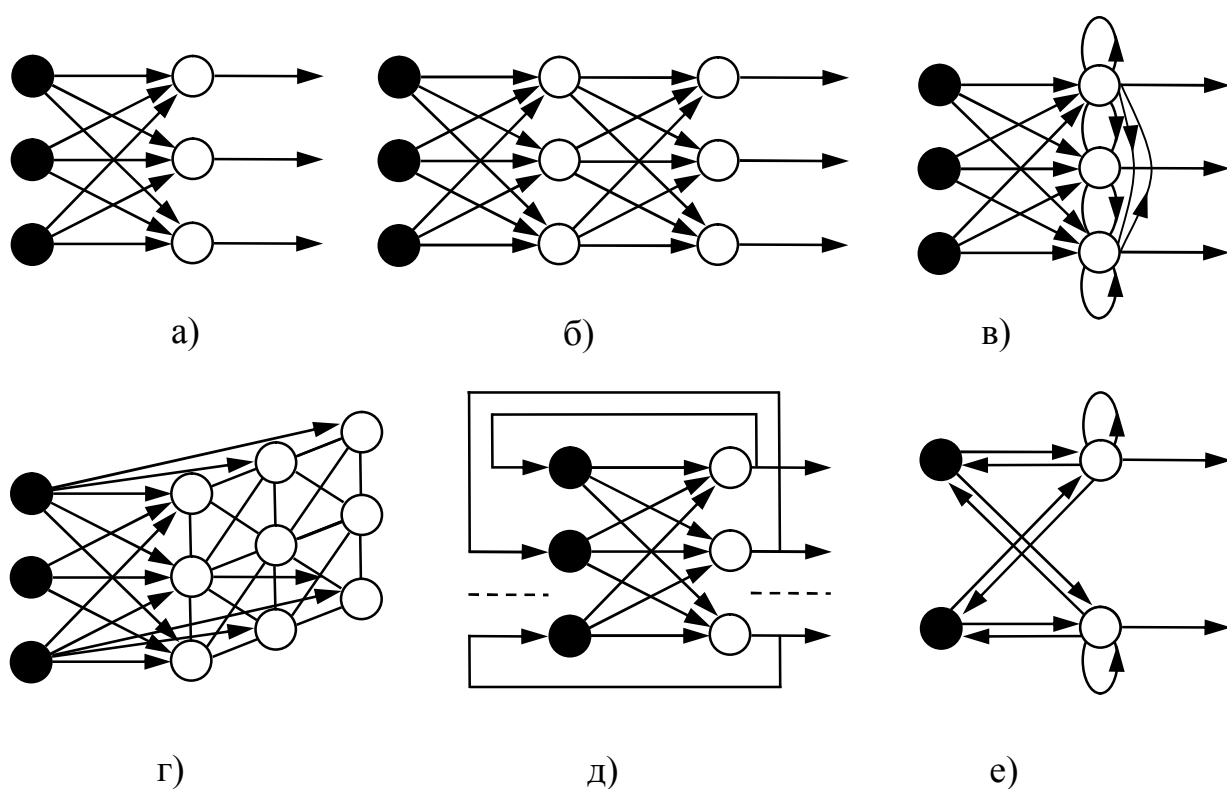


Рисунок А.1 – Структури найбільш поширених ШНМ

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

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

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

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

А.2 Базовий процесорний елемент (штучний нейрон)

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

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

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

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

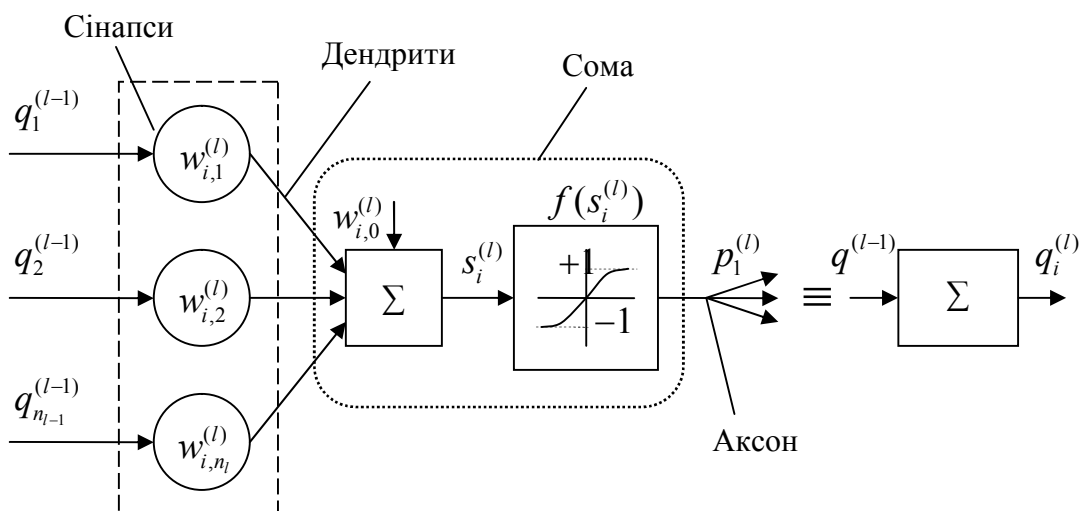


Рисунок А.2 – Схема базового процесорного елементу (штучного нейрона)

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

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

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

- експоненціальні або радіально-базисні (функції Гауса):

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

- гіперболічного тангенса:

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

- сигмоїдні:

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

- бінарні (різного визначення).

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

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

2.3 Статичні лінійні одношарові мережі.

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

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

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

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

однакового для всіх адалин,

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

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

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

видаленням останнього стовпця – вектора коефіцієнтів ініціалізуючих зв'язків

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

Відповідно до (A.2) лінійна багатошарова мережа типу мадаліни здійснює відображення $\mathbf{r}_1 \in R^n$ в $\mathbf{q} \in R$.

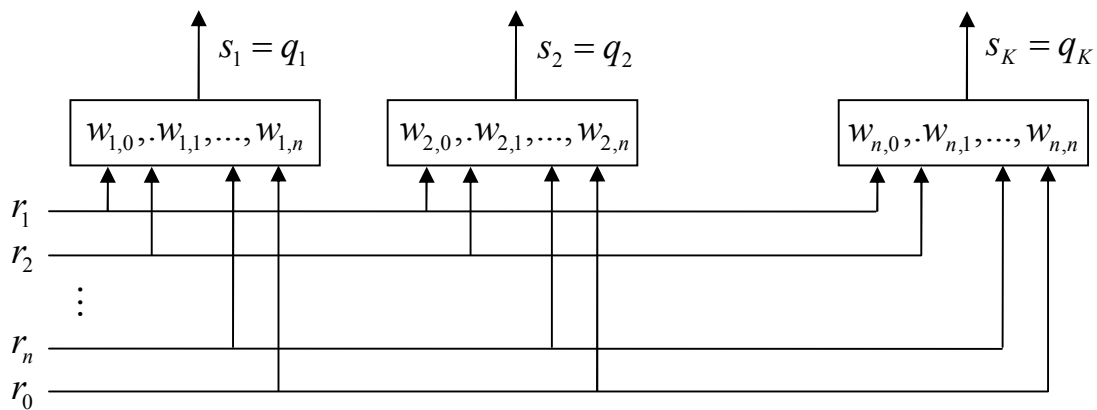


Рисунок А.3—схематичне представлення одношарової лінійної штучної нейронної мережі

А.3 Статичні багатошарові нейронні мережі

Структура статичних багатошарових нейронних мереж (БНМ) зображена на рис. А.4.

У повнозв'язних гомогенних багатошарових нейронних мережах виходи базових елементів кожного шару поступають на входи всіх базових елементів наступного шару, а функція активації $f(s)$ приймається однаковою для всіх БПЕ. Вихідний шар багатошарової мережі виконується у вигляді мадаліни.

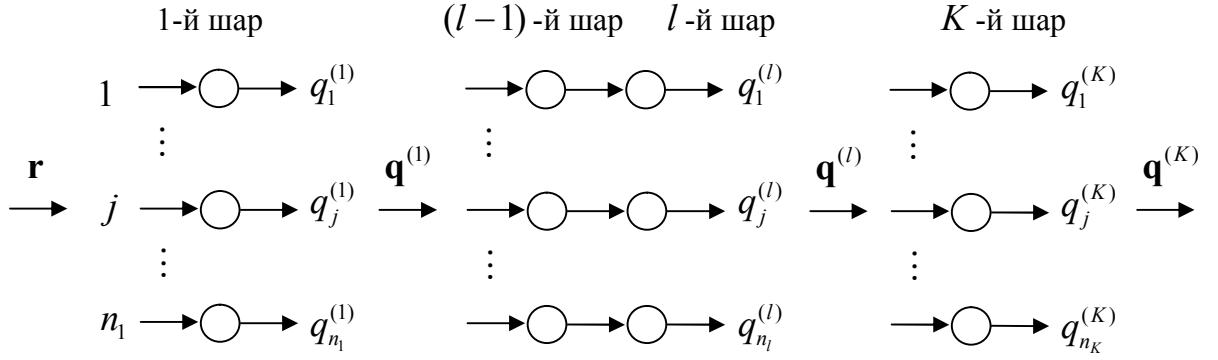


Рисунок А.4 – Структурна схема статичних БНМ

Вхідний шар є буферним для сигналу $\mathbf{r}$. Багатошарова нейронна мережа може бути визначена в символічній формі як $N_{n_0, n_1, \dots, n_K}^K$, де K – число шарів в мережі; n_0 – число входів мережі; $n_i (i = \overline{1, K-1})$ – число базових елементів в i -х «прихованих» або проміжних шарах; n_K – число базових елементів у вихідному K -му шарі і одночасне число виходів $q_1, \dots, q_{n_K}$ багатошарової нейронної мережі. Проміжний l -й шар має n_l базових процесорних елементів. Зв'язки між базовими елементами в шарі відсутні. Виходи базових елементів l -го шару поступають на входи базових елементів тільки наступного $(l+1)$ -го шару. Вихід i -го елемента в l -му шарі багатошарової нейронної мережі може бути визначений так само, як і для будь-якого базового елемента, у вигляді.

$$q_i^{(l)} = f \left(\sum_{j=1}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)} + w_{i,0}^{(l)} q_0^{(l-1)} \right) = f(s_i^{(l)}). \quad (\text{A.3})$$

У векторній формі вихід l -го шару мережі рівний

$$\mathbf{q}_1^{(l)} = \mathbf{f}(\mathbf{W}_1^{(l)} \mathbf{q}_1^{(l-1)} + \mathbf{w}_0^{(l)} q_0^{(l-1)}), \quad (\text{A.4})$$

де $\mathbf{w}_0^{(l)} = \text{col}(w_{1,0}^{(l)}, \dots, w_{n_l,0}^{(l)})$ – вектор вагових коефіцієнтів сигналу ініціалізації

$q_0^{(l-1)}$ в шарі l , який для всіх $l = \overline{1, K}$ приймається рівним +1;

$\mathbf{q}_1^{(l)} = \text{col}(q_1^{(l)}, \dots, q_{n_l}^{(l)})$, $\mathbf{q}_1^{(l-1)} = \text{col}(q_1^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)})$ – вектори виходів базових

процесорних елементів шару l і виходів попереднього $(l-1)$ -го шару, базових елементів, що поступають на входи, шару l . Для вихідного K -го шару маємо:

$$\begin{aligned} \mathbf{q}_l^{(k)} &= \mathbf{W}_1^{(K)} \mathbf{q}_1^{(K-1)} + \mathbf{w}_0^{(K)}, \\ \mathbf{q}_l^{(K-1)} &= \mathbf{f}(\mathbf{W}_1^{(K-1)} \mathbf{q}_2^{(K-1)} + \mathbf{w}_0^{(K-1)}). \end{aligned} \quad (\text{A.5})$$

Співвідношення (A.3) – (A.5) описують статичні багат шарові нейронної мережі з нелінійними функціями активації $f(s)$ пошарово. Функцію s_i^l на i -х виходах суматорів базових елементів шару l називають дискримінантною. У загальному випадку – це відрізок багатовимірного ряду Тейлора, і його найбільший ступінь визначає порядок базових елементів. Так, співвідношення (A.3) відповідає виходу елементу першого порядку, а вектор вагових коефіцієнтів відповідного ряду Тейлора $\mathbf{w}_i^{(l)} = \text{col}(w_{i,0}^{(l)}, \dots, w_{i,n_{l-1}}^{(l)})$ є «пам'ять» базового процесорного елементу. Дискримінантна функція є зважена сума входів БПЕ з коефіцієнтами, рівними значенням «вагів», що настроюються $w_{i,j}^{(l)}$. Дискримінантна функція розділяє вхідний вектор (патерн вхідної активності) на класи – мінімум на два класи у разі лінійної мережі з дискримінантною функцією першого порядку. В цьому випадку функція $s_j = 0$ – дискримінантна лінія, яка розділяє вхідну множину $q_1^{(l-1)}, q_2^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)}$ на дві підмножини:

$$q_j^{(l)} = \begin{cases} 1, & \text{якщо } s_j > 0 \\ 0, & \text{якщо } s_j < 0 \end{cases}.$$

Для елементів другого порядку дискримінантна функція визначається таким чином:

$$s_i^{(l)} = w_{i,0}^{(l)} + \sum_{j=1}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)} + \sum_{j=1}^{n_{l-1}} \sum_{k=1}^j w_{j,k}^{(l)} q_j^{(l-1)} q_k^{(l-1)} = (\mathbf{w}_i^{(l)})^T \mathbf{q}_1^{(l-1)}. \quad (\text{A.6})$$

де $w_{i,0}^{(l)}, w_{i,j}^{(l)}, w_{j,k}^{(l)}$ ($j = \overline{1, n_{l-1}}, k \leq j$) – компоненти вектора вагових коефіцієнтів

i -го базового елементу в l -му шарі мережі, а вектор входу $q^{(l-1)}$ має вигляд:

$$\mathbf{q}^{(l-1)} = \text{col}(1, q_1^{(l-1)}, q_2^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)}, (q_1^{(l-1)})^2, q_1^{(l-1)}, q_2^{(l-1)}, \dots, (q_{n_{l-1}}^{(l-1)})^2). \quad (\text{A.7})$$

З (A.6) і (A.7) витікає, що в кожному l -му шарі нейронної мережі з елементами другого порядку разом з лінійним перетворенням виходів попереднього шару $\mathbf{q}^{(l-1)}$ здійснюється білінійне перетворення цих же сигналів.

Нейронні мережі з дискримінантними функціями високого порядку дозволяють розділяти (класифікувати) вхідну множину на класи за допомогою нелінійних розділяючих поверхонь. Зокрема, розділяюча поверхня $S_i^{(l)} = 0$, де $S_i^{(l)}$ визначається співвідношенням (A.6), є розділяючою поверхнею другого порядку і отримала назву гіперквадрика.

БПЕ з дискримінантною функцією високого порядку має істотно більше число «синаптичних» зв'язків і коефіцієнтів, що настраюються, у складі вектора $\mathbf{w}_i^{(l)}$ при однаковому числі шарів і елементів в структурах багат шарових нейронних мереж з базовими елементами як першого, так і другого порядків. У подальшому будемо розглядати багат шарові нейронні мережі з базовими процесорними елементами першого порядку.

Визначимо зв'язок між входом $\mathbf{r}$ і виходом $\mathbf{u} = \mathbf{q}^{(K)}$ нейронної мережі з елементами першого порядку. Використовуючи (A.4) і $(n_1 \times (n_{l-1} + 1))$ матрицю вагових коефіцієнтів базових елементів l -го шару

$$\mathbf{W}_1^{(l)} = [w_0^{(l)} : w_1^{(l)} : w_2^{(l)} : \dots : w_{n_l}^{(l)}]^T = [(w_0^{(l)}) : (\mathbf{W}_1^{(l)})^T], \quad (\text{A.8})$$

запишемо у векторний-матричній формі вихід l -го шару:

$$\mathbf{q}_1^{(l)} = \mathbf{f}(w_0^{(l)} q_0^{(l-1)} + \mathbf{W}_1^{(l)} \mathbf{q}_1^{(l-1)}) = \mathbf{f}^{(l)}(\mathbf{q}^{(l-1)}), \quad q_0^{(l)} = +1; \quad l = \overline{1, K}. \quad (\text{A.9})$$

У (A.9) $\mathbf{f}^{(l)}$ визначає нелінійне перетворення кожній компоненти вектора $\mathbf{q}^{(l-1)}$ так, як це встановлено співвідношенням (A.3).

Нелінійне рекурентне рівняння, застосоване послідовно для всіх $l = K, K-1, K-2, \dots, 2, 1$ дозволяє встановити зв'язок між входом $\mathbf{r}$ і виходом $\mathbf{u} = \mathbf{q}^{(K)}$ мережі у вигляді:

$$\begin{aligned} \mathbf{u} = \mathbf{q}^{(K)} &= \mathbf{f}^{(K)}(\mathbf{w}_0^{(K)} + \mathbf{W}_1^{(K)} \mathbf{f}^{(K-1)}(\mathbf{w}_0^{(K-1)} + \mathbf{W}_1^{(K-1)} \mathbf{f}^{(K-2)} \times \\ &\times (\dots \mathbf{w}_0^{(l)} + \mathbf{W}_1^{(l)} \mathbf{f}^{(l-1)}(\mathbf{w}_0^{(l-1)} + \mathbf{W}_1^{(l-1)} \mathbf{f}^{(l-2)} \times \\ &\times (\dots \mathbf{w}_0^{(2)} + \mathbf{W}_1^{(2)} \mathbf{f}_1(\mathbf{w}_0^{(1)} + \mathbf{w}_1^{(1)} \mathbf{q}^{(0)}) \dots)) \dots)) = \mathbf{F}(\mathbf{r}). \end{aligned} \quad (\text{A.10})$$

З (A.10) витікає, що багатошарова нейронна мережа виконує через свою структуру складне нелінійне перетворення $\mathbf{F}(\mathbf{r})$ вхідного вектора $\mathbf{r}$ залежно від векторів вагових коефіцієнтів $\mathbf{w}^{(l)}$ нелінійних функцій активації $\mathbf{f}^{(l)}$.

ДОДАТОК Б

ВЛАСТИВОСТІ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ

Б. 1 Здібність до навчання і накопичення інформації

Класичне завдання навчання багатошарової нейронної мережі зводиться до виконання умов, при яких по заданій послідовності $\mathbf{u}^*$ з вказівкою їх приналежності до одного з класів ситуацій, виробляється рішення, що дозволяє класифікувати «незнайомі» ситуації, що знов пред'являються. Зокрема, для статичних багатошарових нейронних мереж таке рішення $\mathbf{u}^*$ формується шляхом настройки вектора вагових коефіцієнтів $\mathbf{w}_i^l$. Якість рішення визначається критерієм $Q(\mathbf{u}^*, \mathbf{u})$, а множинність рішень $\mathbf{u}$ – «конструкцією» алгоритму настройки вагових коефіцієнтів $\mathbf{w}_i^l$. Процес навчання в цьому випадку направлений на досягнення якнайкращого рішення з множини можливих і розуміється при цьому як процес накопичення інформації $\mathbf{u}^*$ і одночасно процес вибору рішення $\mathbf{u}$.

Б.2 Апроксимаційні властивості нейронних мереж

Ці властивості грають вирішальну роль для використання багатошарових нейронних мереж в задачах управління. Доведений ряд теорем що математично обґрунтовують властивості апроксимацій мереж з різними функціями активації. З схем, представлених в попередніх параграфах, витікає, що нейронні мережі обчислюють лінійні або нелінійні функції однієї змінної, а також всілякі суперпозиції функцій від функцій. Що можна отримати, використовуючи тільки такі операції? Які функції вдасться обчислити точно, а які можна скільки завгодно точно апроксимувати за допомогою нейронних мереж? Щоб вивчити їх можливості, потрібно відповісти на ці питання. Виникає і більш загальне питання: чи можна довільну безперервну функцію n змінних представити за допомогою операцій складання, множення і суперпозиції з безперервних функцій однієї

або двох змінних? Таке наближення функцій багатьох змінних за допомогою лінійних операцій і суперпозицій функцій одного змінного і здійснюється нейронними мережами. Кожна мережа складається з штучних нейронів – базових процесорних елементів. Нейрон отримує на вході вектор сигналів $\mathbf{q}$, обчислює його скалярний добуток на вектор вагів $\mathbf{w}$ і деяку функцію одного змінного $f(s)$. Результат поступає на вході інших базових елементів або на вихід. Таким чином, нейронні мережі обчислюють суперпозиції простих функцій одного змінного і їх лінійних комбінацій.

У серії робіт А. Н. Колмогоров і В. И. Арнольд вирішили наступну математичну проблему (складову сутність тринадцятої проблеми Гільберта): будь-яку безперервну функцію n змінних можна точно представити за допомогою операцій складання, множення і суперпозиції з безперервних функцій одного змінного. У теоремі Колмогорова про представлення будь-якої безперервної функції n змінних у вигляді суперпозиції функцій однієї змінної доведено наступне співвідношення:

$$f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \chi_q \left[\sum_{p=1}^n \psi^{pq}(x_p) \right]. \quad (\text{Б.1})$$

де χ_q , ψ^{pq} – безперервні функції однієї змінної, причому χ_q – спеціальні функції для кожної функції $f(x_1, \dots, x_n)$;

ψ^{pq} – незалежно від $f(x_1, \dots, x_n)$ вибрані функції, стандартні для свого класу. Доведено, що функції χ_q можна замінити єдиною функцією χ , а функції ψ^{pq} – на $\alpha^{pq} \psi_p$, де множники $\alpha^{pq} = \text{const}$ і ψ_p – монотонно зростаючі функції.

Пізніше R. Hecht-Nielsen застосував теорему Колмогорова для пояснення апроксимаційних властивостей багатошарових нейронних мереж і показав, що можливе представлення будь-якого безперервного відображення $f: x \in R^n \rightarrow (f_1(x), \dots, f_n(x)) \in R^m$ в мережі з двома прихованими шарами з n БПЕ в першому шарі і $(2n+1)$ БПЕ – в другому. Проте існує, по-перше, необ-

хідність, витікаюча з теореми Колмогорова, в різних функціях активації для кожного БПЕ в першому прихованому шарі i , по-друге, залежність функцій другого шару від форми функції, що апроксимується. Доведені універсальні апроксимаційні властивості багатошарових нейронних мереж з сигмоїдними функціями активації.

У подальшому доведений ряд теорем про апроксимацію безперервних функцій багатьох змінних нейронними мережами з використанням практично довільної безперервної функції одного змінного. Ця функція дійсно може бути довільною, див., наприклад, узагальнену теорему апроксимації Стоуна. У свою чергу теорема Стоуна є узагальненням відомої теореми Вейєрштрасса про рівномірне наближення безперервних функцій поліномами. Тому теорема Стоуна-Вейєрштрасса служить теоретичним обґрунтуванням застосовності штучних нейронних мереж для вирішення різноманітних завдань управління.

Нейронні мережі дозволяють з будь-якою точністю обчислювати довільну безперервну функцію $f_1(s_1, \dots, s_n)$. Отже, з їх допомогою можна скільки завгодно точно апроксимувати функцію, породжену будь-якою безперервною системою. Тому на головне питання: що можуть нейронні мережі? відповідь отримана: нейронні мережі можуть все або майже все – залежно від числа шарів і базових елементів. Залишається відкритим інше питання – як їх цьому навчати? Який механізм апроксимації і пам'яті штучних нейронних мереж? Нижче розглядаються окремі аспекти цієї проблеми.

Багатошарова мережа з одним прихованим шаром і сигмоїдними функціями активації, навчена по градієнтному алгоритму, може апроксимувати функції із заданою точністю у разі відсутності обмеження на число базових елементів в шарі. У ряді робіт дано теоретичне обґрунтування емпіричному спостереженню, що мережі з двома прихованими шарами забезпечують велику точність при обмеженому числі базових елементів в шарах, порівняно з мережами з одним прихованим шаром.

Б.3 Апроксимація і пам'ять в багат шарових нейронних мереж

Як указувалося вище, багат шарова нейронна мережа здійснює нелінійне перетворення $\mathbf{F}$ вхідної множини $\mathbf{r} \in R^N$ у вихідну множину $\mathbf{u} \in U$. Особливістю мережі є її здатність в процесі навчання наближати $\mathbf{u}$ до заданої множини $\mathbf{u} \in U^* \subset U$ з будь-якою точністю, запам'ятовувати умови перетворення у вигляді значень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ ($l = \overline{1, K}$, $i = \overline{1, n_l}$) базових елементів і при повторних пред'явленнях на вході мережі навіть спотвореної множини $\mathbf{r}$ практично відразу ж, без настройки елементів, відновлювати множину $\mathbf{u} \approx \mathbf{u}^*$.

Найбільше застосування, у тому числі і в прикладних завданнях теорії управління, знайшли сигмоїдні, багат шарові нейронні мережі. Поширеність сигмоїдних нейронних мереж не випадкова. Через аналітичні властивості функцій сигмоїдного типу $f(x)$ (нагадаємо, це – непарні, симетричні, такі, що монотонно зростають від 0 до +1 для сигмоїдної функції і від -1 до +1 для функції гіперболічного тангенса, диференціюємі по аргументу x функції) множина всіх базових елементів багат шарової мережі при установці початкових значень коефіцієнтів $\mathbf{w}_i^{(l)}(0) = \mathbf{W}$ розбивається на підмножини слабо активованих БПЕ з $f'(x) \approx 0$ для $|x| > a$ і сильно активованих БПЕ, для яких $0 < f'(x) < \max f'(x)$ при $|x| < a$.

Під час подачі на вхід мережі сигналу $\mathbf{r}$ в процесі навчання підстроюються коефіцієнти $w_{ij}^{(l)}$ лише активованих БПЕ, оскільки в багат шаровій нейронній мережі (це можна показати), $w_{ij}^{(l)} \cong f'(x), x = s_i^{(l)}$. Значення $|x| = a$ визначають аргумент $f(x) = \text{th}(x)$, що розділяє статичну функцію на області, де $f'(x) \neq 0$ ($|x| \leq a$) і $f'(x) \approx 0$ ($|x| > a$) з центром симетрії функції на початку координат $x = 0$. Для сигмоїдної функції $f(x) = (1 + e^{-x})^{-1}$ центр симетрії визначається при $x = 0,5$ і $f'(x) \neq 0$ при $(0,5 - a) \leq x \leq (0,5 + a)$, а при $x < (0,5 - a)$ і $x > (0,5 + a)$ $f'(x) \approx 0$. Необхідно відзначити, що через довільний вибір почат-

кових коефіцієнтів $w_i^{(l)}(0)$ початкове розбиття безлічі базових процесорних елементів на підмножини слабо і сильно активованих елементів має випадковий характер. Особлива ситуація має місце, коли при «невдалій» установці початкових значень $w_i^{(l)}(0)$ надходженні на входи мережі сигналу $\mathbf{r}$ майже всі елементи виявляються слабо активованими. Такий стан називають паралічем мережі. При паралічі процес навчання йде украй поволі або взагалі припиняється. Можливість виникнення паралічу мережі як на стадії установки початкового стану, так і в процесі навчання належить до серйозних недоліків навчаних по алгоритму ВР сигмоїдних багатошарових нейронних мереж. Для боротьби з цим явищем відомі лише евристичні способи.

Для з'ясування питання про те, що відбувається в мережі в процесі апроксимації нелінійним перетворенням вхідної безлічі $\mathbf{r}$ бажаної функції $\mathbf{u}^*$ при виділенні серед всієї множини базових елементів непорожньої підмножини активованих елементів, запишемо для вихідного K -го шару співвідношення:

$$u_i = q_i^{(K-1)} = \mathbf{w}_i^{(K)} \mathbf{q}^{(K-1)} + \mathbf{w}_{i,0}^{(K)} \cdot 1 = \mathbf{w}_i^{(K)} \mathbf{f}^T(\mathbf{s}^{(K-1)}) + \mathbf{w}_{i,0}^{(K)}; \quad (Б.2)$$

$$i = \overline{1, n_K}.$$

Тут $\mathbf{f}(\cdot)$ – векторна функція, компонентами якої є сигмоїдні функції:

$$f(s_i^{(K-1)}) = f\left(\sum_{j=1}^{n_{K-2}} w_{i,j}^{(K-1)} q_j^{(K-2)} + w_{i,0}^{(K-1)}\right) = f((\mathbf{q}^{(K-2)})^T \mathbf{w}_i^{(K-1)}), \quad (Б.3)$$

$$i = \overline{1, n_{K-1}}$$

Очевидно, що для будь-якого проміжного шару $l = \overline{K-1, 1}$ можна записати подібні співвідношення:

$$f(s_i^{(l)}) = f(q^{(l-1)})^T \mathbf{w}_i^{(l)},$$

$$q^{(l-1)} = \text{col}(+1, q_1^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)}); \quad \mathbf{w}_i^{(l)} = \text{col}(w_{i,0}^{(l)}, w_{i,1}^{(l)}, \dots, w_{i,n_{l-1}}^{(l)}); \quad (Б.4)$$

$$i = \overline{1, n_l}.$$

Якщо зіставити (Б.2) з (Б.3) або (Б.4), то без втрати спільності, наприклад для $s_i^{(K-1)}$, можна записати:

$$f(s_i^{(K-1)}) = f(kx), \quad x = (\mathbf{q}^{(K-2)})^T \mathbf{w}_i^{(K-1)}(0), \quad (\text{Б.5})$$

де k – скалярний множник, що в середньому враховує зміну вагових коефіцієнтів i -го елементу в шарі $K-1$ від початкового значення $\mathbf{w}_i^{(K-1)}(0)$ відповідно до алгоритму навчання. Для функції активації типу гіперболічного тангенса ця зміна повинна бути такою, щоб значення скалярної величини $s_i^{(K-1)}$ ($s_i^{(l)}$ – в загальному випадку) для активованих елементів не перевершувало по абсолютній величині заданого числа a з тим щоб $f'(s_i^{(K-1)}) \neq 0$.

Оскільки $df(kx)/dx = k(df(kx)/d(kx))$, через монотонне і обмежене зростання сигмоїдних функцій $f(kx) > f(x)$ для $k > 1$, $x \leq a$ при $f'(x) > 0 \vee x \in (-\infty, +\infty)$ із зміною k (тобто із зміною в процесі навчання мережі вагових коефіцієнтів $\mathbf{w}_i^{(K-1)}$ в даному випадку) змінюється інтервал значень $s_i^{(K-1)}$, для яких $f'(s_i^{(K-1)}) = f'(kx) \approx 0$. Зокрема, із зростанням k ($k > 1$) інтервал значень $s_i^{(K-1)}$, в якому $f'(\cdot) \approx 0$, розширюється і відповідний i -й елемент ($i = \overline{1, n_{K-1}}$) може переміститися в підмножину слабо активованих елементів. Оскільки $\max_x f(kx) > \max_x f(x)$ і ще $k > 1$ то центр симетрії функції $f(kx)$ і відповідно $f'(kx)$ зміщується по осі $s = kx$ на величину $a(1 - k^{-1})$ (щоб це показати, необхідно записати і проаналізувати другу похідну $d^2 f(kx)/dx^2$ з умови рівності її нулю при $kx = a$). Таким чином, залежно від початкових значень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ ($l = \overline{1, K-1}$) і в процесі їх настройки по алгоритму ВР безліч базових елементів мережі розшаровується на підмножини сильно і слабо активованих елементів при нелінійному перетворенні вхідної множини $\mathbf{r} \in R^{(N)}$. Настройка вагових коефіцієнтів слабо активованих елементів трохи змінює вихід мережі $\mathbf{u} = \mathbf{q}^{(K)}$ унаслідок малих значень похідних $f'(s_i^{(l)})$ функцій активації цих елементів. Настройка вагових коефіцієнтів базових елементів другої мно-

жини приводить до віртуальних зсувів центрів симетрії функцій $f(s)$ уздовж осі s і відповідно центрів симетрії парних функцій похідних $f'(s)$. Ці зсуви, як і значення відповідних коефіцієнтів активованих елементів, в результаті навчання утворюють в структурі багат шарової нейронної мережі асоціативну пам'ять про вхідну множину $\mathbf{r}$ в класі R^n в процесі його нелінійного перетворення (Б.1) що апроксимує задану множину $\mathbf{u}^*$. При повторних пред'явленнях на вході мережі нових $\mathbf{r}$ з того ж класу R^n підстроюються лише небагато з сильно активованих елементів, і через це мережу здійснює перетворення $\mathbf{r} \in R^{(N)}$ в $\mathbf{u} \in U$ практично безінерційно. Слід зазначити, що вибір кращого рішення зі всіх можливих в БНМ з сигмоїдними функціями активації здійснюється лише по одному критерію: близькістю до нуля похідної функції активації по аргументу. В цьому випадку в процесі апроксимації змінюється лише форма вибраної функції.

Статичне перетворення (Б.2) загального виду для конкретної багат шарової нейронної мережі з функціями активації $f(s)$, що мають різні, такі, що не піддаються контролю зміщення своїх центрів симетрії мають стохастичний характер і настільки складне аналітичне представлення, що робить практично неможливим аналіз нелінійного перетворення (Б.2) з метою, наприклад, обґрунтованого вибору параметрів багат шарової нейронної мережі. Навчання мережі по алгоритму (Б.2) вносить динаміку в процес перетворення, оскільки алгоритми навчання відповідають нелінійним диференціальним або різницеvim рівнянням, де змінними являються коефіцієнти синаптичних вагів $\mathbf{w}_i^{(l)}$ і виходів $\mathbf{q}^{(l)}$ елементів в l -шарах. Така точка зору дозволяє навчати нейронну мережу еквівалентно представити на множині $\{\mathbf{w}, \mathbf{q}\}$ системою нелінійних диференціальних рівнянь або різницеvim рівнянь і структурами із зворотними зв'язками. Рішення цієї системи рівняннями руху об'єкту відкриває можливість описати динаміку процесів в нейронних мережах, спираючись на методи теорії нелінійних систем. З цією метою розглянемо алгоритми навчання багат шарових нейронних мереж.

ДОДАТОК В

НАВЧАННЯ СТАТИЧНИХ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ

В.1 Загальна характеристика алгоритмів навчання

Під навчанням штучної нейронної мережі розуміється процес настройки вагових коефіцієнтів $w_{i,j}$ її базових процесорних елементів, результатом чого є виконання мережею конкретних завдань – розпізнавання, оптимізації, апроксимації, управління. Досягнення подібної мети формалізується критерієм якості Q , мінімальне значення $\min_w Q = Q^*$ якого відповідає якнайкращому рішення поставленої задачі.

Різноманіття алгоритмів навчання визначається функціональним призначенням мережі, її архітектурою і вибраною стратегією навчання. Розрізняють три основні стратегії навчання: 1) «з вчителем», 2) «без вчителя», тобто з самонавчанням і 3) змішану. У першому випадку нейронна мережа настраюється по заданій навчальній вибірці відповідно до прийнятого правила або алгоритму. У другому випадку наперед не потрібно знати правильний результат навчання і в процесі настройки вагових коефіцієнтів утворюється внутрішня структура активованих базових елементів, відповідна пред'явленому вектору входу мережі. При змішаній стратегії навчання частина вагових коефіцієнтів $w_{i,j}$ настраюється по заданій навчальній вибірці, а інша – відповідно до правил навчання «без вчителя».

Перерахованим стратегіям відповідають конкретні алгоритми навчання штучних нейронних мереж. Кожен алгоритм орієнтований на конкретну архітектуру мережі і застосовується для вирішення певного типу завдань. Так, одно- і багатошарові мережі навчаються по першій стратегії, і це застосовно для вирішення завдань класифікації, апроксимації і управління. Самонавчання використовується в мережах Хопфілда, картах (мережах) Кохонена, що самоорганізо-

вуються, мережах ART (adaptive resonance theory); змішана стратегія навчання застосовується в RBF-мережах з радіально-базисними функціями активації.

Загальним для всіх модифікацій алгоритмів є їх рекурентний характер, а самі алгоритми навчання є лінійними або частіше нелінійні диференціальні (або різницеві) рівняння першого порядку, складені за певним правилом. Ці правила мають евристичний характер, не завжди витікаючи з принципів функціонування біонейронів і нервової мережі живого організму.

Найбільшого поширення набули наступні правила навчання штучних нейронних мереж: правило Хебба; δ -правило; навчання з «конкуренцією» або методом змагання; ART-правило (навчання «без вчителя»); «больцманове» навчання; генетичні алгоритми. У даному розділі стисло опишемо перші два правила. Методу ВР – як базовому методу навчання багат шарових нейронних мереж – буде присвячений окремий розділ.

Правило Хебба – історично перша запропонована концепція навчання нейронних мереж «без вчителя». Відповідно до неї вагові коефіцієнти $w_{i,j}$ синаптичних зв'язків збільшуються, якщо активовано (збуджені) обидва базові елементи – джерело збудження і приймач сигналу збудження. Тим самим регулярно використовувані входи і синаптичні зв'язки активуються за рахунок збільшення «своїх» вагових коефіцієнтів більшою мірою, ніж решта нейронів мережі. Модель навчання Хебба дає пояснення здатності нейронної мережі до звикання і навчання цієї здатності через повторення. Правило або алгоритм Хебба записується таким чином:

$$w_{i,j}(k+1) = w_{i,j}(k) + \gamma q_j q_i, \quad (\text{B.1})$$

де $w_{i,j}(k)$ – значення вагового коефіцієнта, що змінюється від нейрона i до нейрона j до настройки;

$w_{i,j}(k+1)$ – те ж, але після настройки;

q_i, q_j – виходи нейронів i, j відповідно;

$\gamma > 0$ – коефіцієнт «посилення» алгоритму;

$$k = 0, 1, 2, \dots$$

Правило Хебба може бути застосовано для навчання простих мереж типу мадаліни, рекурентних мереж Хопфілда, воно поклало початок процесу «конструювання» безлічі подібних правил навчання нейронних мереж «без вчителя» і, зокрема, на цьому шляху був отриманий вдалий алгоритм навчання «з конкуренцією» для мережі Кохонена (правило Кохонена). Важливою особливістю правила Хебба є те, що зміни синаптичних вагів $w_{i,j}^{(K)}$ залежать тільки від активності базових елементів, зв'язаних між собою даним синапсом (δ -правило реалізує стратегію навчання «з вчителем» або супервізорного навчання. «Вчитель» – це цільовий, бажаний вихід базового елемента типу персептрона u^* , а q – реальний вихід. Для настройки вагових коефіцієнтів базових елементів використовується помилка навчання $\delta = u^* - q$ в алгоритмі градієнтного спуску:

$$w_i(k+1) = w_i(k) - \gamma \delta r_i \quad \gamma > 0. \quad (B.2)$$

де r_i – вхід i -го синаптичного зв'язку базового елемента. Узагальнення δ -правила на багатошарові нейронні мережі приводить до методу зворотного розповсюдження помилки – методу ВР.

Навчання «конкуренцією» на відміну від правила Хебба, по якому може збуджуватися одночасно безліч базових елементів вихідного шару, передбачає «змагання» цих елементів за право активації. Така стратегія отримала назву «переможець бере все». Після активації в процесі настройки змінюються елементи, що лише «перемогли». Відомо, що така ж стратегія навчання властива біологічним нейронним мережам на відміну від навчання по δ -правилу. Навчання «з конкуренцією» використовується в ART-мережах, мережах Кохонена для вирішення задач класифікації зорових, звукових образів, стиснення даних, кластеризації і категоризації усередині класу.

Розглянемо базові алгоритми навчання багатошарових нейронних мереж, що набули переважного поширення при використанні штучних нейронних мереж в задачах управління.

В.2 Метод і алгоритм зворотного розповсюдження помилки

Метод зворотного розповсюдження помилки – метод ВР розроблений для навчання статичних нелінійних багатошарових нейронних мереж і використовує технологію послідовної і пошарової настройки базових процесорних елементів, починаючи з останнього, вихідного шару і закінчуючи настройкою елементів першого шару. «Урок» навчання штучної нейронної мережі може бути повторений необхідне число разів. Для настройки вагових коефіцієнтів базових елементів застосовується δ -правило в його загальній формі: у вигляді алгоритму градієнтного методу мінімізації критерію навчання $Q(\mathbf{e}_u)$. Тут $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)}$ – вектор помилки навчання мережі щодо бажаного (еталонного) виходу $\mathbf{u}^*$ при нелінійному перетворенні ввідного вектора $\mathbf{r}$ у вихідний вектор $\mathbf{u} = \mathbf{q}^{(K)}$ останнього шару K . Помилка $\mathbf{e}_u$, явним чином залежить від коефіцієнтів $w_{i,j}^{(K)}$ і може бути використана як аргумент функціонала Q для настройки базових елементів K -го шару. Явної залежності вектора $\mathbf{e}_u$ і функції $Q(\mathbf{e}_u)$ від вагових коефіцієнтів $w_{i,j}^{(l)}$ ($l = \overline{1, K-1}$) немає. Тому у разі використання критерію навчання $Q(\mathbf{e}_u)$ для настройки коефіцієнтів $w_{i,j}^{(l)}$ помилка $\mathbf{e}_u$ в процесі навчання багатошарової нейронної мережі перераховується послідовно в узагальнені помилки $\boldsymbol{\sigma}^{(l)}$, що явно залежать від значень $w_{i,j}^{(l)}$. Настройка коефіцієнтів $w_{i,j}^{(l)}$ здійснюється за δ -правилом, де використовуються узагальнені помилки $\boldsymbol{\sigma}^{(l)}$. Очевидно, що для шару K помилка $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$. У методі ВР далі використовується локальний квадратичний функціонал $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$. Мінімізація $Q(\boldsymbol{\sigma})$ по вагових коефіцієнтах $w_{i,j}^{(l)}$ ($l = K-1, \dots, 1$), що настроюються, методом градієнтного спуску з перерахунком помилки $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$ для шару K в помилку $\boldsymbol{\sigma}^{(l)} = \mathbf{e}(\mathbf{w}^{(l)})$ для шарів $l = K-1, K-2, \dots, 1$ визначає сутність ВР-технології навчання багатошарових нейронних мереж.

В.2.1. Налаштування коефіцієнтів вихідного шару

Для шару K процедура налаштування вектора всіх вагових коефіцієнтів в K -м шарі $\mathbf{w}^{(k)}$ відповідає зазвичай алгоритму градієнтного спуску:

$$\mathbf{w}^{(k)}(k+1) = \mathbf{w}^{(K)} - \gamma \nabla_{\mathbf{w}} Q(\mathbf{e}_u(\mathbf{w}^{(k)}(k))) , \quad (\text{B.3})$$

де:

$$\nabla_{\mathbf{w}} Q(\cdot) = \text{col} \left(\frac{\partial Q(\cdot)}{\partial \mathbf{w}_1^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_j^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_{n_k}^{(K)}} \right) -$$

вектор розмірністю $(n_k \times 1)$. Вектор $\mathbf{w}^{(k)}$ – складний:

$$\mathbf{w}^{(K)} = \text{col}(\mathbf{w}_1^{(K)}, \dots, \mathbf{w}_i^{(K)}, \dots, \mathbf{w}_{n_K}^{(K)}),$$

де:

$$\mathbf{w}^{(K)} = \text{col}(w_{i,1}^{(K)}, \dots, w_{i,m}^{(K)}, \dots, w_{n_{K-1}}^{(K)})$$

$\mathbf{q}^{(K-1)}(k)$ – вектор розмірністю $((n_{K-1} + 1) \times 1)$ базових елементів в шарі K :

$$\mathbf{q}^{(K-1)}(k) = \text{col}(q_0^{(K-1)}, q_1^{(K-1)}, \dots, q_{n_{K-1}}^{(K-1)}).$$

Сигнал ініціалізації $q_0^{(K-1)}$ зазвичай приймається рівним +1,

Градієнт

$$\nabla_{\mathbf{w}} Q(\mathbf{e}_u) = \mathbf{e}_n^T \frac{\partial \mathbf{e}_u}{\partial \mathbf{w}} = -\mathbf{e}_u^T \frac{\partial \mathbf{q}^{(K)}}{\partial \mathbf{w}^{(K)}},$$

де похідна:

$$\frac{\partial \mathbf{q}^{(K)}}{\partial \mathbf{w}^{(K)}} = \frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{w}^{(K)}} = \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \right) \left(\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} \right)$$

є добуток матриць Якобі. перша з них, така, що розкривається в наступному вигляді:

$$\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{w}_1^{(K)}} \\ \frac{\partial}{\partial \mathbf{w}_2^{(K)}} \\ \dots \\ \dots \\ \frac{\partial}{\partial \mathbf{w}_{n_K}^{(K)}} \end{bmatrix} \times [s_1^{(K)}, s_2^{(K)}, \dots, s_{n_K}^{(K)}] = \begin{bmatrix} \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_1^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_1^{(K)}} \\ \dots & \dots & \dots \\ \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_i^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_i^{(K)}} \\ \dots & \dots & \dots \\ \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_{n_K}^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_{n_K}^{(K)}} \end{bmatrix} = \text{diag} \left(\frac{\partial s_i^{(K)}}{\partial \mathbf{w}_i^{(K)}} \right) -$$

квадратна матриця, кожен елемент якої на діагоналі є вектор-стовпець розмірністю $[(n_{K-1} + 1) \times 1]$, отже повна розмірність матриці рівна $(n_K(n_{K-1} + 1) \times n_K)$. В матриці $\partial \mathbf{s}^{(K)} / \partial \mathbf{w}^{(K)}$ всі елементи, окрім діагональних, рівні нулю, а приватні похідні $\partial s_i^{(K)} / \partial \mathbf{w}_i^{(K)}$ на діагоналі – всі однакові і рівно вектору $\mathbf{q}^{(K-1)}$. Це витікає з визначення дискримінантної функції $s_i^{(K)}$

$$s_i^{(K)} = \sum_{j=0}^{n_{K-1}} w_{i,j}^{(K)} q_j^{(K-1)} = \mathbf{w}_i^{(K)} \mathbf{q}^{(K-1)}.$$

Інша матриця Якобі розкривається у вигляді діагональної матриці $(n_K \times n_K)$

$$\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} = \text{diag}(f'_s(\mathbf{s}^{(K)})) = \begin{bmatrix} f'_s(\mathbf{s}_1^{(K)}) & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & f'_s(\mathbf{s}_{n_K}^{(K)}) \end{bmatrix}.$$

З урахуванням приведених матриць Якобі градієнт функціонала настройки K -го шару багат шарової нейронної мережі можна записати таким чином:

$$\nabla_{\mathbf{w}} Q(\mathbf{e}_u) = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \right) \left(\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} \right) \mathbf{e}_u = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \text{diag}(f'_s(\mathbf{s}^{(K)})) \right) \mathbf{e}_u,$$

а алгоритм настройки вагових коефіцієнтів K -го шару має вигляд

$$\mathbf{w}^{(K)}(k+1) = \mathbf{w}^{(K)}(k) + \gamma \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}(k)} \text{diag}(f'_s(\mathbf{s}^{(K)}(k))) \right) \mathbf{e}_u(k+1) . \quad (\text{B.4})$$

Рекурентний алгоритм (B.4) можна розділити на n_K алгоритмів навчання i -х базових елементів, $i = \overline{1, n_K}$, оскільки діагональна матриця $\partial \mathbf{s}^{(K)} / \partial \mathbf{w}^{(K)}$ складається з однакових елементів $\mathbf{q}^{(K-1)}$.

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i(k+1) \mathbf{q}^{(K-1)}(k+1). \end{aligned} \quad (\text{B.5})$$

Узагальнена помилка σ , навчання мережі для шару K рівна $f'_s(s_i^{(K)})e_{ui}$. У багатошарових мережах останній вихідний шар виконується зазвичай у вигляді мадаліни з лінійними функціями активації і $f'_s(s) = 1$. Тоді $\sigma_i = e_{ui}$ і алгоритм настройки вагових коефіцієнтів шаруючи K приймає вигляд:

$$\mathbf{w}_i^{(K)}(k+1) = \mathbf{w}_i^{(K)}(k) + \gamma e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) \quad (\text{B.6})$$

У алгоритмах (B.4)-(B.6) $k = 1, 2, \dots$ – номер інтеграції. Початкові значення вагових коефіцієнтів $\mathbf{w}^{(K)}(0)$ задаються довільно, але бажано в околиці значень $\mathbf{w}^{*(K)}$, для яких $Q(\mathbf{w}_{(K)}^*) = \min_{\mathbf{w}} Q(\mathbf{w}^{(K)})$.

В.2.2. Налаштування вагових коефіцієнтів прихованих шарів нейронної мережі.

Для прихованих шарів $l = \overline{K-1, 1}$ помилки навчання в явному вигляді не можуть бути визначені. Проте очевидно, що якщо $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)} \neq 0$, то через архітектуру мережі існує ненульова узагальнена помилка і для всіх попередніх шарів. Суть методу ВР якраз і полягає в послідовному визначенні «прихованих» помилок, починаючи з обчислення помилки $\sigma^{(K)}$ у вихідному шарі K , потім помилки $\sigma^{(K-1)}$ в шарі $K-1$ і так до вхідного шару 1. Всі приховані по-

милки $\sigma^{(l)} = \sigma(\mathbf{w}_i^{(l)})$ залежать явно від вагових коефіцієнтів, що настраюються $\mathbf{w}_i^{(l)}$. Розглянемо суть необхідних перетворень в δ -правилі, яке використовує ідею обчислення прихованих помилок в рекурентній процедурі настройки вагових коефіцієнтів в шарах $l = \overline{K-1, 1}$. З цією метою складемо алгоритм настройки вагових коефіцієнтів для окремих i -х базових елементів в шарі l :

$$\mathbf{w}_i^{(l)} = \mathbf{col}(w_{i,0}^{(l)}, \dots, w_{i,j}^{(l)}, \dots, w_{i,n_l}^{(l)})$$

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 $\mathbf{q}^{(l)} = \mathbf{col}(q_0^{(l)}, q_1^{(l)}, \dots, q_{n_l}^{(l)})$

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 $\mathbf{s}^{(l)} = \mathbf{col}(s_0^{(l)}, s_1^{(l)}, \dots, s_{n_l}^{(l)})$, рівна $(n_{l-1} + 1) \times (n_l + 1)$, відповідно розмірність матриці Якобі $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)} = \dim \text{diag}(f'_s(s_0^{(l)})) = (n_l + 1) \times (n_l + 1)$. Якщо покласти $f'_s(s_0^{(l)}) = 1$, то верхній діагональний елемент матриці $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)}$ буде рівний нулю. Отже, матриці

$$\left(\frac{\partial \mathbf{s}^{(l)}}{\partial \mathbf{w}_i^{(l)}} \right) = \left\| \frac{\partial s_i^{(l)}}{\partial w_{i,j}^{(l)}} \right\|, \quad i = \overline{0, n_l}, \quad j = \overline{0, n_{l-1}};$$

$$\left(\frac{\partial \mathbf{q}^{(l)}}{\partial \mathbf{s}^{(l)}} \right) = \text{diag}(f'_s(s^{(l)}))$$

складені. Позначимо першу з них символом $\mathbf{S}^{(l)}$, другу $\mathbf{F}^{(l)}$. Елементами матриці служать приватні похідні $\partial s_i^{(l)} / \partial w_{i,j}^{(l)}$. Оскільки

$$s_i^{(l)} = \sum_{j=0}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)},$$

матриця $\mathbf{S}^{(l)}$ має наступну структуру:

$$\mathbf{S}^{(l)} = \begin{bmatrix} q_0^{(l-1)} & q_0^{(l-1)} & \dots & q_0^{(l-1)} \\ q_1^{(l-1)} & q_1^{(l-1)} & \dots & q_1^{(l-1)} \\ \dots & \dots & \dots & \dots \\ q_{n_{j-1}}^{(l-1)} & q_{n_{j-1}}^{(l-1)} & \dots & q_{n_{j-1}}^{(l-1)} \end{bmatrix} = [\mathbf{q}^{(l-1)} : \mathbf{q}^{(l-1)} : \dots : \mathbf{q}^{(l-1)}],$$

розмірність $\mathbf{S}^{(l)}$ рівна $(n_{l-1} + 1) \times (n_{l-1} + 1)$ і $\mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)})$.

Елементами діагональної $\mathbf{F}^{(l)}$ матриці є приватні похідні функцій активації $f(s)$ по своєму аргументу $s_i (i = \overline{0, n_l})$. Можна показати, що добуток матриць $\mathbf{S}^{(l)}$ і $\mathbf{F}^{(l)}$ зводиться до зовнішнього добутку векторів $\mathbf{q}^{(l-1)}$ і $\mathbf{f}'_s(s^{(l)}) = (f'_s(s_0^{(l)}), \dots, f'_s(s_{n_l}^{(l)}))$.

$$\mathbf{S}^{(l)} \mathbf{F}^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(s^{(l)}))^T.$$

Похідна скалярного функціонала $Q(\boldsymbol{\sigma})$ по векторному аргументу $\mathbf{q}^{(l)} (l = \overline{K-1, 1})$ по своєму сенсу є вектор чутливості функціонала Q до зміни вектора виходу $\mathbf{q}^{(l)}$ l -го шару мережі. Для шару l введемо вектор чутливості функціонала Q :

$$\Lambda^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \text{col} \left(\frac{\partial Q}{\partial q_0^{(l)}}, \frac{\partial Q}{\partial q_1^{(l)}}, \dots, \frac{\partial Q}{\partial q_{n_l}^{(l)}} \right) = \mathbf{col}(\lambda_0^{(l)}, \lambda_1^{(l)}, \dots, \lambda_{n_l}^{(l)}).$$

Приватні функції чутливості $\lambda_i^{(l)}$ обчислюються, по-перше, відповідно до правила диференціювання складних функцій; по-друге, відповідно до структури багат шарової нейронної мережі і мають вигляд:

$$\lambda_i^{(l)} = \frac{\partial Q}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \frac{\partial Q}{\partial q_j^{(l+1)}} \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \lambda_j^{(l+1)} d_{j,i}^{(l+1)} = (\Lambda^{(l+1)})^T \mathbf{d}_i^{(l+1)},$$

$$i = \overline{1, n_l}.$$

Повний вектор функцій чутливості функціонала Q до зміни вектора виходу нейронів l -го шару є

$$\Lambda^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \mathbf{D}^{(l+1)} \Lambda^{(l+1)}, \quad l = \overline{K-1, 1}. \quad (\text{B.7})$$

Матрицю $\mathbf{D}^{(l+1)}$ назвемо матрицею переходу у зворотному напрямі, оскільки для розрахунку функції $\Lambda^{(l)}$ використовуються функції чутливості $\Lambda^{(l+1)}$ і $\mathbf{D}^{(l+1)}$ подальшого шару. Матриця $\mathbf{D}^{(l+1)}$ складена з вектор-рядків $(\mathbf{d}_i^{(l+1)})^T$ де $i = \overline{0, n_l}$. Нагадаємо, що за умовчанням $q_0^{(l)} = +1$, а вагові коефіцієнти ініціалізуючих входів $w_{j,0}^{(l+1)}$ всіх базових елементів в шарі $l+1$ також настроюються. Елементами векторів за визначенням є приватні похідні:

$$\frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \frac{\partial}{\partial q_i^{(l)}} f \left(\sum_{j=0}^{n_l} w_{j,i}^{(l+1)} q_j^{(l)} \right) = \left(\frac{\partial f(s_j^{(l+1)})}{\partial s_j^{(l+1)}} \right) \left(\frac{\partial s_j^{(l+1)}}{\partial q_i^{(l)}} \right) = f'_s(s_j^{(l+1)}) w_{j,i}^{(l+1)}.$$

Для всієї множини індексів $i = \overline{0, n_l}, j = \overline{1, n_{l+1}}$ матриця переходу $\mathbf{D}^{(l+1)}$ має наступну структуру:

$$\mathbf{D}^{(l+1)} = \frac{\partial \mathbf{q}^{(l+1)}}{\partial \mathbf{q}^{(l)}} = \begin{bmatrix} \frac{\partial}{\partial q_0^{(l)}} \\ \frac{\partial}{\partial q_1^{(l)}} \\ \dots \\ \frac{\partial}{\partial q_{n_l}^{(l)}} \end{bmatrix} \times (q_0^{(l+1)}, q_1^{(l+1)}, \dots, q_{n_{l+1}}^{(l+1)}) = \begin{bmatrix} \frac{\partial q_0^{(l+1)}}{\partial q_0^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_0^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_0^{(l)}} \\ \frac{\partial q_0^{(l+1)}}{\partial q_1^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_1^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_1^{(l)}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial q_0^{(l+1)}}{\partial q_{n_l}^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_{n_l}^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_{n_l}^{(l)}} \end{bmatrix}.$$

Відзначимо, що розмірність матриці переходу $\mathbf{D}^{(l+1)}$ рівна $[(n_{l+1} + 1) \times (n_l + 1)]$. Приватні похідні в матриці $\mathbf{D}^{(l+1)}$ визначені раніше. Тепер можна скласти вектор-рядки $(\mathbf{d}_i^{(l+1)})^T$:

$$(\mathbf{d}_i^{(l+1)})^T = (f'_s(s_0^{(l+1)})w_{0,i}^{(l+1)}, f'_s(s_1^{(l+1)})w_{1,i}^{(l+1)}, \dots, f'_s(s_{n_{l+1}}^{(l+1)})w_{n_{l+1},i}^{(l+1)})$$

розмірністю $[1 \times (n_{l+1} + 1)]$. Введемо матриці:

а)

$$\mathbf{W}^{(l+1)} = [\mathbf{w}_0^{(l+1)} : \mathbf{w}_1^{(l+1)} : \dots : \mathbf{w}_{n_{l+1}}^{(l+1)}] = \|\mathbf{w}_{i,j}^{(l)}\|, \quad i = \overline{0, n_{l+1}}; \quad j = \overline{0, n_l},$$

тобто

$$\dim \mathbf{W}^{(l+1)} = (n_l + 1) \times (n_{l+1} + 1),$$

де

$$\mathbf{w}_i^{(l+1)} = \text{col}(w_{i,0}^{(l+1)}, \dots, w_{i,j}^{(l+1)}, \dots, w_{i,n_l}^{(l+1)});$$

б)

$$\mathbf{F}^{(l+1)} = \text{diag}(f'_s(\mathbf{s}^{(l+1)})) = \begin{bmatrix} f'_s(s_0^{(l+1)}) & \dots & 0 & 0 \\ 0 & f'_s(s_1^{(l+1)}) & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & f'_s(s_{n_{j-1}}^{(l+1)}) \end{bmatrix}$$

$$\dim \mathbf{F}^{(l+1)} = (n_{l+1} + 1) \times (n_{l+1} + 1).$$

Відмітимо, що для фіктивного «нульового» базового елементу $q_0^{(l+1)} = +1$ і елемент матриці $\mathbf{F}^{(l+1)} f'_s(s_0^{(l+1)}) = 0$. З урахуванням введених матриць $\mathbf{W}^{(l+1)}$ і

$\mathbf{F}^{(l+1)}$ матрицю $\mathbf{D}^{(l+1)}$ запишемо у вигляді:

$$\mathbf{D}^{(l+1)} = \left\| \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} \right\|_{i=0, \overline{n_l}}^{j=0, \overline{n_{l+1}}} = (\mathbf{W}^{(l+1)})^T \mathbf{F}^{(l+1)} = (\mathbf{W}^{(l+1)})^T \text{diag} (f'_s(s^{(l+1)})). \quad (\text{B.8})$$

Тепер можна записати формулу обчислення градієнта функціонала навчання багатошарової нейронної мережі $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$ по вагових коефіцієнтах будь-якого шару $l = \overline{K-1, 1}$:

$$\nabla \mathbf{w}_i^{(l)} Q = \mathbf{S}^{(l)} \mathbf{F}^{(l)} \Delta^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(\mathbf{s}^{(l)}))^T \Lambda^{(l)} \quad (\text{B.9})$$

Повний алгоритм навчання багатошарової нейронної мережі по методу ВР включає початковий алгоритм настройки вагових коефіцієнтів K -го шару (В.6) і алгоритми настройки базових елементів шарів $l = \overline{K-1, 1}$ використовують формулу (В.9).

В.2.3. 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)

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k+1)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i^{(K)}(k+1) \mathbf{q}^{(K-1)}(k+1); \\ \gamma &> 0; \quad k = 0, 1, 2, \dots; \quad e_{ui} = u_i^* - u_i; \quad i \in \overline{1, n_K}; \\ \mathbf{q}^{(K-1)} &= \text{col}(q_0^{(K-1)}, q_1^{(K-1)}, \dots, q_j^{(K-1)}, \dots, q_{n_{K-1}}^{(K-1)}); \end{aligned} \quad (\text{B.10})$$

б)

$$\begin{aligned}
\mathbf{w}_i^{(l)}(k+1) &= \mathbf{w}_i^{(l)}(k) - \gamma \mathbf{q}^{(l-1)}(k+1)(\mathbf{f}'_s(s_i^{(l)}(k+1)))^T \Lambda^{(l)}(k); \\
\Lambda^{(l)}(k) &= \mathbf{D}^{(l+1)} \Lambda^{(l+1)}(k); \\
\mathbf{D}^{(l+1)}(k) &= (\mathbf{W}^{(l+1)}(k))^T \text{diag}(f'_s(\mathbf{s}^{(l)}(k))); \\
\gamma > 0; \quad l &= \overline{K-1, 1}; \quad \mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, q_1^{(l-1)}, \dots, q_{l_{K-1}}^{(l-1)}); \quad i = \overline{0, n_l},
\end{aligned} \tag{B.11}$$

Зокрема, алгоритм настройки БПЕ шару $K-1$ у відповідності до пункту б) має вид

$$\begin{aligned}
\mathbf{w}_i^{(K-1)}(K+1) &= \mathbf{w}_i^{(K-1)}(k) - \gamma \mathbf{q}^{(K-2)}(k+1)(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k), \\
\Lambda^{(K-1)}(k) &= \mathbf{D}^{(K)}(k) \Lambda^{(K)}(k); \\
\mathbf{D}^{(K)}(k) &= \mathbf{W}^{(K)}(k) \text{diag}(f'_s(\mathbf{s}^{(K)}(k))) = \mathbf{W}^{(K)}(k); \\
\Lambda^{(K)}(k) &= \text{col}\left(\frac{\partial Q}{\partial \mathbf{q}_1^{(K)}}, \dots, \frac{\partial Q}{\partial \mathbf{q}_{n_K}^{(K)}}\right) = -\mathbf{e}_u(k).
\end{aligned}$$

тоді

$$\begin{aligned}
\Lambda^{(K-1)}(k) &= -\mathbf{W}^{(K)}(k) \mathbf{e}_u(k), \\
(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k) &= \\
&= -(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \mathbf{W}^{(K)}(k) \mathbf{e}_u(k) = \sigma_1^{(K-1)}(k+1)
\end{aligned}$$

– скалярна узагальнена помилка навчання нейронної мережі для i -го базового елемента в шарі $K-1$. З її використанням отримуємо алгоритм настройки, формою в точності відповідний δ -правилу:

$$\begin{aligned}
\mathbf{w}_i^{(K-1)}(k+1) &= \mathbf{w}_i^{(K-1)}(k) - \gamma \sigma_i^{(K-1)}(k) \mathbf{q}^{(K-2)}(k+1); \\
\sigma_i^{(K-1)}(k+1) &= -(\mathbf{f}'_s(s_i^{(K-1)}(k)))^T \mathbf{W}^{(K)}(k) \mathbf{e}^{(K)}(k+1); \\
\mathbf{e}_u^{(K)}(k) &= \mathbf{u}^*(k) - \mathbf{q}^{(K)}(k).
\end{aligned} \tag{B.12}$$

Аргумент k позначає момент використання відповідних значень заданих множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$ на k -й ітерації обчислень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ на

«уроках» навчання, число яких залежить від числа шарів мережі і кількості нейронів в шарах.

Фаза навчання завершена, коли досягнутий для заданих в своєму класі множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$. Мережа фіксує цей стан в значеннях своїх вагових коефіцієнтів і відповідних «віртуальних» зсувах центрів симетрії функцій активації і їх похідних активованих базових елементів і зберігає цей стан після зняття сигналу $\mathbf{q}^{(0)} = \mathbf{r}$ на входах мережі а виді віртуальної статичної нелінійної характеристики, зібраної з фрагментів сигмоїдних функцій. Якщо тепер па входах пред'явити ту ж множину або близьку до неї з того ж класу, то нейронна мережа практично миттєво відновить на виході ту ж множину $\mathbf{u} = \mathbf{q}^{(K)}$, що і після фази навчання. Цей етап функціонування мережі називають фазою спогаду. Мережа, таким чином, виконує функцію асоціативної пам'яті. Якщо ж множини $\mathbf{u}^*$, $\mathbf{q}^{(0)} = \mathbf{r}$ є функціями часу: $\mathbf{q}^{(0)} = \mathbf{r}(t)$, $\mathbf{u}^* = \mathbf{u}^*(t)$, то стандартний алгоритм ВР стає неавтономним і вимагає завдання всієї множини значень змінних $\mathbf{r}(t)$ і $\mathbf{u}^*(t)$. Неминуче зростання числа шарів і базових елементів в них і пов'язане з цим збільшення тривалості навчання викликає необхідність забезпечення стійкості процесу настройки мережі по алгоритму (B.10) (B.11).

Очевидно, що в завданнях управління $\mathbf{u}^*(t)$ – функція оптимального управління динамічним об'єктом. Управління за допомогою багатошарової нейронної мережі в двохетапному режимі можливо, якщо тільки функція оптимального управління наперед розрахована і може бути визначена вся множина значень помилки навчання мережі $\mathbf{e}_u(t)$. на інтервалі її визначення. Використання іншої вимірювальної інформації, наприклад помилки відтворення заданої функції $\mathbf{r}(t)$ на виході динамічного об'єкту, вносить запізнювання до алгоритму настройки і приводить до нестійкого процесу. Виникає проблема синтезу динамічних алгоритмів навчання багатошарової нейронної мережі, що враховують вплив інерційності, що вноситься до процесу настройки по алгоритму ВР.

В.3 Перенавчання і узагальнення нейронних мереж

Одна з найбільш серйозних труднощів викладеного підходу полягає в тому, що таким чином ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати, – помилку, яку можна чекати від мережі, коли їй подаватимуться абсолютно нові спостереження. Інакше кажучи, ми хотіли б, щоб нейронна мережа володіла здатністю узагальнювати результат на нові спостереження. Насправді мережа навчається мінімізувати помилку на навчальній множині, і у відсутність ідеальної і нескінченно великої навчальної множини це зовсім не те ж саме, що мінімізувати «справжню» помилку на поверхні помилок в наперед невідомій моделі явища.

Найсильніше ця відмінність виявляється в проблемі перенавчання. При дуже близькій підгонці явище простіше буде продемонструвати не для нейронної мережі, а на прикладі апроксимації за допомогою поліномів, при цьому суть явища абсолютно та ж.

Поліном (або многочлен) – це вираз, що містить тільки константи і цілі ступені незалежної змінної. Графіки поліномів можуть мати різну форму. Чим вище ступінь многочлена (і, тим самим, ніж більше членів в нього входить), тим більше складною може бути ця форма. Якщо у нас є деякі дані, ми можемо поставити мету підігнати до них поліноміальну криву (модель) і отримати, таким чином, пояснення для наявної залежності. Проте наші дані можуть бути зашумлені, тому не можна вважати, що сама краща модель задається кривою, яка в точності проходить через все наявні крапки. Поліном низького порядку може бути недостатньо гнучким засобом для апроксимації даних, тоді як поліном високого порядку може виявитися занадто гнучким і буде точно слід даним, приймаючи при цьому хитромудру форму, що не має ніякого відношення до форми справжньої залежності.

Нейронна мережа стикається з точно такою ж трудностю. Мережі з великим числом вагів моделюють складніші функції і, отже, схильні до перенавчання. Мережа ж з невеликим числом вагів може опинитися недостатньо гнучкою,

щоб змодельовати наявну залежність. Наприклад, мережа без проміжних шарів насправді моделює звичайну лінійну функцію.

Як же вибрати «правильний» ступінь складності для мережі? Складніша мережа майже завжди дає меншу помилку, але це може свідчити не про хорошу якість моделі, а про перенавчання.

Відповідь полягає в тому, щоб використовувати механізм контрольної перехресної перевірки (крос-перевірки), при якому частина навчальних спостережень резервується і не використовується в процесі навчання по алгоритму зворотного розповсюдження. Натомість у міру роботи алгоритму вона використовується для незалежної перевірки результату. На самому початку роботи помилка мережі на навчальній і перевірочній множинах буде однаковою, але у міру того як мережа навчається, помилка навчання, природно, убиває, і, поки навчання зменшує дійсну функцію помилок, помилка на контрольній (перевірочній) множині також убиватиме. Якщо ж контрольна помилка перестала убивати або навіть стала рости, це указує на те, що мережа почала дуже близько апроксимувати дані і навчання слід зупинити.

Відмічене явище занадто точної апроксимації в процесі навчання і називається перенавчанням. Якщо таке трапляється, то рекомендується зменшити число прихованих елементів і/або шарів, бо мережа є дуже могутньою для даного завдання. Якщо ж мережа, навпаки, була узята недостатньо складною для того, щоб моделювати наявну залежність, то перенавчання, швидше за все, не відбудеться, і обидві помилки – навчання і перевірки – не досягнуть достатнього рівня малості.

Описані проблеми з локальними мінімумами і вибором розміру мережі призводять до того, що при практичній роботі з нейронними мережами, як правило, доводиться експериментувати з великим числом різних мереж, деколи навчаючи кожен з них по кілька разів (щоб не бути введеним в оману локальними мінімумами) і порівнюючи отримані результати. Головним показником якості результату тут є контрольна помилка. При цьому, відповідно до загальнонаукового принципу, згідно якому при інших рівних слід віддати перевагу простішій

моделі, з двох мереж з приблизно рівними помилками контролю має сенс вибрати ту, яка менше.

Необхідність багатократних експериментів веде до того, що перевірна множина починає грати ключову роль у виборі моделі, тобто стає частиною процесу навчання. Тим самим ослабляється її роль як незалежного критерію якості моделі – при великому числі експериментів є ризик вибрати «вдалу» мережу, що дає добрий результат на перевірочній (контрольній) множині.

Для того, щоб додати остаточній моделі належну надійність, часто (принаймні, коли об'єм навчальних даних це дозволяє) поступають так: резервують ще одну – тестову множину спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на навчальній і контрольній множинах, реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки один раз: якщо її використовувати повторно для коректування процесу навчання, то вона фактично перетвориться на контрольну множину.

ДОДАТОК Г

УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ З ПРОГНОЗОМ

Г.1 Алгоритм нейроуправління з прогнозом

У роботі використовується ефективна реалізація узагальненого управління з прогнозом (Generalized Predictive Control, GPC) з використанням багатошарової прямонаправленої нейронної мережі, як нелінійній моделі об'єкту управління [44]. Завдяки використанню оптимізаційного алгоритму Ньютона-Рафсона, число ітерацій, необхідних для збіжності, значно менше, ніж при використанні інших методів. Головні витрати алгоритму Ньютона-Рафсона в обчисленні Гесиана, але навіть з цими додатковими розрахунками низьке число ітерацій робить алгоритм Ньютона-Рафсона швидшим, ніж інші методи, при цьому він може використовуватися для управління в режимі реального часу (real-time control).

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

Узагальнене управління з прогнозом, введене Кларком і його співробітниками в 1987, належить до класу методів цифрового управління, званих модель-орієнтоване управління з прогнозом (Model-Based Predictive Control, MBPC) [45-51]. MBPC техніки були успішно проаналізовані і впроваджені в процес управління виробництвом в кінці 1970-х і продовжують використовуватися, оскільки вони системно враховують обмеження реальних об'єктів в режимі реального часу. GPC застосовуються в управлінні немінімально фазовими об'єктами, нестійкими об'єктами і об'єктами із змінним або невідомим часом затримки. Узагальнене управління з прогнозом також є робастним по відношенню до помилок моделювання, перевизначення і недовизначення значень параметрів і шумів датчиків [45]. GPC було спочатку розроблене для моделей лінійних об'єктів, що приводить до рівнянь, які можуть бути вирішені аналіти-

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

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

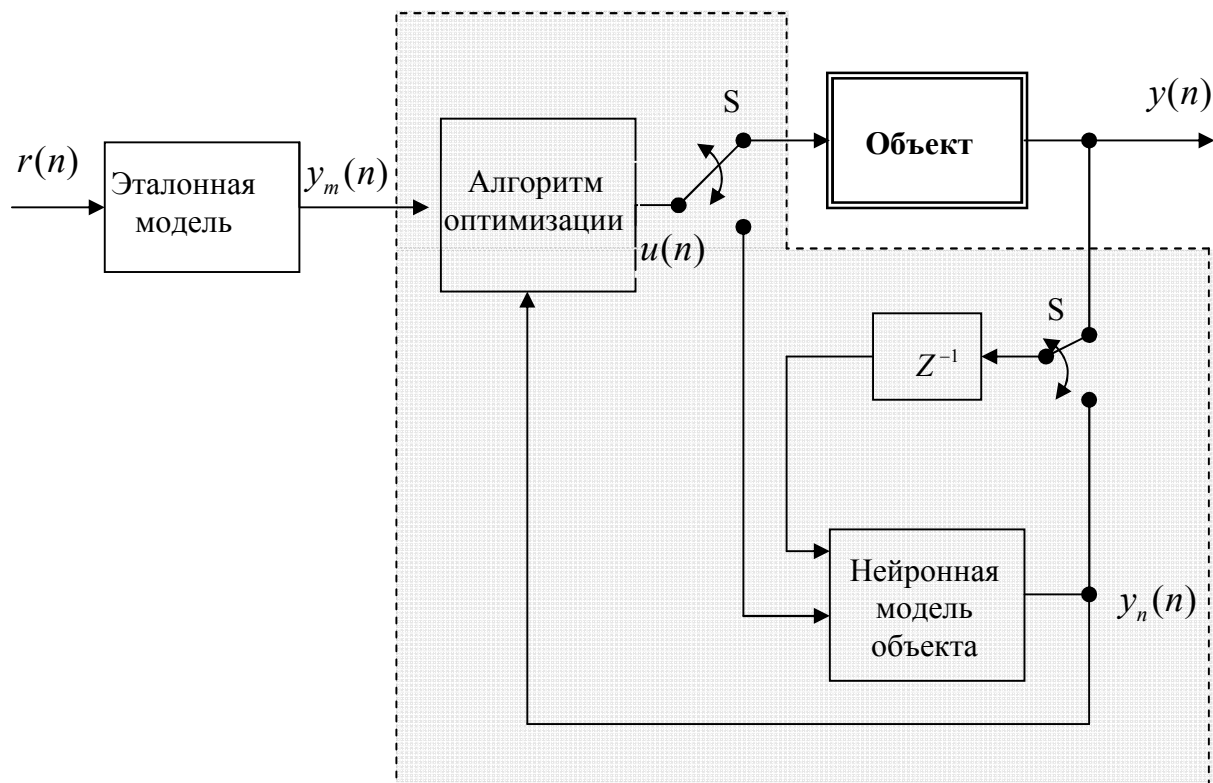


Рисунок Г.1 – Схема системи узагальненого нейроуправління з прогнозом

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

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

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

Узагальнений алгоритм нейроуправління з прогнозом складається з наступних основних кроків.

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

Якість обчислення при реалізації узагальненого управління з прогнозом в основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізацій, які реалізуються в CFM, такі як: Non-gradient, Simplex і Successive Quadratic Programming. Вибір методу мінімізації

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

Якість обчислення при реалізації узагальненого управління з прогнозом в основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізацій, які реалізуються в CFM, такі як: Non-gradient [48], Simplex [49] і Successive Quadratic Programming [50,51]. Вибір методу мінімізації може бути заснований на наступних критеріях: число ітерацій, обчислювальні витрати і точність рішення. У загальному випадку, ці алгоритми вимагають великого числа ітерацій, що робить управління в режимі реального часу скрутним. Дуже невелике число публікацій присвячені реалізаціям управління в режимі реального часу для об'єктів, що мають великі постійні часу [50,51]. Для можливості використання в системах з об'єктами, що мають малі постійні часу, потрібний швидший оптимізаційний алгоритм. Алгоритм Ньютона-Рафсона володіє квадратичною збіжністю, тоді як інші алгоритми мають нижчу швидкість збіжності. Вища швидкість збіжності алгоритму Ньютона-Рафсона вимагає високих обчислювальних витрат, але виправдовується показником швидкості збіжності.

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

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

де N_1 – нижня межа прогнозу;

N_2 – верхня межа прогнозу;

N_u – діапазон управління;

y_m – бажана траєкторія;

y_n – передбачений вихід нейронної мережі;

λ – ваговий множник;

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

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

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

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

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

кожної ітерації розраховується також значення управляючого вектора

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

де $k=1, \dots$ – номер ітерації.

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

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

де якобіан позначений, як

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

а гесіан, як

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

Рішення рівняння (8.2) вимагає явного звернення матриці Гессе. Цей процес може зажадати великих обчислювальних витрат. Одним з підходів, використовуваних, щоб уникнути звернення матриці, служить LU декомпозиція [52] для отримання вхідного вектора $\mathbf{U}(k+1)$. Це досягається шляхом запису рівняння (Г.2) у вигляді системи лінійних рівнянь $\mathbf{Ax} = \mathbf{b}$, що приводить до виразу

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

де

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

У даній формі рівняння (Г.3) може бути вирішене за допомогою двох процедур, описаних в [52]: процедури LU декомпозиції, ludcmp і процедури рішення системи лінійних рівнянь lubksb. Після того, як обчислений вектор $\mathbf{x}$. Процедура повторюється до тих пір, поки зміна кожній компоненти $\mathbf{U}(\mathbf{k} + 1)$ не стає меншим деякого ε . При рішенні системи відносно $\mathbf{x}$ для кожної ітерації алгоритму Ньютона-Рафсона необхідне обчислення кожного елементу якобіана і гессиана. Елемент якобіана з індексом h має наступний вигляд:

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

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

дельта функції, яка визначається як

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

отримаємо

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

Елемент з індексами m, h гессиана визначається таким чином:

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

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

Можна скористатися також визначенням дельта-функції для виразу

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

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

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

Г.2 Структура нейронної мережі для моделювання об'єкту

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

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

ким чином, що безліч входних значень продукує бажані вихідні значення. Помилка формується між відповідями нейронної мережі $y_n(n)$ і об'єкту $y(n)$. Надалі ця помилка використовується для оновлення вагів за допомогою одного з розглянутих вище методів, наприклад градієнтного спуску [53]. Цей процес повторюється до тих пір, поки помилка не досягне допустимого рівня.

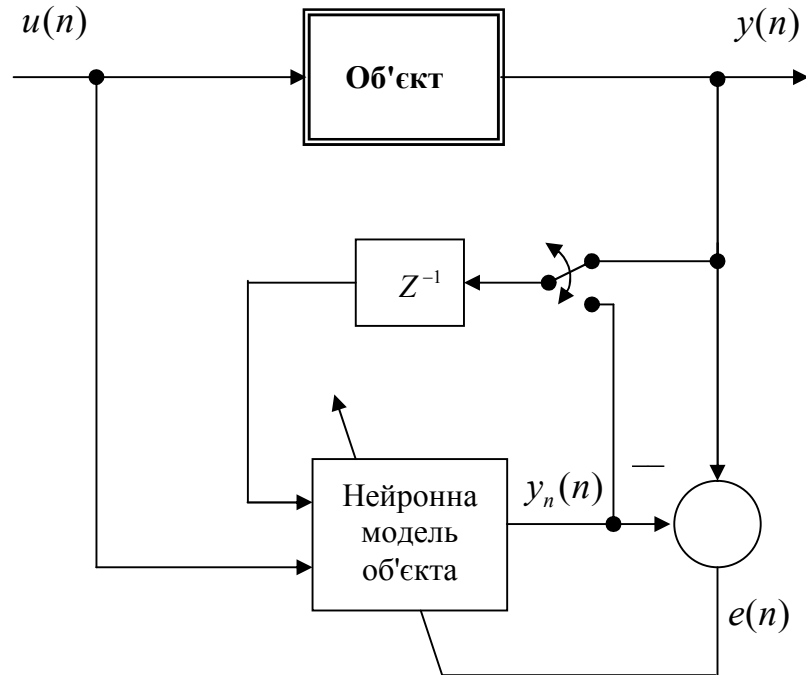


Рисунок Г.2 – Схема тренування мережі в режимі offline

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

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

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

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

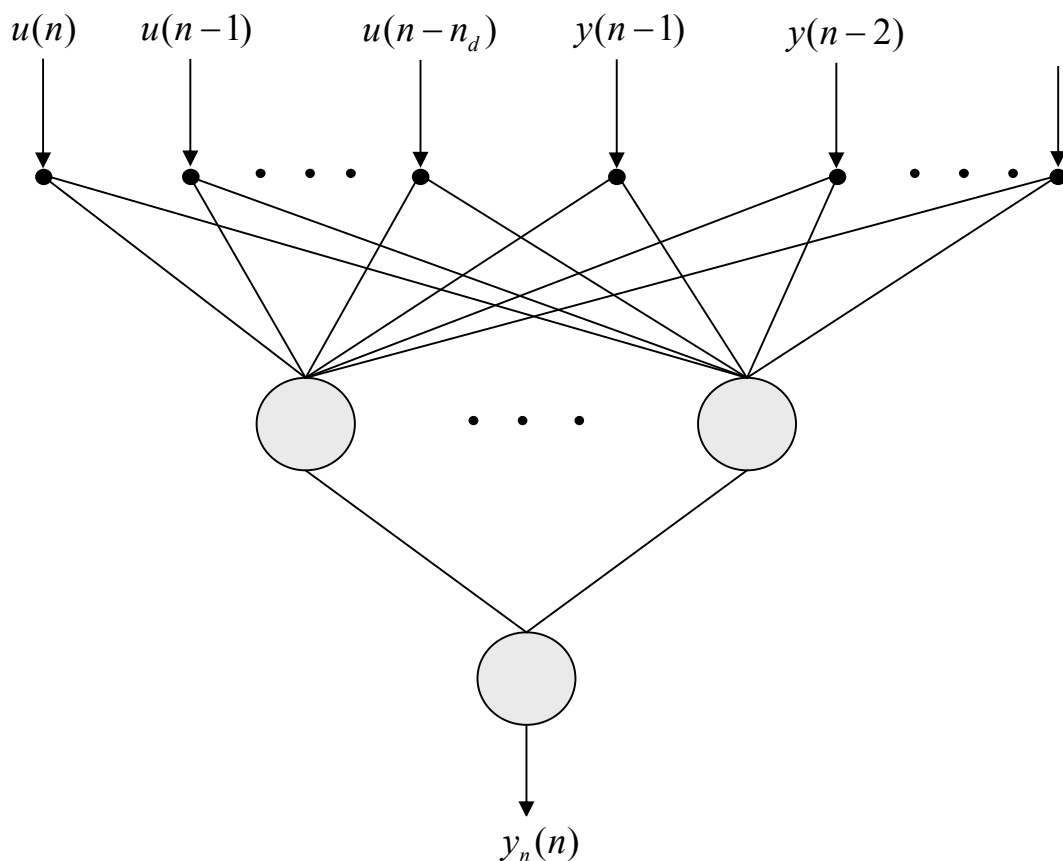


Рисунок Г.3 – Багатошарова нейронна мережа прямого розповсюдження з елементами затримок за часом

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

Рівняння мережі з даною архітектурою виглядають таким чином:

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

і

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

де $y_n(n)$ – вихід нейронної мережі;

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

$net_j(n)$ – аргумент функції активізації j -го нейрона;

S^{L-1} – число нейронів в прихованому шарі;

L – число шарів;

n_d – число вузлів входу, що асоціюються з $u(\cdot)$ без урахування $u(n)$;

d_d – число вузлів входу, що асоціюються з $y(\cdot)$;

w_j – вага, що сполучає j -й прихований нейрон з нейроном виходу;

$w_{j,i}$ – вага, що сполучає i -й вузол входу з j -м прихованим нейроном;

$y(n-i)$ – затриманий вихід об'єкту, використовуваний як вхід для мережі;

$u(n-i)$ – вхід мережі і його затримки;

b_j – зсув j -го прихованого нейрона;

b – зсув нейрона виходу.

Г.3 Прогноз з використанням нейронної мережі

NGPC алгоритм використовує вихід моделі об'єкту для прогнозу динаміки об'єкту для довільного входу від теперішнього моменту часу n до деякого майбутнього моменту $n + k$. Це досягається зрушенням рівнянь (Г.4) і (Г.5) на k маємо

$$y_n(n+k) = \sum_{j=1}^{S^{L-1}} \{w_j f_j(\text{net}_j(n+k))\} + b \quad (\text{Г.6})$$

і

$$\begin{aligned} n_j(n+k) = & \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} u(n+k-i), & k - N_u < i \\ u(n+N_u), & k - N_u \geq i \end{cases} + \\ & + \sum_{i=1}^{\min(k, d_d)} (w_{j, n_d+i+1} y_n(n+k-i)) + \sum_{i=k+1}^{d_d} w_{j, n_d+i+1} y(n+k-i) + b_j. \end{aligned} \quad (\text{Г.7})$$

Особливості рівняння (Г.7) пов'язані з рекурсивним характером прогнозу. Перша сума в (Г.7) визначається відповідно до приведених умов. Умова $k - N_u < i$ управляє попередніми майбутніми значеннями u до $u(n + N_u - 1)$. Умова $k - N_u \geq i$ встановлює входи від $u(n + N_u)$ до $u(n + k)$ рівними $u(n + N_u)$. Друга умова виникає тільки, якщо $N_2 > N_u$. Наступне підсумовування в (Г.7) управляє рекурсивною частиною прогнозу. Підсумовування відповідає зворотному зв'язку по виходу нейронної мережі y_n і виконується k або d_d раз залежно від того, яке значення менше. Остання сума в (Г.7) містить попередні значення y . У наступному підрозділі виводяться похідні від рівнянь (Г.6) і (Г.7) для входу $u(n + h)$.

Г.4 Похідні рівнянь нейронної мережі

Для мінімізації функціонала якості в режимі реального часу за допомогою ітераційного рішення, заснованого на градієнті, необхідно використовувати формули для ефективного обчислення градієнта нейронної мережі.

Для обчислення якобіана і гесиана необхідна перша і друга похідні мережі по вхідному вектору. Елементи якобіана виходять диференціюванням $y_n(n+k)$ з рівняння (Г.6) по $u(n+h)$

$$\frac{\partial y_n(n+k)}{\partial u(n+h)} = \sum_{j=1}^{S^{L-1}} w_j \frac{\partial f_j(net_j(n+k))}{\partial u(n+h)}. \quad (\text{Г.8})$$

Застосовуючи правило диференціювання складної функції для $\frac{\partial f_j(net_j(n+k))}{\partial u(n+h)}$ маємо

$$\frac{\partial f_j(net_j(n+k))}{\partial u(n+h)} = \frac{\partial f_j(net_j(n+k))}{\partial net_j(n+k)} \frac{\partial net_j(n+k)}{\partial u(n+h)} \quad (\text{Г.9})$$

де $\frac{\partial f_j(net_j(n+k))}{\partial net_j(n+k)}$ – похідна функції виходу і

$$\frac{\partial net_j(n+k)}{\partial u(n+h)} = \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} \delta(k-i, h), & k - N_u < i \\ \delta(N_u, h), & k - N_u \geq i \end{cases} + \quad (\text{Г.10})$$

$$+ \sum_{i=1}^{\min(k, d_d)} w_{j, n_d+i+1} \frac{\partial y_n(n+k-i)}{\partial u(n+h)} \delta_1(k-i-1).$$

Відмітимо, що в (Г.10) в останній сумі була введена ступінчаста функція δ_1 . Це було зроблено, щоб підкреслити, що сума рівна нулю для $k-i < 1$, отже доданки, що задовольняють цій умові, не вимагають обчислень. Щоб отримати елементи гесиана, необхідно продиференціювати рівняння (Г.8), (Г.9) і (Г.10) по $u(n+m)$, в результаті маємо

$$\frac{\partial^2 y_n(n+k)}{\partial u(n+h)\partial u(n+m)} = \sum_{j=1}^{S^{L-1}} w_j \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial u(n+h)\partial u(n+m)} \quad (\text{Г.11})$$

$$\begin{aligned} \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial u(n+h)\partial u(n+m)} &= \frac{\partial f_j(\text{net}_j(n+k))}{\partial \text{net}_j(n+k)} \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial u(n+h)\partial u(n+m)} + \\ &+ \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial \text{net}_j(n+k)^2} \frac{\partial \text{net}_j(n+k)}{\partial u(n+h)} \frac{\partial \text{net}_j(n+k)}{\partial u(n+m)}, \end{aligned} \quad (\text{Г.12})$$

і

$$\frac{\partial^2 \text{net}_j(n+k)}{\partial u(n+h)\partial u(n+m)} = \sum_{i=1}^{\min(k, d_d)} w_{j, n_d+i+1} \frac{\partial^2 y_n(n+k-i)}{\partial u(n+h)\partial u(n+m)} \delta_1(k-i-1).$$

Рівняння (Г.12) є результатом застосування правила диференціювання складної функції двічі.

Слід відмітити, що нелінійні оптимізації є процесами, що вимагають великих обчислювальних витрат. Алгоритм Ньютона-Рафсона використовується для створення ефективного обчислювального процесу. При написанні коду, що реалізовує даний алгоритм, всі значення, пораховані в одній процедурі, передаються іншим процедурам для уникнення повторення розрахунку. Найбільш важлива частина коду – обчислення гесиана. Змінні $\frac{\partial \text{net}_j(n+k)}{\partial u(n+h)}$ і $\frac{\partial y_n(n+k)}{\partial u(n+h)}$ обчислювані для якобіана, також використовуються для обчислення гесиана, тому вони передаються в процедуру обчислення гесиана. Оскільки гесіан симетричний, тільки верхньотрикутна частина матриці підлягає визначенню. Використання даних двох зауважень дозволяє значно понизити час завантаження процесора.

Таким чином, узагальнене нейроуправління з прогнозом з використання оптимізаційного алгоритму Ньютона-Рафсона для мінімізації функціонала якості дозволяє отримати високу якість управління у разі складних нелінійностей в об'єкті і дає можливість застосування алгоритму в режимі реального часу.

ДОДАТОК Г

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;
% Система управління електроприводом рольганга
%(3-х роликів секція перед ножицями)
% 2025 рік

% Вихідні дані
kd=0.28;
In=282;
Mn=In/kd;
kn=0.109;
ktp=18;
kc=0.034;
Tmc=5e-3;
Ta=0.083;
Te=0.061;
Re=0.0484;
Jd=3.9875;
Jm=12.7;
Js=Jd+Jm;
w0=5;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);
b12=0;
Uz=24;
Tme=2*Tmc+Ta;
kic=Re/(2*Tmc*kc*ktp);
kpc=kic*Te;
kpe=Js*kd^2*kc/(2*Tme*kn);
kie=kpe/(4*Tme);

% Матриці з розкритим контуром струму (матр.без скор.)
A=[-b12/Jm 1/Jm b12/Jm zeros(1,6);
-c12 0 c12 zeros(1,6);
b12/Jd -1/Jd -b12/Jd CF/Jd zeros(1,5);
0 0 -CF/(Re*Te) -1/Te 1/(Re*Te) 0 0 0 0;
0 0 0 (-kc*kpc*ktp)/Tmc -1/Tmc ktp/Tmc (kpc*ktp)/Tmc
(kpe*kpc*ktp)/Tmc -(kpe*kpc*ktp)/Tmc;
0 0 0 -kic*kc 0 0 kic kpe*kic -kpe*kic;
zeros(1,7) kie -kie;
zeros(1,7) -1/Ta 0;
0 0 kn/(kd*Ta) zeros(1,5) -1/Ta ]
B=[0 -1/Jm;
zeros(6,2);
1/Ta 0;
0 0]
C=zeros(4,9);
C(1,1)=1; C(2,2)=1; C(3,3)=1; C(4,4)=1;
D=zeros(4,2);
```