

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Приладобудівний факультет

Кафедра комп'ютерно-інтегрованих оптичних та навігаційних систем

До захисту допущено:

Завідувач кафедри

_____ Надія БУРАУ

«__» _____ 20__ р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютерно-інтегровані технології та
системи навігації і керування»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»
на тему: «Синтез та аналіз нейромережових регуляторів автоматичних
систем у програмному середовищі MATLAB»

Виконала :

студентка IV курсу, групи ПГ-91

Молла Карина Ескендерівна _____

Керівник:

Доктор технічних наук, професор

Бурау Надія Іванівна _____

Рецензент:

Доцент кафедри ВП, к.т.н., доцент

Шевченко Вадим Володимирович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студентка Молла Карина Ескендерівна _____

Київ – 2023 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Приладобудівний факультет

Кафедра комп'ютерно-інтегрованих оптичних та навігаційних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма: «Комп'ютерно-інтегровані технології та системи навігації і керування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Надія БУРАУ

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студентці

Молла Карині Ескендерівні

1. Тема дипломної роботи «Синтез та аналіз нейромережових регуляторів автоматичних систем у програмному середовищі MATLAB», керівник Бурау Надія Іванівна, д.т.н., професор, затверджені наказом по університету від «___» _____ 20__ р. № _____
2. Термін подання студентом дипломної роботи: 10 червня 2023 року.
3. Вихідні дані до роботи: регулятори, побудовані на основі НМ, для використання в системах автоматичного керування; вхідний сигнал: Випадковий сигнал у межах [20;23].
4. Перелік завдань, які потрібно розробити: Огляд літературних джерел за напрямком систем керування для рухомих об'єктів/мобільних роботів. Огляд та

класифікація нейронних мереж, їх сфер застосування, особливості застосування нейронних мереж для систем керування. Синтез нейромережових регуляторів у програмному середовищі MATLAB. Моделювання та аналіз вихідних сигналів побудованих регуляторі. Висновки по роботі.

5. Перелік графічного (ілюстративного) матеріалу: презентація, яка містить структурні схеми, алгоритми, математичні вирази, графіки

6. Консультанти з розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «20» квітня 2023 року

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1.	Огляд літературних джерел за темою ДР	05.05.2023	
2.	Огляд та класифікація НМ	15.05.2023	
3.	Синтез НМ регуляторів у середовищі MATLAB	25.05.2023	
4.	Моделювання та аналіз вихідних сигналів регуляторів	05.06.2023	
5.	Оформлення рукопису ДР	10.06.2023	
6.	Оформлення презентації, підготовка до захисту	12.06.2023	

Студентка

Керівник дипломної роботи

Карина МОЛЛА

Надія БУРАУ

АНОТАЦІЯ

Дана дипломна робота присвячена огляду перспективності розвитку мобільних роботів, застосуванню їх у людському житті; застосуванню нейронних мереж у робототехніці, зокрема в системах керування; а також синтезу і аналізу нейромережових регуляторів в середовищі Simulink.

У першому розділі було розглянуто різні види мобільних роботів, їхні особливості та роль у житті людини. Було зроблено огляд систем руху роботів, завдяки яким він здатен пересуватися у просторі. Розглядаються різноманітні системи керування, а також використання нейронних мереж у цих системах. Також у розділі наведена загальна інформація про нейронні мережі, її види та класифікацію. Приведено переваги застосування нейронних мереж у робототехніці.

Другий розділ присвячений реалізації, моделюванню та аналізу нейромережових регуляторів у програмному середовищі MATLAB, а саме Simulink. Проведено огляд програмного пакету MATLAB Deep Learning Toolbox, який містить ряд блоків, які можуть бути безпосередньо використані для побудови НМ в середовищі Simulink. Проводиться огляд двох нейромережових регуляторів: регулятор на основі еталонної моделі та регулятор з прогнозуванням, після підбору систем для об'єкта керування складається схема у Simulink та проводиться навчання обраних контролерів. Отримуються результати моделювання та проводиться їх оцінка.

Ключові слова: робототехніка, нейронна мережа, мобільний робот, нейромережовий регулятор на основі еталонної моделі, нейромережовий регулятор з прогнозуванням.

ABSTRACT

This thesis is devoted to an overview of the prospects for the development of mobile robots, their application in human life; the use of neural networks in robotics, in particular in control systems; and the synthesis and analysis of neural network regulators in the Simulink environment.

The first chapter discussed different types of mobile robots, their features, and their role in human life. An overview of the robot's motion systems was made, thanks to which it is able to move in space. Various control systems are discussed, as well as the use of neural networks in these systems. The chapter also provides general information about neural networks, their types and classification. The advantages of using neural networks in robotics are also presented.

The second section is devoted to the implementation, modeling, and analysis of neural network regulators in the MATLAB software environment, namely Simulink. An overview of the MATLAB Deep Learning Toolbox software package is given, which contains a number of blocks that can be directly used to build neural networks in the Simulink environment. Two neural network controllers are reviewed: a controller based on the reference model and a controller with forecasting, after selecting systems for the control object, a diagram is drawn up in Simulink and the selected controllers are trained. The simulation results are obtained and evaluated.

Keywords: robotics, neural network, mobile robot, Model Reference Controller, Neural Network Predictive Controller.

ЗМІСТ

АНОТАЦІЯ	4
ABSTRACT	5
ПЕРЕЛІК СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 ОГЛЯД ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ У СИСТЕМАХ КЕРУВАННЯ МОБІЛЬНИХ РОБОТІВ	9
1.1 Застосування мобільних роботів для завдань керування	9
1.1.1 Загальні поняття про роботів	9
1.1.2 Розвиток мобільних роботів як рухомого об'єкта	13
1.1.3 Системи руху мобільних роботів.....	15
1.1.4 Системи керування мобільних роботів.....	20
1.1.5 Застосування мобільних роботів.....	25
1.2 Характеристика штучних нейронних мереж	28
1.2.1 Загальна інформація про нейронні мережі.....	28
1.2.2 Класифікація та види нейронних мереж.....	31
Висновки до розділу	36
РОЗДІЛ 2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ MATLAB ДЛЯ СИНТЕЗУ НЕЙРОТЕХНІЧНИХ РЕГУЛЯТОРІВ	37
2.1 Загальна характеристика Deep Learning Toolbox.....	37
2.2 Синтез та аналіз регулятора на основі еталонної моделі в Simulink	41
2.2.1 Модель об'єкта керування.....	41
2.2.2 Навчання НМ регулятора	44
2.2.3 Аналіз ефективності НМ регулятора після навчання	52
2.3 Синтез та аналіз регулятора з прогнозуванням в Simulink	54
2.3.1 Модель об'єкту керування	54
2.3.2 Навчання НМ регулятора	56
2.3.3 Аналіз ефективності НМ регулятора після навчання	62
Висновки до розділу	64
ВИСНОВКИ	65
ПЕРЕЛІК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	66

ПЕРЕЛІК СКОРОЧЕНЬ

МР – мобільний робот.

НМ – нейронна мережа.

MRC – Model Reference Controller.

NNPC – Neural Network Predictive Controller.

ОК – об'єкт керування.

ВСТУП

У сучасному світі робототехніка є однією з найбільш актуальних галузей, які швидко розвиваються. Застосування роботів охоплює широкий спектр діяльності, від промислового виробництва до медицини, автоматизації та наукових досліджень. Ефективне керування роботами є одним з найважливіших аспектів для забезпечення їхньої надійної та точної роботи в різних умовах.

Однак, викликами управління роботами виникають при складних умовах, коли інформація про середовище обмежена або неповна, а також при параметрах об'єкта та умовах роботи, які постійно змінюються. В таких випадках, традиційні методи керування можуть бути недостатньо ефективними, оскільки вони не здатні адаптуватися до зміни умов та вимагають чітко визначених моделей та параметрів.

З метою подолання цих викликів, в останні роки все більшу увагу приділяють застосуванню нейромережевих регуляторів у системах керування роботами.

Нейромережеві регулятори використовують потужні алгоритми машинного навчання, що дозволяють моделювати складні залежності, прогнозувати результати та адаптуватися до зміни умов. Вони здатні "вчитися" на основі накопиченої інформації та досвіду, що дає їм гнучкість та адаптивність у різних ситуаціях.

Метою цієї дипломної роботи є синтез нейромережевих регуляторів у середовищі Simulink та дослідження їхнього застосування в системах керування мобільними роботами.

Робота включає аналіз різних аспектів таких регуляторів, включаючи їхню структуру, тренування та оцінку їхньої ефективності в реальних умовах.

Дипломна робота складається з анотації, переліку скорочень, вступу, двох розділів, висновків та списку використаних джерел.

РОЗДІЛ 1 ОГЛЯД ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ У СИСТЕМАХ КЕРУВАННЯ МОБІЛЬНИХ РОБОТІВ

1.1 Застосування мобільних роботів для завдань керування

1.1.1 Загальні поняття про роботів

Взагалі поняття «робот» – це поняття, яке не є чітко визначеним, саме тому його класу можна віднести багато автоматичних пристроїв. Поєднає ці всі пристрої те, що робот може безпосередньо виконувати різноманітні команди оператора, може працювати згідно з заздалегідь складеною програмою або керуватися набором загальних вказівок зі застосуванням штучного інтелекту [1].

Ці завдання дають змогу полегшити, а в деяких випадках і повністю замінити людську працю на виробництві, в будівництві, при роботі з важким вантажем, матеріалами, що мають шкідливий вплив, а також за інших важких або небезпечних для людини умов.

На сьогоднішній день роботизовані пристрої можна поділити на види.

Побутовий робот – це пристрій, призначенням якого є допомога людині у повсякденному житті [1]. У найближчому майбутньому передбачають їх широке використання.

Уже на даний момент широкодоступними є, наприклад, роботи-іграшки, роботи-пилососи, роботизовані газонокосарки, роботи для чистки басейнів тощо [1]. Їх вигляд можна побачити на рис. 1.1.



а)



б)



в)

Рис. 1.1. Приклади побутових роботів [2-4]: а) робот-пилосос; б) робот-іграшка; в) робот для чистки басейну

Якщо казати про промислових роботів, то у більшості випадків – це «руки», тобто маніпулятори, які закріплюють на основу. Вони призначені для виконання одноманітної роботи, такої як свердління, складання, переміщення. Прикладом може бути ливарний робот, зварювальний робот чи фарбувальний робот [1], які зображено нижче на рис. 1.2.



а)



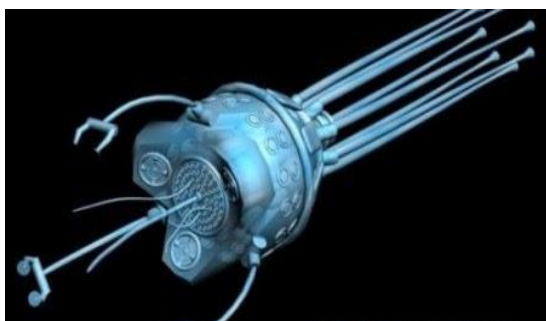
б)

Рис. 1.2. Приклади промислових роботів[5-6]: а) зварювальний робот; б) фарбувальний робот

Наступний вид – це нанороботи. Назва каже сама за себе, це роботи, розмір яких можна порівняти з молекулою (менше 10 нм), вони мають функції руху, обробки та передачі інформації, виконання програм. Є нанороботи, які здатні

створювати власну копію, тобто можуть самовідтворюватися, їх називають реплікаторами. На жаль, на сьогоднішній день даний напрямок технологій знаходиться в основному на науково-технічній стадії розвитку, але має велику перспективу застосування у медицині, генній інженерії та інших галузях. На даний момент уже були представлені деякі примітивні молекулярні машини (приблизний вигляд можна побачити нижче на рис. 1.3(а)) і більш того застосовані на практиці [1]. Було винайдено наноробота, якого можна запрограмувати на пошуки спеціальних клітин. Він кріпиться до мембрани і, знайшовши ціль, пробивається крізь неї, утворюючи пори, що спричиняють гибель клітини. Також вчені мають змогу контролювати степінь пробиття, що дозволяє «атакувати» потрібні їм клітини. Подібні розробки застосовують у експериментах з боротьбою проти раку. Уже є відомі випадки вдалих експериментів без шкідливого впливу.

Існує ще одне визначення, воно описує наноробота, як машину, яка має змогу взаємодіяти або маніпулювати об'єктами, розмір яких вимірюється у наномасштабі. Спираючись на нього, нанороботом можна назвати навіть великі апарати, такий як атомно-силовий мікроскоп, так як він може виконувати маніпуляції над об'єктами на нанорівні, вигляд якого можна побачити на рис. 1.3(б).



а)



б)

Рис. 1.3. Приклади нанороботів[7-8]: а) наноробот; б) атомно-силовий мікроскоп

Людиноподібний робот або робот-гуманоїд – це машина, ходова частина якої виготовлена у вигляді людиноподібного тіла [1]. Такий дизайн робота зумовлений конкретною метою:

- функціональна мета: заради використання людських інструментів або середовищ життя людини;
- експериментальна мета: для вивчення прямостоячої ходьби;
- медична мета: вивчення впливу на організм різноманітних навантажень тощо.

Деякі людиноподібні роботи можуть моделювати лише певну частину людського тіла (як робот, який зображено на рис. 1.4(а)). У більшості ж випадків такі роботи мають голову, тулуб, пару рук та ніг (як робот, який зображено на рис. 1.4(б)). Робот-гуманоїд може мати голову, таким чином він має змогу відтворити людські емоції через міміку [1].



а)



б)

Рис. 1.4. Приклади людиноподібних роботів[9-10]: а) який моделює частину людського тіла; б) який моделює ціле людське тіло

Також є ще один вид роботів – це мобільні роботи, але його ми розглянемо окремо в наступному пункті. Не залежно від призначення та функцій роботів можна сформулювати три закони робототехніки:

- 1) робот не може завдати людині шкоди або допустити її виникнення шляхом своєї бездіяльності;
- 2) робот має виконувати накази людини, якщо вони не суперечать твердженню з першого правила;

- 3) робот повинен піклуватися про власну безпеку у всіх випадках, коли вони не суперечать твердженням з першого та другого правила [1].

Наразі при створення роботів можна виділити дві дослідницько-конструкторські задачі: розробка конструкції окремого робота (агента) та розробка алгоритмів та програмного забезпечення гуртової взаємодії роботів-агентів, беручи за основу теорію колективної поведінки. Більший інтерес дослідницьких груп викликає друга задача, саме на ній концентрується увага при розробці роботів [1].

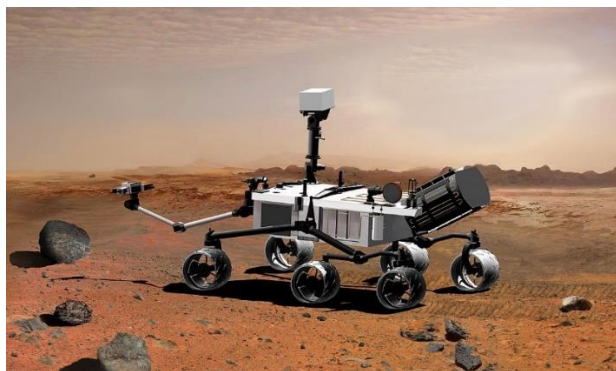
1.1.2 Розвиток мобільних роботів як рухомого об'єкта

Мобільні роботи (МР) відрізняються від інших роботів здатністю переміщуватися в просторі автономно. Якщо він сам має можливість визначити дії, які необхідно здійснити для виконання поставленого завдання, то робота можна назвати автономним. Також він потребує когнітивного блоку або системи керування для координації всіх підсистем, які є складовими робота [11]. Деякі приклади таких роботів наведено нижче на рис. 1.5.

Основи мобільної робототехніки охоплюють такі області, як рух, сприйняття, пізнання та навігація.

Автономні роботи можуть бути здатні до різних видів руху. Рух може бути керованим вручну оператором, автономним або комбінованим, коли оператор та робот працюють разом.

Задля якісного сприйняття МР може бути оснащеним різними сенсорами, такими як камери, лазерні сканери, мікрофони, датчики тиску і багато інших. Ці сенсори допомагають роботу отримувати інформацію про все, що оточує його, наприклад розпізнавати об'єкти, вимірювати відстані, виявляти звуки тощо. Сприйняття дає можливість роботу взаємодіяти з оточенням та збирати необхідні дані для подальшої обробки і прийняття рішень [11].



а)



б)

Рис. 1.5. Приклади мобільних роботів[12-13]: а) марсохід; б) робот для гасіння пожеж

Пізнання також відіграє важливу роль в мобільній робототехніці. МР можуть бути запрограмовані для вивчення нових вмінь та набуття знань про своє оточення. Вони можуть використовувати алгоритми машинного навчання для аналізу даних та виявлення закономірностей, які допомагають роботам адаптуватися до змін умов та вирішувати складні задачі [11].

Навігація також є важливою складовою мобільної робототехніки. Роботи повинні мати здатність рухатися у просторі, визначати своє положення, а в подальшому і планувати оптимальний шлях до бажаної кінцевої точки. Для цього вони можуть використовувати різноманітні технології, такі як GPS, інерційні сенсори, компаси та інші. Навігаційні алгоритми дозволяють роботам визначати своє місцезнаходження, створювати карту простору, що навколо нього, а потім і допомагають створювати маршрут з одного пункту в інший [11].

Один з викликів при створенні МР полягає у забезпеченні безпеки під час руху. Для цього використовуються різні підходи, такі як контроль швидкості, система уникнення зіткнень та попередження про небезпеку. Усе це допомагає значно зменшити ризик виникнення різних аварійних ситуацій, а також забезпечити безпеку для самого робота, навколишнього середовища та людей [11].

Також важливо враховувати такий фактор, як енергоефективність. Використання енергоефективних компонентів та алгоритмів допомагає збільшити тривалість роботи та зменшити частоту використання зарядки і заміни батареї.

Варто зауважити, що використання альтернативних джерел енергії (таких як використання енергії сонця, вітру, води тощо) може бути дуже корисним у нашій справі, адже є порятунком навколишнього середовища, але і допомагає забезпечити стабільне живлення мобільного робота навіть у тих місцях, де можуть бути труднощі з енергоресурсами [11].

Ще одним з напрямків розвитку є інтеграції МР у великі системи та мережі. Це можливо за рахунок взаємодії роботів один з одним. Їм потрібно обмінюватися даними, координувати свої дії. Такі системи дають змогу створити розподіленні команди роботів, у яких вони можуть співпрацювати для виконання складного завдання [11].

Не варто забувати про взаємодію з людиною. Розробляється велика кількість систем спілкування, завдяки яким роботи здатні сприймати інструкції, розуміти мову чи навіть визначати емоції. Завдяки цьому з'являються нові можливості для співпраці з людьми у тих сферах, де раніше це було неможливим, або зробити цю взаємодію більш природньою та комфортнішою [11].

1.1.3 Системи руху мобільних роботів

Система пересування робота є важливим аспектом конструкції МР. Вона залежить не лише від середовища, у якому робот буде виконувати свою роботу (над чи під землею, над чи під водою, у повітрі або навіть у космосі та на інших планетах), але й від технічних критеріїв, а саме: маневреність, керованість, рельєф місцевості, ефективність, стійкість тощо.

Залежно від цих критеріїв МР здатні ходити, стрибати, котитися, ковзати, бігати, плавати або літати [11].

За системами пересування МР можна розділити на такі категорії:

1. Наземний:

- а) колісний (одно-, дво-, три-, чотири- та багатоколісний);

- б) танковий (гусеничні);
- в) крокуючі (дві, чотири, шість ніг та багатоноги);
- г) гібрид.

2. Повітряний.

3. Водний.

4. Інший.

Стаціонарний. Прикладами цього типу є маніпулятори та промислові роботи. Сама основа робота є нерухомою, а складаються вони з відкритого кінематичного ланцюга, в основному з інструментом на кінці, тому вони здатні виконувати різні завдання, такі як зварювання, фарбування, механічна обробка, переміщення тощо [11].

Також важливими стаціонарними роботизованими системами є захватні пристрої, так як захоплення є важливою частиною в транспортуванні.

Наземні роботи. Їх можемо класифікувати наступним чином [11].

Колісні мобільні роботи. Колеса є однією з найважливіших систем для пересування роботів. Використання коліс є простішим рішенням, їх легше відтворити. А також зазвичай цей варіант бюджетніший. Керування колесами є не таким складним, вони спричиняють не таке сильне зношення поверхні, якою вони пересуваються. Також перевага у тому, що немає великих труднощів з утворенням балансу при русі [11].

Основним недоліком є те, що вони не так добре можуть пересуватися через перешкоди, такі як кам'яниста місцевість, ділянки з низьким тертям або гострим поверхням тощо [11].

Існує чотири основних типи коліс (зображені на рис. 1.6.):

- 1) Стаціонарне стандартне колесо. Це звичайне колесо, яке використовується для створення стаціонарної опори. Хоча воно не призначене для переміщення, але може забезпечувати стабільність та підтримку для рухомих механізмів, наприклад, для підставок або маніпуляторів [11].
- 2) Роликове колесо. Воно складається з одного чи декількох роликів, які знаходяться на осі. Таке колесо призначене для кочення або ковзання по

поверхні, чим забезпечує легке переміщення. Цей тип коліс часто використовується у роликових конвеєрах чи ковзанках [11].

- 3) Шведське колесо. Колесо (також відоме як «шведський стіл») представляє собою механізм, який складається з декількох паралельних коліс, які пов'язані між собою. Так як воно здатне обертатися навколо своєї осі, то таке колесо дозволяє забезпечити поворотність та зручність руху для певних додаткових пристроїв (наприклад, важіль або навантажувач). Широко використовується таке колесо у промислових системах [11].
- 4) Кулькове (сферичне) колесо. Цей тип колеса має кульковий підшипник і може обертатися навколо власної осі у будь-якому напрямку. Воно забезпечує високу маневреність та здатність до повороту на місці. Кулькові колеса використовуються у випадках, коли потрібно забезпечити високу маневреність та зручність переміщення, вони допомагають забезпечити якісне керування об'єктом навіть в обмеженому просторі [11].

Знання про кількість та тип коліс у робота є важливим для моделювання кінематики та динаміки.

Загалом дослідження колісних роботів концентруються на проблемах тяги, стабільності, маневреності та контролю. Стійкість у більшості випадків не є надто великою проблемою для колісних роботів, адже вони майже завжди сконструйовані таким чином, щоб усі колеса мали постійний контакт з землею. Трьох коліс цілком достатньо для стабільного балансу робота. Для застосування більшої кількості коліс потрібно використовувати системи підвіски. Їх завдання полягає в тому, щоб забезпечувати всім колесам контакт з поверхнею, по якій рухається МР [11].

Одноколісні роботи мають лише одне колесо, яке може бути фіксованим. Одноколісна система є нестабільною, тому щоб підтримувати положення рівноваги, необхідно одночасно контролювати поздовжню та поперечну стійкість [11].



а)



б)



в)



г)

Рис. 1.6. Приклади типів коліс[14-17]: а) стаціонарне колесо; б) роликове колесо; в) шведське колесо; г) кулькове колесо

Двоколісні роботи мають два однакові колеса, які встановлюються зазвичай паралельно і керуються двома незалежними приводами. Вважається, що кожне колесо розташовано перпендикулярно відносно поверхні, а контакт між колесами та поверхнею є нековзким, а коченням [11].

У триколісних роботах можна виділити два типи: з диференційованим керуванням, коли є два ведучих колеса та одне, яке може вільно обертатися, чим підтримує рівновагу робота; і коли є два колеса, які приводить у рух один привод, та одне ведене колесо [11].

Чотириколісний робот є більш стабільним за попередні типи, так як центр ваги розташований усередині «прямокутника», який утворюють колеса. Колеса можуть мати диференційоване керування, керуватися приводом два на два або керуватися за допомогою керма [11].

У багатоколісних роботів використовуються потужніші колеса, через що їх конструкції стають більш складними, але і стабільність вони гарантують кращу [11].

Танкові роботи, замість коліс, використовують протектори або гусениці. Гусеничні роботи більшу ділянку, яка має контакт з поверхнею, по якій вони рухаються. Цей аспект впливає на покращення маневреності роботів на пухкій поверхні, у порівнянні з колісними. Але в той самий час велика площа дотику з землею впливає на процес зміни напрямку робота, адже щоб виконати цю дію йому потрібно здійснити заносний поворот. Мінус таких роботів у неточності розрахунків через ці заноси [11].

Крокуючий МР використовує ноги для пересування. Ця форма пересування є більш дорожчою за попередньо розглянуті. З переваг можна назвати їх здібність пересуватися м'яким і нерівним рельєфом, у них краща мобільність, менший вплив на землю, краща стабільність, краща енергоефективність, а також краща маневреність та адаптивність [11].

Проблемою в таких роботах є забезпечення постійного балансу. Наприклад, одноногі роботи повинні постійно стрибати, щоб підтримувати рівновагу. Загалом більшу популярність мають роботи з парною кількістю ніг.

Гібридні мобільні роботи – це роботи, які комбінують у своїй структурі вищеописані типи, наприклад, ноги та колеса чи гусениці та колеса тощо [11].

Повітряні мобільні роботи використовують повітряну тягу для піднесення, руху та балансування у повітрі. Одним з найпоширеніших типів таких роботів є безпілотний літальний апарат (також відомий як «дрон»). Дрони зазвичай оснащені геліодвигунами або електродвигунами [11].

Водні роботи користуються великою популярністю, так як людство вже довгий час цікавиться секретами підводного світу. Плаваючі роботи здатні утримуватися на поверхні води, а тому можуть стати у нагоді для різних досліджень, моніторингу та рятуванні людей. Є підводні роботи, вони здатні досліджувати дно водойми там, де не може людина [11].

Системи руху МР є важливою складовою їхньої ефективності та функціональності, адже завдяки ним робот здатен переміщатися. З розвитком технологій систем руху мобільні роботи мають змогу ставати все швидшими та маневренішими.

1.1.4 Системи керування мобільних роботів

Механічна структура робота повинна бути керованою, щоб виконувати завдання і вирішувати поставлені задачі. Система керування охоплює три різні складові, а саме: сприйняття, обробка даних та дію [11].

Система сприйняття надає інформацію про навколишнє середовище, самого робота і зв'язок між ними. Далі ця інформація обробляється, після чого відповідні команди надсилаються до виконавчих механізмів, які і змушують рухатися конструкцію. Після того, як середовище, напрямок руху, місце призначення або мета відомі, архітектура робота повинна згенерувати план шляху, який робот повинен пройти, щоб дістатися кінцевого пункту призначення. На основі задач робота та інформації, отриманої з датчиків, системи керування та пізнання повинні прийняти рішення про подальші дії [11].

Також система керування відповідає за координацію всіх вхідних даних, а також планування руху робота. МР може знадобитися «когнітивна» модель, призначення якої відобразити робота, середовище, в якому він знаходиться, а також спосіб їхньої взаємодії [11].

Роботи є прикладом нелінійної динаміки. Методи нелінійного керування використовують знання та (або) параметри системи з метою відтворення її поведінки. Нелінійне керування, оцінка та спостереження є причиною утворення складних алгоритмів. Одним з завдань системи керування є відстеження положення мобільного робота. Звісно, виникає і похибка цього відстеження. Спосіб, за допомогою якого відбувається обчислення та відстеження цієї похибки (частота зворотного зв'язку, коефіцієнти підсилення для пропорційних і похідних похибок тощо), породжує велику кількість методів керування, які можна застосувати до мобільних роботів [11].

Залежно від проаналізованих властивостей і способу вирішення рівнянь нелінійної динаміки можна класифікувати підходи керування наступним чином [11]:

Глобальне керування на основі лінеаризації – це підхід до керування системами, який ґрунтується на лінеаризації динаміки системи навколо рівноважного стану чи точки референсу. Лінеаризація полягає в апроксимації нелінійної системи за допомогою лінійної моделі, що спрощує процес аналізу та керування. Для цього можуть використовувати різні методи, такі як розкладання у ряд Тейлора чи лінеаризація за допомогою диференціальних рівнянь [11].

Керування на основі наближеної лінеаризації – це підхід до керування системами, який використовує наближену лінеаризацію динаміки системи навколо поточного стану або точки референсу. Для лінеаризації можна використати перший чи другий порядок розкладання в ряд Тейлора навколо поточного стану системи [11].

Розглянемо декілька прикладів популярних стратегій контролю.

Обчислені методи керування крутним моментом – це підходи до керування динамічними системами, зокрема механічними системами, з використанням розрахунку крутного моменту (чи керуючого сигналу) на основі обчислень та алгоритмів [11]. Є декілька основних обчислених методів керування з крутним моментом:

- 1) *модельно-прогнозуюче керування*, де використовується математична модель системи та прогнозування її поведінки в подальшому для прийняття оптимальних рішень у кожен момент часу. Даний метод дозволяє враховувати обмеження системи та оптимізувати керуючий сигнал з урахуванням майбутньої поведінки системи [11];
- 2) *адаптивне керування крутним моментом*, де керуючий сигнал адаптується на основі вимірювань та оцінки стану системи. Адаптивне керування дає змогу системі відновлюватися від змін у динаміці або невідомих параметрів [11];
- 3) *метод оптимального керування*, де застосовують методи оптимізації для знаходження керуючого сигналу, який мінімізує певну критеріальну функцію якості, таку як квадратичне відхилення від бажаного стану [11];

- 4) *метод лінійного керування*, де крутний момент розраховується на основі лінійної моделі системи, алгоритми пропорційно-інтегрально-диференціюючого регулятора використовують для обчислення керуючого сигналу, який залежить від поточного стану та помилки [11];
- 5) *метод нечіткого керування*, де залучаються нечіткі логічні правила для прийняття рішень щодо розрахунку крутного моменту. Цей метод використовує нечіткі множини та правил інтерпретації, що дозволяють враховувати нечіткість та неоднозначність вимірювань та умов систем [11].

Надійні методи керування є дуже важливими для забезпечення безпеки, стійкості та надійності систем:

- 1) *дублювання та резервування*. Цей метод полягає в наявності дублюючих або резервних компонентів або систем, які забезпечують продовження роботи у випадку відмови основних компонентів. Наприклад, можуть бути застосовані дубльовані датчики або виконавчі пристрої, які активуються автоматично, якщо основні компоненти недоступні або відмовляють [11];
- 2) *паралельне керування*. У цьому методі використовують кілька незалежних контролерів, які працюють паралельно та незалежно один від одного. Рішення приймається на основі голосування або комбінації вхідних сигналів від кожного контролера, чим забезпечується надійність та стійкість системи, навіть якщо один з контролерів буде видавати неправильний сигнал [11];
- 3) *діагностика та прогнозування відмов*. Цей підхід включає в себе використання діагностичних алгоритмів та методів прогнозування відмов для виявлення потенційних проблем чи відмов у системі, що дозволяє вжити завчасно заходів для усунення або керування виявленими проблемами, забезпечуючи надійність та безпеку системи. У цьому підході можна використовувати методи статичного аналізу, штучного інтелекту, машинного навчання тощо [11];
- 4) *резервне дублювання з реконфігурацією*. Цей метод включає наявність дубльованих компонентів, які можуть взаємодіяти та автоматично замінятися в разі відмови. Реконфігурація може бути здійснена шляхом переключення

на інший резервний компонент або використання алгоритмів переналаштування, що забезпечує неперервну роботу системи навіть при виникненні відмов [11];

- 5) *керування з використанням багатоагентних систем*. Багатоагентні системи використовують групу агентів, які працюють разом для досягнення бажаної мети. Це можуть бути розумні агенти або роботи, які співпрацюють та взаємодіють між собою. Багатоагентні системи можуть бути більш надійними та стійкими до відмов, так як можуть компенсувати непрацездатність окремих агентів або перерозподіляти завдання між ними [11].

Методи керування ковзним режимом використовуються для керування динамічними системами, які знаходяться або прагнуть до ковзного режиму (ковзний режим є станом, коли система функціонує в околі певного заданого режиму або екстремальної точки, і цей режим вважається бажаним або оптимальним [11]):

- 1) *метод керування ковзаючого режиму*. У цьому підході використовується ковзна поверхня, яка є штучно створеною поверхнею в просторі станів системи. Ковзна поверхня визначається таким чином, що коли система знаходиться на цій поверхні або ковзає по ній, то виконується бажана динаміка. Алгоритми цього методу гарантують, що система буде сходитися до ковзного режиму та притягуватися до ковзної поверхні, навіть коли наявна модель збурень чи невідомих параметрів[11];
- 2) *керування за критерієм оптимального ковзання*. Цей підхід використовує оптимальні ковзні поверхні, які мінімізують заданий критерій якості, такий як інтеграл похибки або квадратичне відхилення. Це дозволяє отримати керуючий сигнал, який мінімізує помилку та забезпечує оптимальний перехід до ковзного режиму [11];
- 3) *метод керування ковзним режимом на основі нечіткої логіки*, де ковзний режим визначається за допомогою нечітких правил та логічних відношень.

Ці методи дозволяють враховувати нечіткість та неоднозначність вимірювань та керуючих сигналів [11].

Методи нейронних мереж використовується для керування системами за допомогою штучних нейронних мереж:

- 1) *метод зворотного розповсюдження помилки*. Цей метод використовується для навчання нейронної мережі на основі зворотного розповсюдження помилки. Спочатку нейронна мережа навчається за допомогою тренувальних даних, а потім застосовується для керування системою. Під час роботи системи нейронна мережа отримує вхідні дані та генерує вихідні сигнали, які використовуються для керування системою [11];
- 2) *метод прямого поширення*. Цей метод використовується для керування системою на основі прямого поширення сигналів через нейронну мережу. Вхідні дані проходять через штучну нейронну мережу, кожен нейрон обчислює свої вагу та активаційну функцію, після чого на виході отримується керуючий сигнал для системи [11];
- 3) *метод рекурентних нейронних мереж* (тип нейронних мереж, в яких зв'язки між нейронами можуть бути циклічними). Вони дозволяють зберігати попередні стани мережі та враховувати історію вхідних даних. Цей метод широко використовуються для керування системами зі змінною динамікою, такими як обробка послідовностей, машинне навчання, керування рухом роботів тощо [11];
- 4) *метод глибинного навчання*. При глибинному навчанні використовуються нейронні мережі з багатьма шарами для навчання складних функцій та виявлення складних залежностей у даних. Цей підхід може застосовуватися для керування системами, де потрібно враховувати багато вхідних факторів та здійснювати складні рішення [11].

Методи нейронних мереж мають переваги у вирішенні складних завдань керування, а також адаптації до змінних умов. Вони можуть застосовуватися з метою моделювання нелінійних систем та покращення продуктивності і стійкості керованої системи [11].

1.1.5 Застосування мобільних роботів

Існує велика кількість сфер, де люди використовують роботів, постійно розширюються. Розглянемо нижче деякі приклади застосування:

1. Відомства надзвичайних ситуацій та екологічної безпеки

1.1. Дослідження небезпечних для людини зон та об'єктів:

- отримання різних проб, проведення різних видів моніторингу в агресивних середовищах (наприклад радіаційний чи хімічний моніторинг);
- обстеження зруйнованих об'єктів, кількісна оцінка пошкоджень будівель та обладнання [19].

1.2. Ліквідація наслідків надзвичайних ситуацій:

- гасіння пожеж в умовах високих температур;
- відключення обладнання в умовах агресивних середовищ;
- розрізання різноманітних металоконструкцій, фідерів, трубопроводів тощо;
- дегазація, дезінфекція, дезактивація приміщень та обладнання [19].

1.3. Рятувальні роботи:

- розгрібання завалів та розчистка проходів;
- транспортування продуктів життєзабезпечення;
- відчинення дверей, вікон та люків;
- евакуація потерпілих [19].

1.4. Роботи при іонізуючому випромінюванні:

- виявлення джерел та усунення причин радіоактивності;
- послаблення рівнів радіації;
- дезактивація критичних зон [19].

1.5. Знешкодження боєприпасів:

- розкриття пошкодженого упакування та вилучення боєприпасів;

- їх виявлення та транспортування до пунктів знешкодження;
- ліквідація боєприпасів, транспортування яких є неможливим або небезпечним [19].

1.6. Пожежогасіння:

- Виявлення джерел займання та задимлення;
- Транспортування та установка засобів для пожежогасіння [19].

2. Поліція:

2.1. Виявлення та ліквідація джерел небезпеки та контрабанди:

- обстеження приміщень, транспортних засобів та обладнання;
- рентгенівське просвічування вантажів, контейнерів та іншої тари [19].

2.2. Антитерористичні заходи та операції:

- виявлення, вилучення та знешкодження вибухових пристроїв та вибухонебезпечних предметів;
- транспортування об'єктів в безпечне місце або переміщення у спеціальні контейнери;
- підриг вибухонебезпечних об'єктів без прийняття участі в цьому людини;
- прослуховування радіотелефонних каналів;
- охорона об'єктів;
- патрулювання територій [19].

3. Відомства збройних сил та оборони

3.1. Розвідувальні операції:

- розвідка наземних цілей;
- розвідка місцевості та маршрутів; нанесення їх на карту;
- корегування стрільби;
- радіорозвідка та пеленгація радіостанцій [19].

3.2. Зв'язок:

- ретрансляція та радіоперехоплення;
- руйнування ліній зв'язку;

- створення завад на лінії передачі інформації;
- може слугувати як радіомаяк [19].

3.3. Бойове забезпечення:

- вогнева підтримка та прикриття атакуючих;
- ліквідація авто та бронетехніки, радіостанцій та командних пунктів;
- транспортування боєприпасів;
- евакуація постраждалих [19].

3.4. Інженерне забезпечення:

- прокладання маршрутів по мінному полю;
- ідентифікація та знешкодження вибухових пристроїв;
- гасіння техніки у разі загорання;
- мінування об'єктів та місцевості [19].

3.5. Диверсії:

- доставка, встановлення та активація вибухових пристроїв;
- руйнування трубопроводів, електромереж, підстанцій, мостів тощо [19].

4. Медицина

4.1. Діагностика:

- пошук та отримання інформації пацієнта щодо його стану здоров'я;
- вимірювання деяких показників, таких як пульс, тиск чи температура.

4.2. Допомога при операціях:

- виконання точних розрізів;
- віддалене керування інструментами.

4.3. Догляд за пацієнтами:

- надання інформації щодо стану пацієнта;
- допомога у пересуванні;
- доставка ліків.

Отже, МР давно стали великою частиною людського життя. Вони приносять багато користі в тих випадках, коли людина не здатна виконати поставлене

завдання або його виконання може ставити її життя під загрозу. Розвиток мобільних роботів є дуже перспективною тенденцією. Проте, завжди потрібно зберігати баланс між людським впливом та автоматизацією, а також брати до уваги етичні та соціальні аспекти використання подібних технологій.

1.2 Характеристика штучних нейронних мереж

1.2.1 Загальна інформація про нейронні мережі

Штучні нейронні мережі – це математичні моделі та програмні або апаратні реалізації, які базуються на принципах подання та обробки інформації, що використовуються в біологічних нейронних мережах (мережі нервових клітин живих організмів) [21].

Нервова система людини складається з клітин, відомих як нейрони, а також має надзвичайну складність, так як приблизно нейронів беруть участь у понад передавальних зв'язках, які можуть мати довжину від метра і більше. Хоча кожний нейрон має багато спільних характеристик з іншими клітинами, але має і унікальну здатність, а саме: прийом, обробка та передача електрохімічних сигналів по нервовим шляхам, які утворюють комунікаційну систему мозку [21].

Спрощена схема біологічного нейрона зображена на рис. 1.7. Дендрити, які схожі на гілки дерева, виходять від тіла і направляються до інших нейронів, де отримують сигнали в точках з'єднання, які називаються синапсами [21].

Вхідні сигнали, які було отримано синапсами, направляються до тіла нейрона. Тут вони підсумовуються, причому деякі входи намагаються збудити нейрон, тоді як інші намагаються запобігти його збудженню. Коли загальне збудження в тілі нейрона перевищує певний поріг, нейрон стає

збудженим і надсилає сигнал іншим нейронам через аксон, що є відростком на його виході [21].

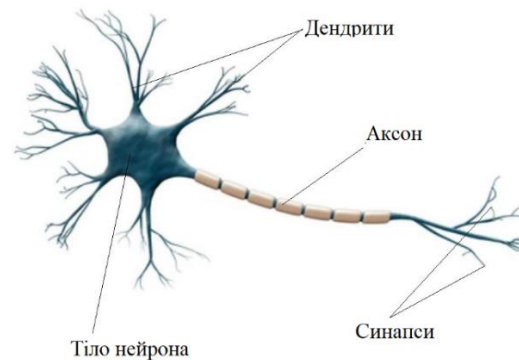


Рис. 1.7. Спрощена схема біологічного нейрона [20]

Більшість штучних нейронних мереж саме ці прості властивості і моделюють.

Штучний нейрон (також відомий як формальний нейрон або нейроподібний елемент) - це базовий обчислювальний пристрій або його модель, який має кілька вхідних сигналів і один вихідний. Він є основним будівельним блоком штучних нейронних мереж і відповідає за обчислення та передачу інформації [21].

Схему штучного нейрона можна побачити на рис. 1.8.

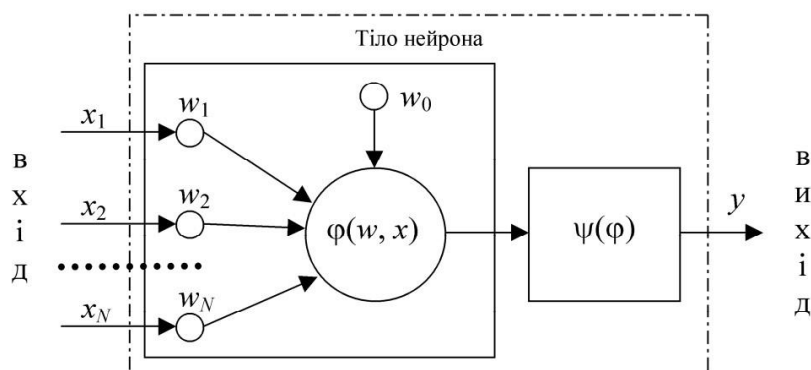


Рис. 1.8. Схема штучного нейрона [21]

На вхід одношарового нейрона надходить вектор вхідних сигналів $x = \{x_j\}$, $j = 1, 2, \dots, n$, де n – кількість входів [21].

Кожен вхідний сигнал x_j зважується (масштабуються певним чином) відносно зіставленої йому ваги зв'язку (вагового коефіцієнта) w_j , яка відповідає за моделювання перетворення сигналу у синапсі (міжнейронному контакті) [21].

Дискримінантна функція нейрона φ , яку ще називають ваговою або постсинаптичною, комбінує зважені вхідні сигнали для отримання постсинаптичного потенціалу, а також це значення подає до функції активації (передатної функції) ψ . Ця функція генерує скалярне значення на виході нейрона. Таким чином, формальний нейрон виконує перетворення вхідних сигналів, яке моделює процеси, які відбуваються у синапсах та тілі біологічного нейрона, і представляє скалярну функцію векторного аргументу [21].

Отже, математична модель функціонування штучного нейрона можна описати наступним співвідношенням [21]:

$$y = \psi [\varphi(w, x)], \quad (1.1)$$

де x – вектор вхідних сигналів (аргументів); y – значення на виході нейрона;

ψ – передатна функція; φ – вагова функція; $w = \{w_j\}$ – вектор, якій містить значення вагових коефіцієнтів w_j і значення зсуву w_0 [21].

Так як набір вагових коефіцієнтів нейрона w моделює його пам'ять, то нейрони можна розглядати як запам'ятовуючі пристрої або примітивні процесори, які здійснюють обчислення передатної функції на основі значення вагової функції вхідних сигналів та ваг [21].

1.2.2 Класифікація та види нейронних мереж

В основному нейронна мережа представляє собою сукупність нейронів, які зв'язані певним чином. Основними відмінностями нейромережевих моделей є способи зв'язку між нейронами, функції, які вони виконують, а також механізми та напрямки передачі сигналів по мережі [21].

Вид обробки інформації, який виконується НМ, залежить від характеристик нейроподібних елементів та особливостей архітектури мережі. Ці особливості включають топологію зв'язків між нейронами, вибір певних нейроподібних елементів для введення та виведення інформації, методи навчання мережі, наявність або відсутність конкуренції між нейронами, а також способи керування та синхронізації передачі інформації між нейронами [21].

Класифікувати НМ ми можемо наступним чином:

За **типом вхідної інформації**: *аналогові* та *бінарні*. Аналогові НМ використовують інформацію у формі дійсних чисел, тоді як бінарні НМ операційні з інформацією, яка представлена у двійковому форматі [21].

За **типом передатної функції**: *неперервні*, *дискретні* та *дискретно-неперервні*. Неперервні мережі, також відомі як аналогові, дійсні або диференційовані мережі, мають кожен елемент, який реалізує неперервно диференційовану функцію активації. Дискретні мережі, відомі також як бінарні або порогові, мають кожен елемент, який реалізує недиференційовану функцію активації. Дискретно-неперервні мережі містять елементи, що використовують як диференційовані, так і недиференційовані функції активації [21].

За **типом графа міжнейронних зв'язків**: *ациклічні* (без циклів) та *мережі з циклами* (рівноважні мережі з циклами і мережі з обмеженими циклами). Ациклічні мережі, які не містять циклів, відповідають графам без повторних зв'язків між нейронами. Рівноважні мережі з циклами мають замкнуті петлі, але всередині цих петель сигнали розподіляються таким чином, що немає нагромадження сигналів у певних нейронах. Мережі з обмеженими циклами мають замкнуті петлі, де

можливе нагромадження сигналів, що може спричиняти додаткові зміни в системі [21].

За **типом структур нейронів**: *гомогенні* (однорідні) та *гетерогенні* (неоднорідні) нейромережі. Гомогенні НМ складаються з нейронів одного типу, які мають однакову функцію активації. Усі нейрони в таких мережах виконують однакові обчислення. Гетерогенні нейромережі містять нейрони з різними функціями активації. Кожен нейрон може мати власну унікальну функцію активації, що дозволяє виконувати різноманітні обчислення в різних частинах мережі [21].

За **кількістю шарів нейронів**: *одношарові* та *багатошарові* нейромережі. Одношарові НМ складаються з одного шару нейронів. У таких мережах немає проміжних шарів між вхідними та вихідними шарами. Багатошарові НМ містять більше одного шару нейронів, які взаємопов'язані між собою. У них присутні проміжні шари, які передають і обробляють сигнали між вхідними та вихідними шарами. Багатошарові НМ здатні виконувати складніші обчислення та моделювати складніші відносини між вхідними та вихідними даними [21].

За **типом вагової функції**: *зважені* (із вагами зв'язків) та *без ваг зв'язків*. У мережах зважених зв'язків кожен зв'язок між нейронами має прикріплену до себе вагу, яка відображає його вплив на вихідний сигнал нейрона. Ці ваги дозволяють надавати різним зв'язкам різний ступінь важливості при обробці інформації. У мережах без ваг зв'язків всі зв'язки мають однакову вагу або взагалі не мають ваг. Це означає, що всі вхідні сигнали рівномірно впливають на вихідний сигнал нейрона без урахування їхньої різниці в важливості [21].

За **принципом синтезу**: ті, які *навчаються*, і ті, які *конструюються*. Мережі, які навчаються, є мережами, в яких графи міжнейронних зв'язків та ваги входів налаштовуються або змінюються під час процесу навчання. При виконанні методу навчання мережа адаптується до вхідних даних, щоб здійснювати бажані обчислення або виконувати конкретні завдання. Мережі, які конструюються, є мережами, в яких кількість і тип нейронів, граф міжнейронних зв'язків та ваги входів нейронів визначаються при створенні мережі, виходячи з постановки задачі

або специфікацій вимог. Ці мережі проектуються заздалегідь, без потреби в процесі навчання, і вони здатні виконувати визначені завдання згідно зі своєю архітектурою [21].

За **топологією зв'язків**: *повнов'язні* (інтерактивні), *неповнов'язні* (шаруваті, багат шарові, ієрархічні), *ієрархічно інтерактивні* та *слабов'язні* (з локальними зв'язками). У повнов'язних мережах кожен нейрон пов'язаний з кожним іншим нейроном, включаючи самого себе. Всі нейрони обмінюються сигналами між собою, і вихідні сигнали мережі можуть бути залежними від вихідних сигналів всіх або певної підмножини нейронів після кількох тактів функціонування мережі. Неповнов'язні мережі зазвичай мають шарувату структуру. У них відмінні латеральні (бічні) зв'язки, які з'єднують нейрони в межах одного шару, і проєкційні (аферентні) зв'язки, які з'єднують шари нейронів. Деякі нейрони можуть мати додаткові зовнішні входи, що формують рецепторне поле. Інформація рухається від вхідного до вихідного прошарку, пройшовши через точки галуження між шарами. Кожен наступний шар забезпечує більш високий рівень обробки інформації, ніж попередній. Кількість нейронів у кожному шарі може бути будь-якою і не пов'язана з кількістю нейронів в інших шарах. Ієрархічно інтерактивні мережі мають шари різного рівня, які взаємодіють двосторонніми зв'язками. Крім того, вони можуть мати зв'язки між елементами в межах одного шару. Структура зв'язків в цих мережах не є однорідною, як в повнозв'язних мережах. Такі мережі показують ієрархічну організацію, де шари різного рівня взаємодіють між собою, а також взаємодіють в межах кожного шару. У слабов'язних мережах нейрони розташовані вузлами прямокутної або гексагональної решітки. Кожен нейрон має зв'язки зі своїми найближчими сусідами, кількість яких може бути різною. Зазвичай використовуються решітки з околom фон Неймана (4 сусіди), Голео (6 сусідів) або Мура (8 сусідів). Такі мережі характеризуються локальними взаємодіями між нейронами, де кожен нейрон обмінюється інформацією лише зі своїми найближчими сусідами [21].

За **характером зв'язків**: *мережі прямого поширення* та *рекурентні*. Мережі прямого поширення не мають зворотних зв'язків, і сигнал поширюється через

мережу лише в одному напрямку – від входу до виходу. У таких мережах інформація обробляється без зворотного впливу, і вихідні сигнали залежать лише від вхідних даних. Рекурентні мережі мають зворотні зв'язки, що дозволяють зворотне поширення інформації між різними шарами мережі. Вони характеризуються як прямим, так і зворотним поширенням інформації між шарами нейронної мережі. У рекурентних мережах інформація може циркулювати, і вихідні значення мережі можуть змінюватися до досягнення стану рівноваги [21].

За *характером поділу зв'язків*: *немонотонні* та *монотонні* мережі. Немонотонні мережі не дозволяють встановити однозначний зв'язок між зміною внутрішніх параметрів мережі і вихідним сигналом. Тобто, зміна будь-якого внутрішнього параметра не має визначеного впливу на вихідний сигнал мережі. Вони характеризуються немонотонними залежностями між параметрами і вихідним сигналом. У монотонних мережах кожен шар, за винятком останнього, поділяється на два блоки: збудливий і гальмуючий. Зв'язки в мережі організовані таким чином, що елементи збудливої частини шару збуджують елементи збудливої частини наступного шару і гальмують елементи наступного шару. Так само, гальмуючі елементи збуджують гальмуючі елементи і гальмують збуджуючі елементи наступного шару. Назви «гальмуючий» і «збуджуючий» стосуються впливу елементів на вихідні сигнали. Для нейронів монотонних мереж важлива монотонна залежність вихідного сигналу нейрона від параметрів вхідних сигналів [21].

За *типом методу навчання*: *мережі з динамічними зв'язками й ітеративним навчанням, заснованому на принципі корекції помилок*, та *мережі з фіксованими зв'язками і неітеративним навчанням*. Мережі з динамічними зв'язками й ітеративним навчанням базуються на моделі повторення шляхів нервового збудження, яку запропонували фізіологи. Основний метод навчання в цих мережах — зворотне поширення помилки, який широко використовується в сучасних нейрокомп'ютерах. Головною перевагою цього методу є його асимптотична збіжність, що гарантує потенційну досяжність необхідної точності відповіді мережі. Однак недоліком цього методу є потреба в багаторазових

ітераціях навчання на всьому обсязі даних. Це вимагає значних обчислювальних ресурсів і часу, що обмежує можливість застосування цього методу в адаптивних системах реального часу. Мережі з фіксованими зв'язками і неітеративним навчанням використовують методи прямого обчислення значень матриці зв'язків, що дозволяє запам'ятовувати інформацію без необхідності в ітераціях навчання. Це прискорює процес навчання порівняно з ітеративними методами. Однак такі мережі характеризуються великою кількістю нейронів і низьким обсягом інформації, яку можна запам'ятати. Через це вони ще не знаходять широкого застосування і застосовуються в основному в дослідницьких проектах. Мережі з фіксованими зв'язками і неітеративним навчанням мають потенціал для прискорення процесу навчання, але потребують подальшого дослідження і вдосконалення, щоб стати ефективними в реальних застосуваннях [21].

Область застосування НМ також є обширною. Їх використовують для розпізнання сигналів, мови, зображень та тексту; для біомедичної діагностики; для моделювання залежностей в природничих науках та техніці; для соціально-економічного прогнозування; для керування в техніці та економіці; для побудови інформаційно-пошукових систем та для криптографії [21].

Отже, НМ є потужним інструментом, який здатний моделювати складні залежності і здійснювати різноманітні завдання в різних галузях. Вони можуть бути класифіковані за різними ознаками. Залежно від конкретних потреб і завдань, можна вибрати підходящий тип мережі нейронів, який найкраще підходить для розв'язання проблеми. Використання мереж нейронів стає все більш поширеним у різних сферах людської діяльності. Розуміння основних принципів та типів мереж нейронів дозволяє використовувати їх потенціал для розв'язання різноманітних завдань у сучасному світі.

Важливим є застосування НМ для завдань керування процесами та об'єктами, зокрема багатоцільовими мобільними роботами. Для систем адаптивного керування, у випадках неповної інформації про рухомий об'єкт та/чи умови його руху, при наявності невизначеності параметрів чи характеристик об'єкта використання нейромережових регуляторів забезпечує ефективність

керування [22-24]. Тому важливим завданням для створення систем керування МР у різних фізичних середовищах є синтез регуляторів на основі штучних нейронних мереж.

Висновки до розділу

У даному розділі було виконано огляд мобільних роботів. Їхні типи та загальний розвиток як рухомого об'єкта. Було розглянуто сфери людського життя, де вони можуть бути використані.

Було зроблено огляд на системи руху, так як вони є дуже важливою складовою мобільних роботів, адже саме за допомогою них робот може пересуватися в просторі.

Особливу увагу приділили саме системам керування, так як завдяки ним мобільний робот здатен виконувати поставленні задачі.

Після чого розглянули, що собою представляють нейронні мережі. Після чого було надано їхню класифікацію. Як можна побачити, класифікувати нейронні мережі можна великою кількістю способів. Після чого на основі цих нейронних мереж можна отримати регулятор.

Так як використання нейронних мереж набуває великої популярності через свою зручність, продуктивність та великий асортимент вибору, то розробка нейромережових регуляторів є важливим завданням при створенні мобільних роботів та синтезі систем керування ними.

РОЗДІЛ 2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ MATLAB ДЛЯ СИНТЕЗУ НЕЙРОТЕХНІЧНИХ РЕГУЛЯТОРІВ

2.1 Загальна характеристика Deep Learning Toolbox

Для синтезу та аналізу систем керування та, зокрема нейромережевих регуляторів, в задачах дослідження найчастіше використовується програмний пакет MATLAB, створений для розв'язання багатьох математичних та інженерних розрахунків, комп'ютерного моделювання [25].

Пакет *Deep Learning Toolbox* (раніше *Neural Networks*) містить ряд блоків, які можуть бути безпосередньо використані для побудови НМ в середовищі Simulink. Основні нейромережеві бібліотеки блоків зображені на рис. 2.1.

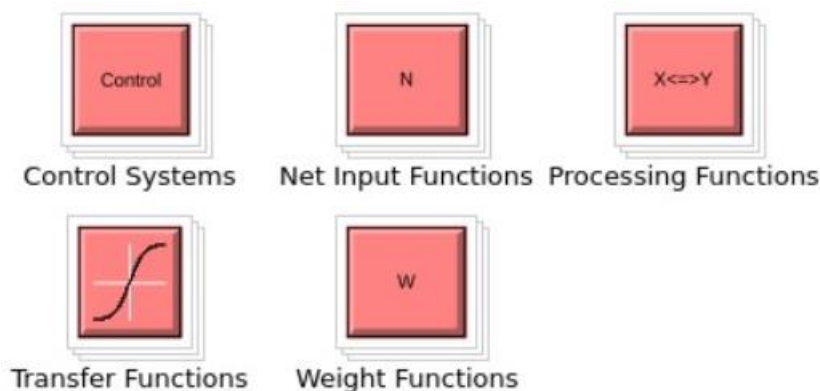


Рис. 2.1. Основні нейромережеві блоки Simulink

Блоки перетворення входів мережі (*Net Input Function*). Функції, які входять до цієї групи, відповідають за обчислення накопичення потенціалу нейрона шляхом поелементного добутку або суми зважених входів нейрона. Вони також враховують обчислення похідних від таких добутків або сум [26]. Зображені ці блоки на рис. 2.2.

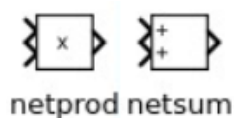


Рис. 2.2. Бібліотека блоків перетворення сигналів

Функції обробки вхідних даних або вихідних результатів нейромережі (*Processing Functions*) (зображені ці блоки на рис. 2.3):

Функція *fixunknowns* використовується для заміни невідомих значень вхідних даних на певні значення. Вона дозволяє обрати стратегію заміни, таку як заміна невідомих значень на середнє значення, медіану або певне константне значення. Застосовується вона до вхідних даних перед навчанням моделі, щоб усунути проблему невідомих значень, які можуть мати негативний вплив на навчання та прогнозування. Функція *fixunknowns_reverse* дозволяє відновити ці значення, використовуючи стратегію заміни, яку використовували під час навчання моделі [26].

Функція *mapminmax* використовується для нормалізації вхідних даних шляхом перетворення їх до нового діапазону значень [0;1]. Вона знаходить мінімальне та максимальне значення вхідних даних і використовує їх для перетворення кожного значення в діапазон [0;1]. Це допомагає забезпечити стандартизовані та нормалізовані дані, що часто поліпшує процес навчання нейромережі. Функція *mapminmax_reverse* застосовується для відновлення даних до початкового стану [26].

Функція *lvqoutputs* приймає вхідні дані та поточний стан моделі та обчислює вихідні значення нейромережі для заданого вхідного набору шляхом застосування алгоритму квантування навчальних векторів (*Learning Vector Quantization*). Функція *lvqoutputs_reverse* застосовується для відновлення даних до початкового стану [26].

Функція *mapstd* використовується для стандартизації вхідних даних шляхом віднімання середнього значення, а після цього ділиться на стандартне відхилення, що допомагає забезпечити нульове середнє значення та одиничне стандартне

відхилення в нормалізованих даних. Функція *mapstd_reverse* застосовується для відновлення даних до початкового стану [26].

Функція *processpca* використовує метод головних компонентів (Principal Component Analysis) для зменшення розмірності даних. Функція *processpca_reverse* застосовується для відновлення даних до початкового стану [26].

Функція *removeconstantrows* перевіряє кожен рядок, видаляючи усі з постійними значеннями. Це допомагає знизити шум та зайву інформацію в даних, зберігаючи лише рядки з варіацією значень. Так як видалені рядки неможливо відновити, то функція *removeconstantrows_reverse* використовується для оцінки впливу видалення постійних рядків на початкові дані. Вона може повернути остаточну статистику про кількість видалених рядків та їх відсоткове співвідношення до загальної кількості рядків у вихідних даних [26].

Функції *removerows* та *removerows_reverse* виконують ті ж самі дії, що і функції *removeconstantrows* та *removeconstantrows_reverse*, але не з постійними значеннями, а з заданими [26].

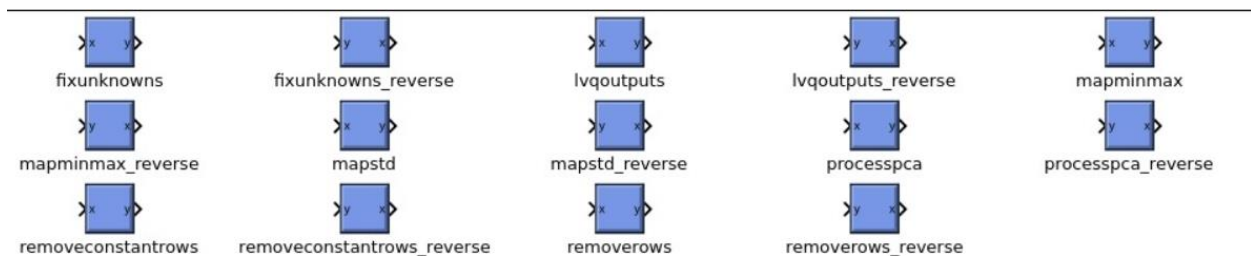


Рис. 2.3. Бібліотека блоків обробки сигналів

У блоках бібліотеки передатної функції (Transfer Function) наявні різні функції перетворення. Вони допомагають моделювати різноманітні типи передатних функцій [26]. Зображені ці блоки на рис. 2.4.



Рис. 2.4. Бібліотека блоків передатної функції

У бібліотеці вагової функції (Weight Function) наявні наступні блоки функцій (зображені ці блоки на рис. 2.5):

Dist – функція обчислення відстані між двома векторами.

Dotprod – функція обчислення дотичного (скалярного) добутку між двома векторами.

Negdist – функція обчислення від’ємної відстані між двома векторами.

Normprod – функція обчислення нормалізованого добутку двох векторів або матриць [26].

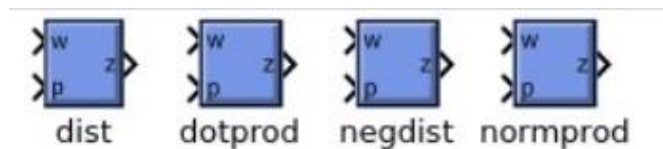


Рис. 2.5. Бібліотека блоків передатної функції

Також серед різноманіття бібліотек є бібліотека блоків нейромережових регуляторів, яка зображена на рис. 2.6. Там представлені нейромережові регулятори, які мають різну структуру та підхід: регулятор на основі еталонної моделі (Model Reference Controller), регулятор на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) та регулятор з передбаченням (Neural Network Predictive Controller). Також у цій бібліотеці наявний блок для перегляду результатів (X(2Y) Graph) [26].

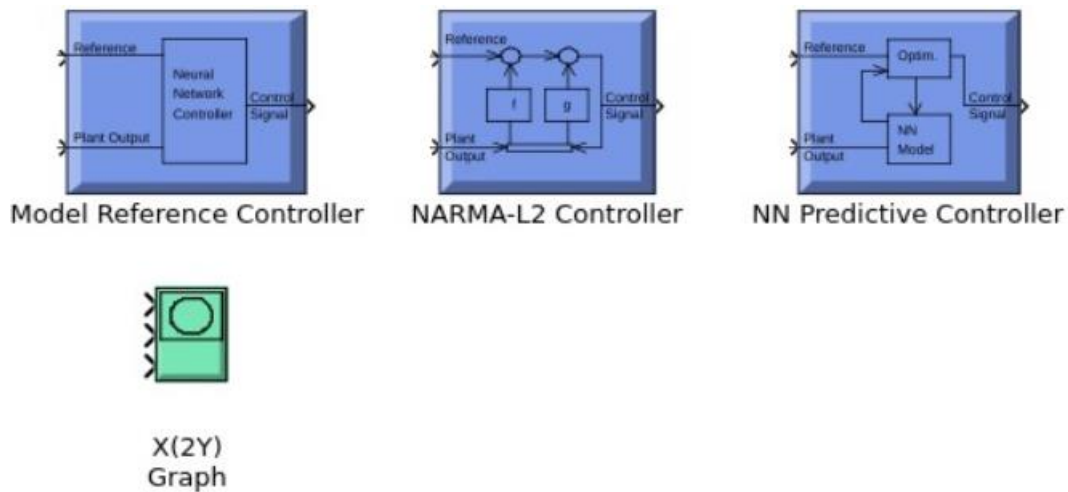


Рис. 2.6. Бібліотека блоків нейромережевих регуляторів

Отже, розібравши різні функції, можемо сказати, що реалізація НМ у середовищі MATLAB має великі можливості завдяки різноманітним функціям та *Deep Learning Toolbox*. З кожним днем розвиток в цьому напрямі дає нам все більш ефективні методи вирішення поставленого перед нами завдання зручнішим для нас способом.

2.2 Синтез та аналіз регулятора на основі еталонної моделі в Simulink

2.2.1 Модель об'єкта керування

Регулятор на основі еталонної моделі (Model Reference Controller, MRC) – це нейронна мережа, яку було навчено керувати об'єктом таким чином, щоб він відстежував поведінку еталонної моделі. Модель керованого об'єкта використовується при налаштуванні параметра регулятора [28].

Такий регулятор не потребує великого об'єму розрахунків, а архітектура регулятора з еталонною моделлю вимагає навчання нейронної мережі об'єкта

керування і нейронної мережі регулятора. До того ж, навчання регулятора є доволі складним, так як навчання ґрунтується на динамічному варіанті методу зворотного розповсюдження помилки [28].

На рис. 2.7 зображена структурна схема системи складається з еталонної моделі, яка задає бажану траєкторію руху об'єкта керування, та двох нейронних мереж: для регулятора і для моделі об'єкта керування. Мета цього навчання полягає у тому, щоб рух об'єкта керування відстежував вихід еталонної моделі [28].

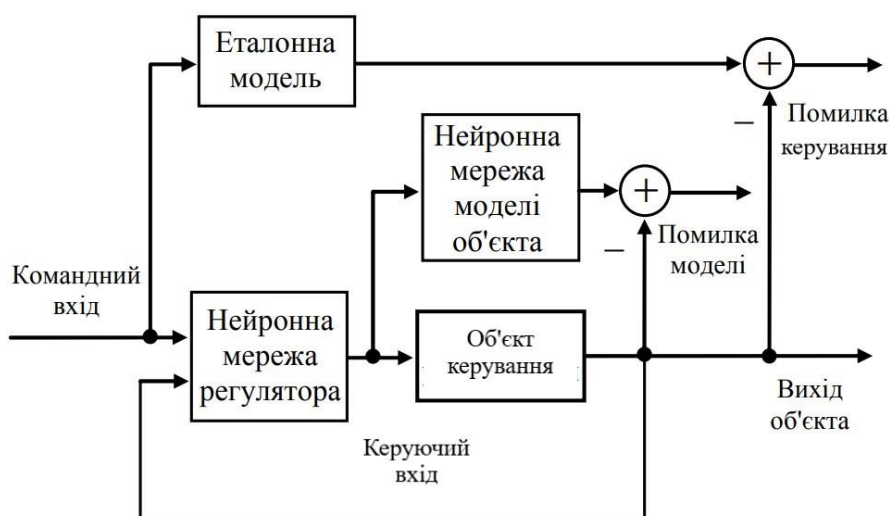


Рис. 2.7. Структурна схема MRC [24]

У розрахунках нейронна мережа використовується для визначення керуючого сигналу, який потім буде порівнюватися з еталонним. Тому потрібно налаштувати параметри нейронної мережі таким чином, щоб розрахований та еталонний сигнали співпадали. Використовувати можна різні методи такі як метод зворотного розповсюдження помилки чи градієнтного спуску [28].

Зазвичай подібний регулятор використовують для систем, у яких не так багато входів та виходів, у яких є чітка еталонна модель, які легко передбачити [28].

Тому у даній роботі ми використаємо саме цей контролер, взявши за об'єкт керування рух простої одноланкової руки робота, який зображено на рис. 2.8.

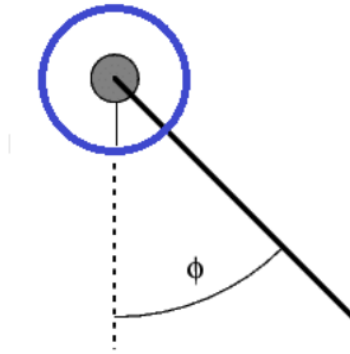


Рис. 2.8. Рух простої одноланкової руки робота [28]

Запишемо рівняння, яке буде описувати рух цієї руки:

$$\frac{d^2\phi}{dt^2} = -10\sin\phi - 2\frac{d\phi}{dt} + u, \quad (2.1)$$

де ϕ — кут плеча, а u — крутний момент, який створюється двигуном постійного струму.

З рівняння (2.1) отримуємо рівняння, яке описує еталонну модель, яку буде намагатися повторити рука робота:

$$\frac{d^2y_p}{dt^2} = -9y_p - 6\frac{dy_p}{dt} + 9p, \quad (2.2)$$

де p_r — вихід еталонної моделі; p — вхідний еталонний сигнал.

Реалізуємо цю систему у Simulink.

Для цього складаємо схему ОК, яка зображена на рис. 2.9.

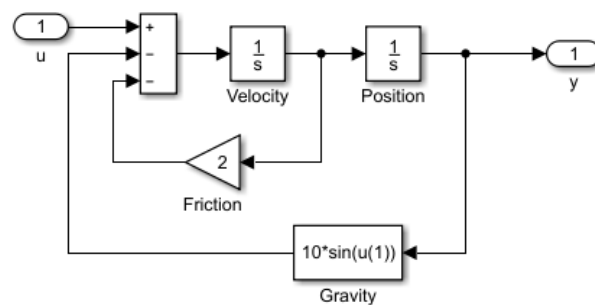


Рис. 2.9. Схема об'єкта керування

2.2.2 Навчання НМ регулятора

Наступним кроком є обрання у пакеті *Deep Learning Toolbox* потрібного нам блоку контролера, який має назву Model Reference Controller, а також задаємо вхідний сигнал через блок Random Reference. Ми будемо подавати на вхід випадковий сигнал у межах [20;23], щоб мати змогу краще провести оцінку та тестування працездатності й ефективності контролера, так як при рандомному сигналі створюються різноманітні вхідні сигнали з різними характеристиками.

Також цей тип сигналу може допомогти виявити можливі проблеми або недоліки в контролері, такі як нестабільність чи погана пристосованість до зміни умов тощо.

Отримана схема зображена на рис. 2.10.

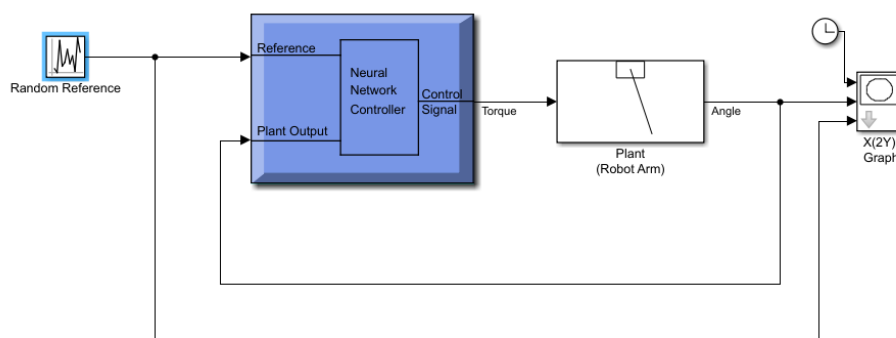


Рис. 2.10. Схема керування з використанням нейромережевого контролера на основі еталонної моделі

Тепер можемо почати навчання контролера. Для цього натискаємо на нього кілька разів. Та бачимо вікно налаштування, яке зображено на рис. 2.11. Так як тренування нейроконтролера вимагає чіткого розуміння методики виконання процесу, а також великого досвіду в цій справі, то ми скористаємося вже підібраними параметрами [30].

Рис. 2.11. Графічний інтерфейс налаштувань Model Reference Control

У вікні Model Reference Control (Модельно-еталонне керування) задаємо наступні параметри [30]:

Архітектура мережі (Network Architecture)

Розмір прихованого шару (Size of Hidden Layer) = 13.

Вхідні еталонні без затримок (No.Delayed Reference Inputs) = 2.

Вихідні сигнали контролера без затримок (No.Delayed Controller Outputs) = 1.

Вихідні сигнали об'єкта без затримок (No.Delayed Plant Outputs) = 2.

Дані для тренування(Training Data)

Максимальне еталонне значення (Maximum Reference Value) = 0,7.

Мінімальне еталонне значення (Minimum Reference Value) = - 0,7.

Максимальне значення інтервалу (Maximum Reference Value) = 2.

Мінімальне значення інтервалу (Minimum Interval Value) = 0,1.

Навчальні вибірки контролера (Controller Training Samples) = 6000.

Параметри для тренування(Training Parameters)

Епохи тренування контролера (Controller Training Epochs) = 10.

Сегменти тренування контролера (Controller Training Segments) = 30.

Як вже згадувалося вище, MRC має тренувати 2 мережі, тому натискаємо на Plant Identification (Ідентифікація об'єкта). Ми бачимо вікно налаштувань, зображене на рис. 2.12.

Рис. 2.12. Графічний інтерфейс налаштувань Plant Identification

У вікні Plant Identification (Ідентифікація об'єкта) ми задаємо наступні параметри [30]:

Архітектура мережі (Network Architecture)

Розмір прихованого шару (Size of Hidden Layer) = 10.

Інтервал дискретизації (Simpling Interval) = 0.05.

Вхідні сигнали без затримок на об'єкті (No.Delayed Plant Inputs) = 2.

Вихідні сигнали без затримок на об'єкті (No.Delayed Plant Outputs) = 2.

Дані для тренування(Training Data)

Навчальні вибірки (Training Samples) = 10000.

Максимальний вхідний сигнал (Maximum Plant Inputs) = 15.

Мінімальний вхідний сигнал (Minimum Plant Inputs) = -15.

Максимальне значення інтервалу (Maximum Interval Value) = 2.

Мінімальне значення інтервалу (Minimum Interval Value) = 0,1.

Задаємо обмеження вихідних даних (Limit Output Data):

Максимальний вихідний сигнал (Maximum Plant Inputs) = 3,1.

Мінімальний вихідний сигнал (Minimum Plant Inputs) = -3,1.

Параметри для тренування (Training Parameters)

Епохи тренування (Training Epochs) = 300.

Задаємо функцію тренування НМ, після чого натискаємо Generate Training Data (Генерація навчальних даних). Тепер чекаємо, коли процес закінчиться (процес може займати значний проміжок часу).

На рис. 2.13 ми можемо побачити приклад результату генерації навчальних даних.

Отримавши результати, ми можемо або прийняти дані, якщо ми вважаємо, що отримані результати задовольняють встановлені критерії та придатні для використання у подальшій роботі, або відхилити дані, якщо дані не задовольняють встановлені вимоги, наприклад через помилки, шум, відхилення від норми тощо.

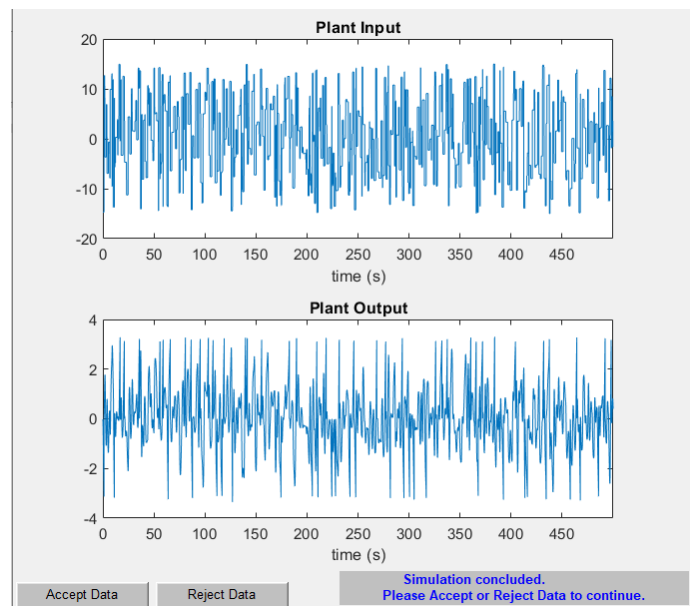


Рис. 2.13. Результат генерації навчальних даних об'єкта

Отримані результати на рис. 2.13 є прийнятними, тому натискаємо Accept Data (прийняти дані). Тепер можна запускати тренування мережі, для цього

натискаємо відповідну кнопку Train Network. Процес тренування ми можемо бачити та контролювати у інтерфейсі, який зображено на рис. 2.14.

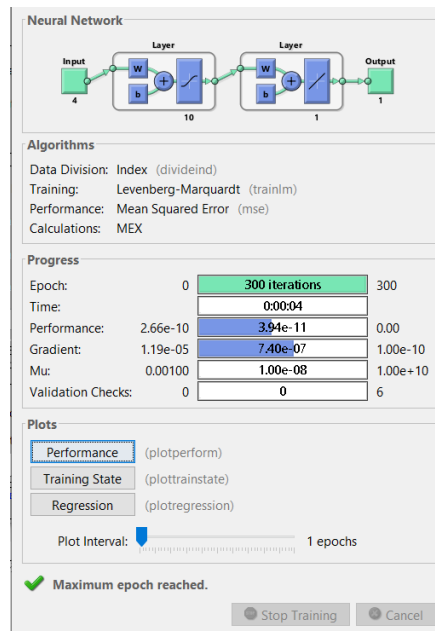


Рис. 2.14. Графічний інтерфейс для тренування нейронної мережі об'єкта

Після завершення процесу ми можемо бачити тренувальні дані (Training data), які зображені на рис.2.15, тестові дані (Testing data), які зображені на рис. 2.16, та дані валідації (Validation data) які зображені на рис. 2.17.

Тренувальні дані використовуються для оновлення ваг та налаштування параметрів мережі таким чином, щоб мережа могла навчитися прогнозувати правильні відповіді.

Тестові дані використовуються для оцінки загальної продуктивності навченої моделі після тренування. Тестові дані не використовуються під час тренування, а використовуються після нього для оцінки точності та ефективності моделі на незалежних від тренувальних даних.

Дані валідації допомагають оцінити, наскільки добре модель здатна узагальнювати нові дані, які вона не бачила під час тренування. Ці дані дозволяють виявити перенавчання та визначити оптимальні значення гіперпараметрів моделі.

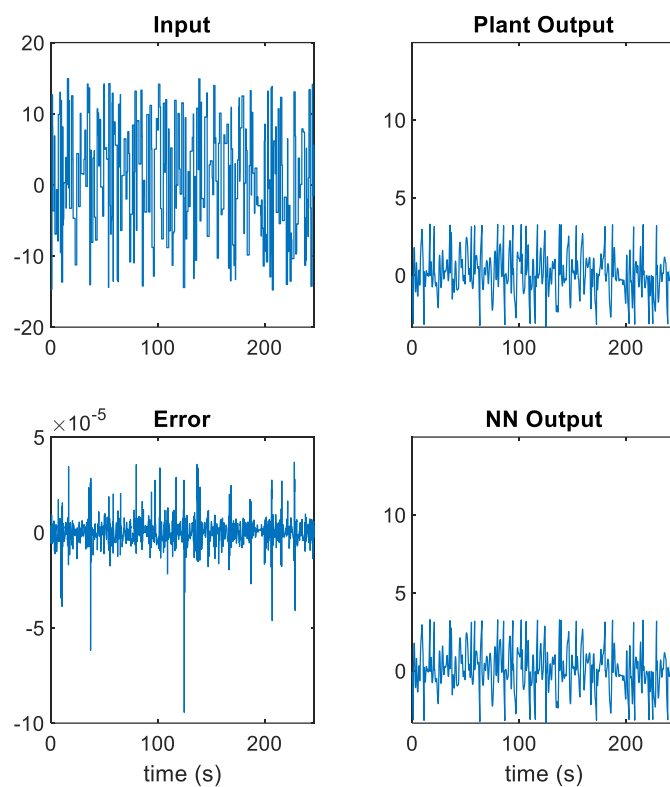


Рис. 2.15. Отримані тренувальні дані

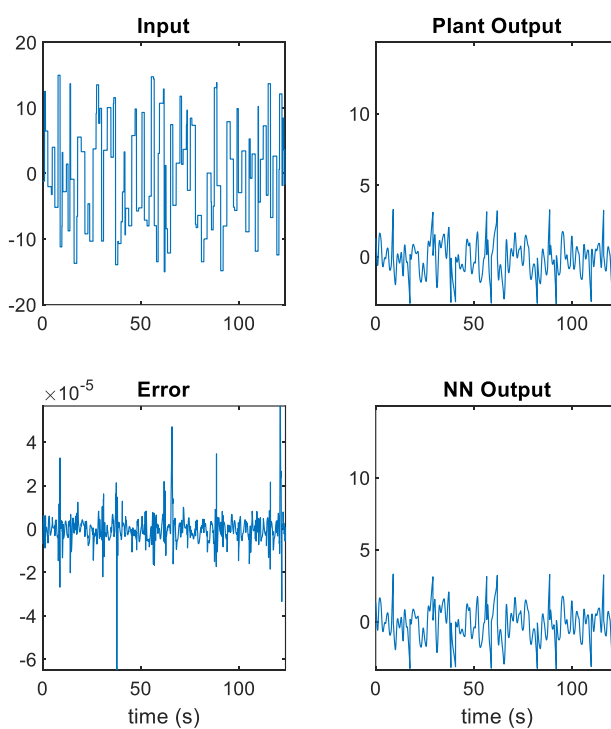


Рис. 2.16. Отримані тестові дані

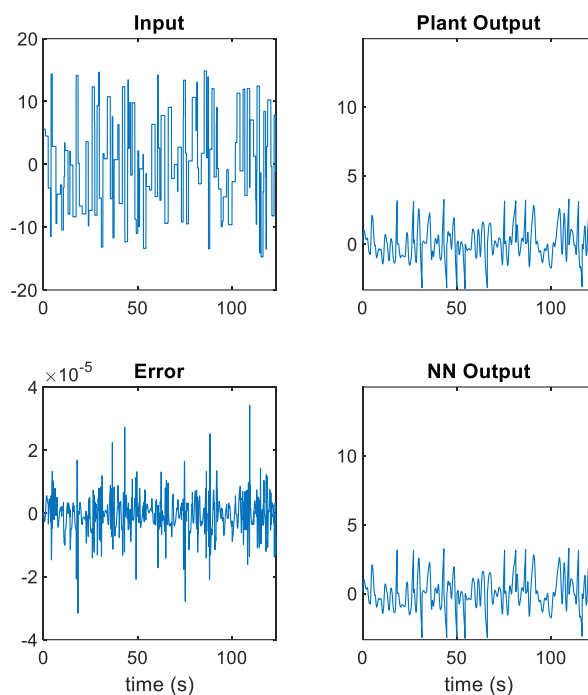


Рис. 2.17. Отримані дані валідації

Отримані результати нас задовольняють, якщо на виході ми маємо однакові результати.

Аналізуючи результати з рис. 2.15, рис. 2.16 та рис. 2.17, ми бачимо, що тренування було проведено правильно, так як на всіх трьох рисунках вихідні графіки є ідентичними, значення помилки є на п'ять порядків меншим за значення вхідних вихідних сигналів. Тому закриваємо графічні вікна та у вікні Plant Identification натискаємо ok.

Повертаємося до графічного інтерфейсу Model Reference Control. Так як ми закінчили тренування НМ об'єкта керування, то проведемо тренування мережі контролера. Натискаємо Generate Training Data (Генерація навчальних даних). Тепер чекаємо, коли процес закінчиться (процес може займати значний проміжок часу).

На рис. 2.18 наведено результат генерації навчальних даних.

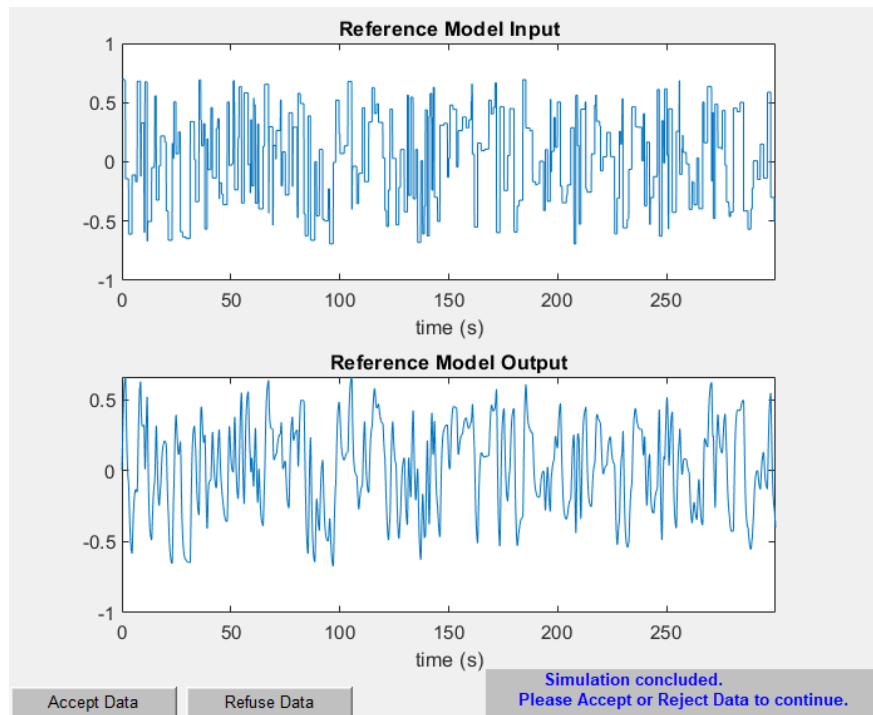


Рис. 2.18. Результат генерації навчальних даних контролера

Отримані результати на рис. 2.18 є прийнятними, тому натискаємо Ассерпт Data (прийняти дані). Тепер можна запускати тренування мережі, для цього натискаємо відповідну кнопку Train Network. Процес тренування ми можемо бачити та контролювати у інтерфейсі, який зображено на рис. 2.19.

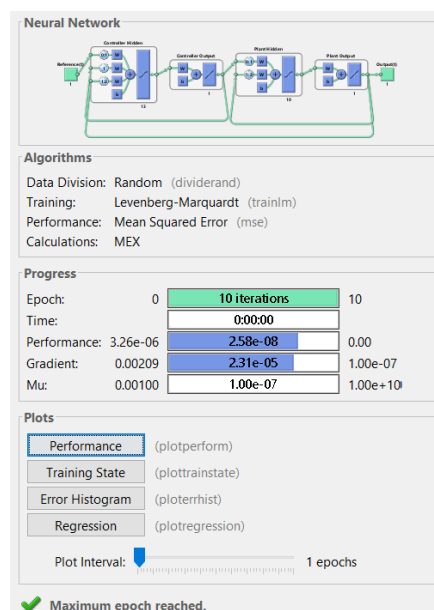


Рис. 2.19. Графічний інтерфейс для тренування нейронної мережі контролера

2.2.3 Аналіз ефективності НМ регулятора після навчання

Після закінчення процесу тренування отримуємо реакцію системи на бажаний сигнал, яку зображено на рис. 2.20.

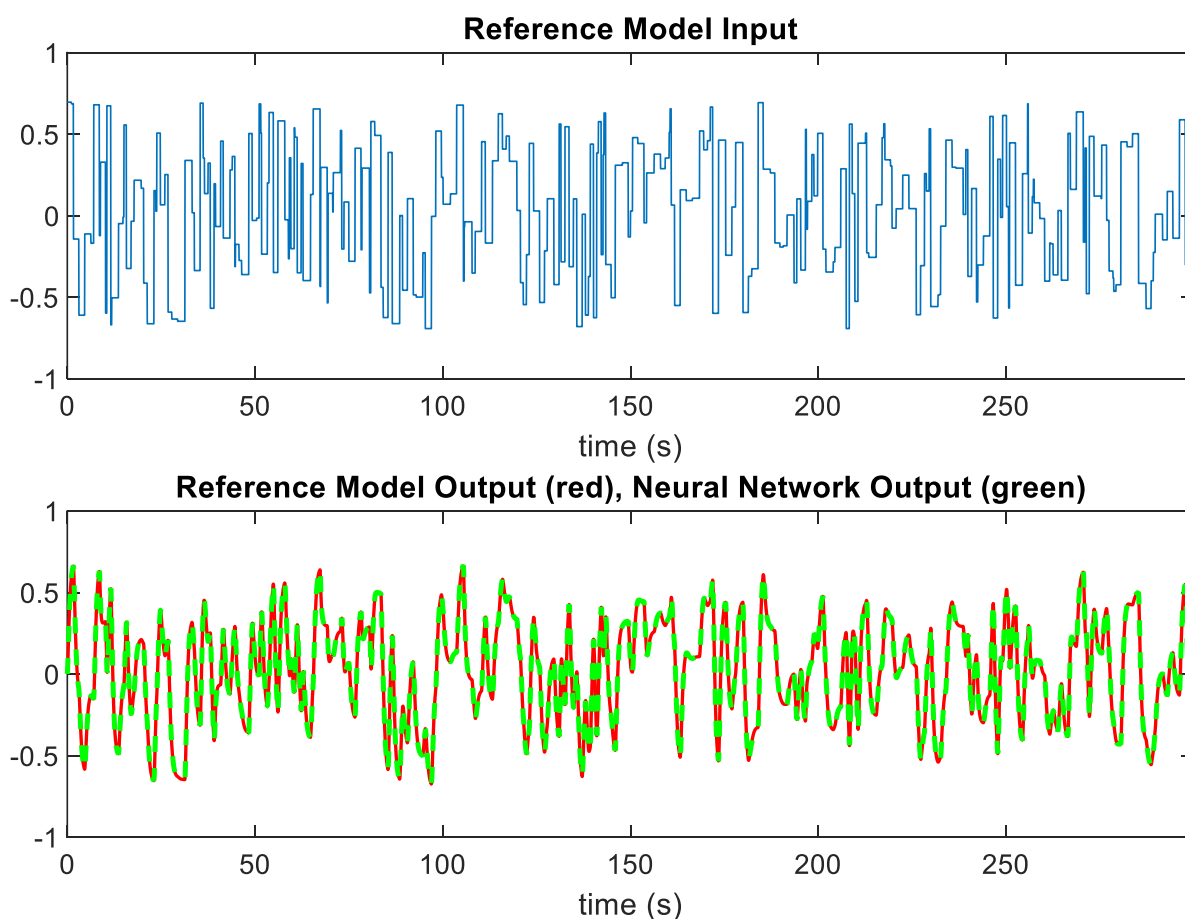


Рис. 2.20. Результати тренування мережі Model Reference Controller

Аналізуючи отримані результати на рис. 2.20, можемо бачити що вихідний сигнал еталонної моделі та навченої мережі контролера співпадають, значення помилки є на п'ять порядків меншим за значення вхідних вихідних сигналів, що вказує на правильність проведення тренування.

Так як результати є прийнятними, то зберігаємо усі налаштування, чим завершуємо тренування контролера.

Тепер закриваємо усі вікна та повертаємося до нашої схеми. Задаємо час моделювання. Я обрала 150 с. та запускаємо моделювання системи. Чекаємо, поки завершиться процес моделювання. Після чого отримуємо результати, які зображено на рис. 2.21.

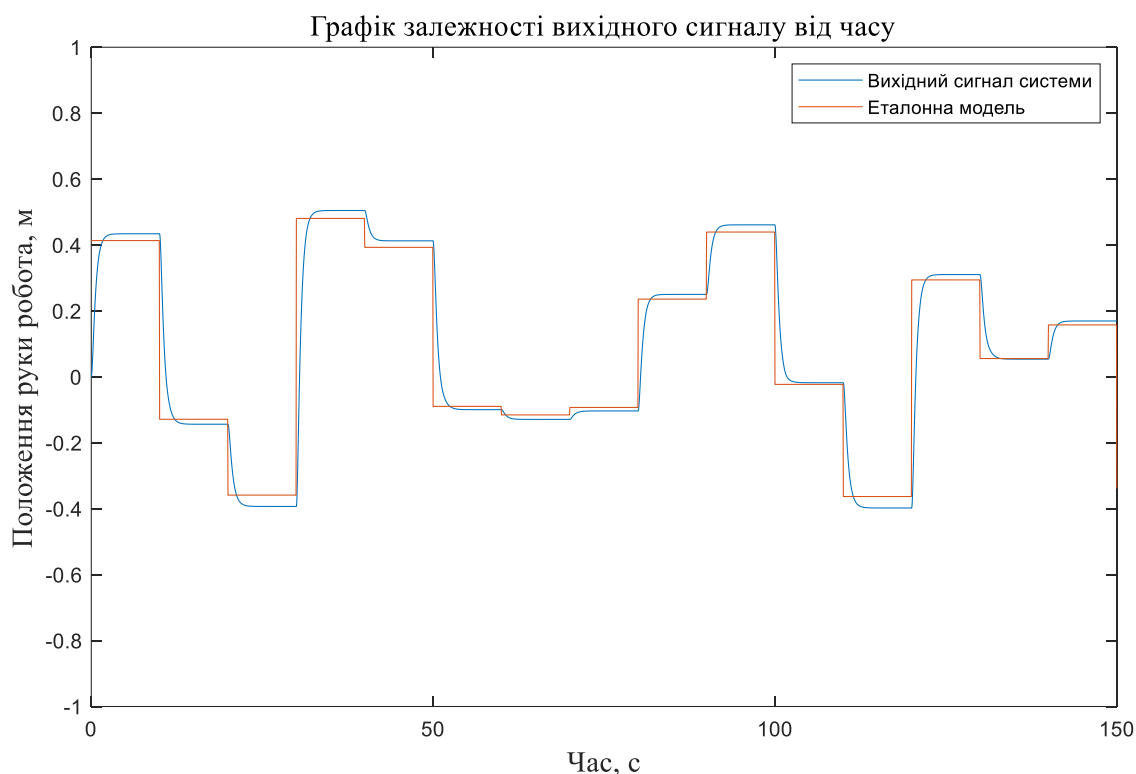


Рис. 2.21. Результати моделювання

Аналізуючи отримані результати на рис. 2.21, можна зробити висновок, що система постійно намагається досягти бажаного значення. Ми бачимо, що вихідний сигнал системи має відповідність з еталонним, що свідчить про високу якість слідування та точне керування системою. Також можна помітити, що система швидко реагує на зміни вхідного випадкового сигналу, що свідчить про високу та ефективну адаптивність регулятора та здатність системи до швидкого реагування на зміни.

Також завдяки проведенню цього моделювання можна оцінити якість слідування системи до еталонного чи стабільність системи. Ми можемо виявити

можливі проблеми, такі як затримка, перерегулювання, коливання, для виправлення, що допоможе нам в подальшому вдосконаленні контролера чи ОК.

Отже, ми провели синтез нейромережевого контролера на основі еталонної моделі та реалізували систему керування рухом простої одноланкової руки робота. Нам вдалося отримати після моделювання графік залежності вихідного сигналу від часу. Це нам допомагає з більшою ефективністю досліджувати та керувати системами в реальному часі, забезпечуючи високу точність та надійність керування.

2.3 Синтез та аналіз регулятора з прогнозуванням в Simulink

2.3.1 Модель об'єкту керування

Регулятор з прогнозуванням (Neural Network Predictive Controller, NN Predictive Controller, NNPC) використовує при роботі модель нейронної мережі для прогнозування майбутніх реакцій об'єкта керування на потенційні збурення в системі. Алгоритм оптимізації обчислює керуючі сигнали, які мінімізують різницю між бажаними та дійсними змінами сигналу на виході моделі, чим оптимізує поведінку об'єкта на заданому часовому інтервалі. Регулятор вимагає великої кількості обчислень, так як для розрахунку оптимального закону керування оптимізація відбувається на кожному такті керування [27].

Проектування складається з двох етапів: етапу ідентифікації об'єкту керування нейрорегулятора та етапу синтезу закону керування. На першому етапі розробляється модель об'єкта у вигляді нейронної мережі, а на другому— виконується синтез регулятора на основі цієї мережі [27].

Структура регулятора, яка зображена на рис. 2.22, має такі складові:

НМ, яка є основною складовою регулятора, вона, отримуючи вхідні параметри системи, генерує можливі прогнози щодо наступних значень системи.

Блок оптимізації, який на базі отриманих прогнозів визначає найбільш оптимальний спосіб керувати системою.

Блок керування, який надсилає сигнали керування системою, спираючись на отримані оптимальні значення.

Блок зворотного зв'язку, який на основі реальних значень, які було отримано після виконання команди керування, оновлює НМ з метою точності прогнозів в подальшому [27].

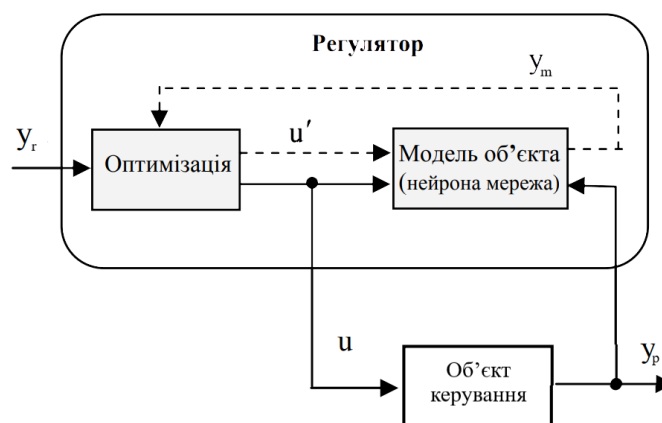


Рис. 2.22. Структурна схема NNPC [22]

Зазвичай такий регулятор використовується для систем, у яких багато входів та виходів, складна структура чи нелінійна поведінка [27].

Тому у цій роботі ми використаємо саме цей контролер, взявши за об'єкт керування реактор з резервуаром безперервної дії, який зображено на рис. 2.23, так як його процес є нелінійним і проявляє стабільний та нестабільний стан в різних областях його роботи, тому керувати ним в повному діапазоні є складним завданням [29].

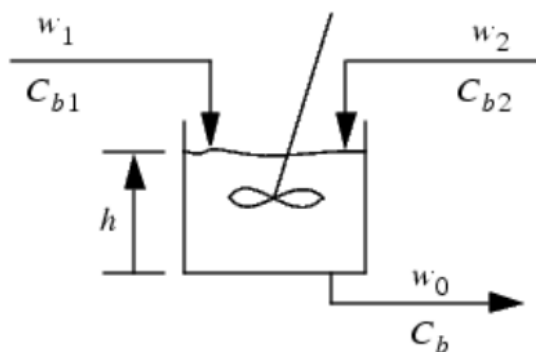


Рис. 2.23. Реактор з резервуаром безперервної дії [29]

Запишемо динамічну модель даної системи [29]:

$$\begin{aligned} \frac{dh(t)}{dt} &= w_1(t) + w_2(t) - 0.2\sqrt{h(t)} \\ \frac{dC_b(t)}{dt} &= (C_{b1}(t) - C_b(t)) \frac{w_1(t)}{h(t)} + (C_{b2}(t) - C_b(t)) \frac{w_2(t)}{h(t)} - \frac{k_1 C_b(t)}{(1 + k_2 C_b(t))^2}, \end{aligned} \quad (2.3)$$

де $h(t)$ – рівень рідини; $C_b(t)$ – концентрація продукту на виході процесу; $w_1(t)$ – витрата концентрованого продукту $C_{b1}(t)$; $w_2(t)$ – витрата концентрованого продукту $C_{b2}(t)$. Вхідні концентрації $C_{b1}(t) = 24.9$ та $C_{b2}(t) = 0.1$; константи швидкості споживання $k_1 = k_2 = 1$.

Завдання регулятора полягає в тому, щоб підтримувати концентрацію продукту, регулюючи витрату $w_1(t)$. Щоб спростити нашу задачу, задаємо $w_2(t) = 0.1$; рівень рідини у резервуарі $h(t)$ при цьому експерименті не контролюється [29].

2.3.2 Навчання НМ регулятора

Реалізуємо цю систему у Simulink. Для цього складаємо схему ОК, яка зображена на рис.2.24.

Наступним кроком є обрання у пакеті *Deep Learning Toolbox* потрібного нам блоку контролера, який має назву Neural Network Predictive Controller, а також задаємо вхідний сигнал через блок Random Reference. Ми будемо подавати на вхід випадковий сигнал у межах [20;23], щоб мати змогу краще провести оцінку та тестування працездатності й ефективності контролера, так як при випадковому сигналі створюються різноманітні вхідні сигнали з різними характеристиками.

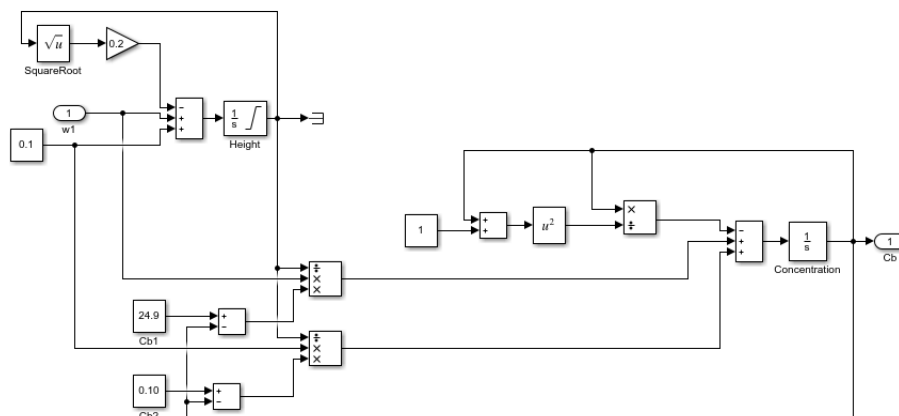


Рис. 2.24. Схема об'єкта керування

Також цей тип сигналу може допомогти виявити можливі проблеми або недоліки в контролері, такі як нестабільність чи погана пристосованість до зміни умов тощо.

Отримана схема зображена на рис. 2.25.

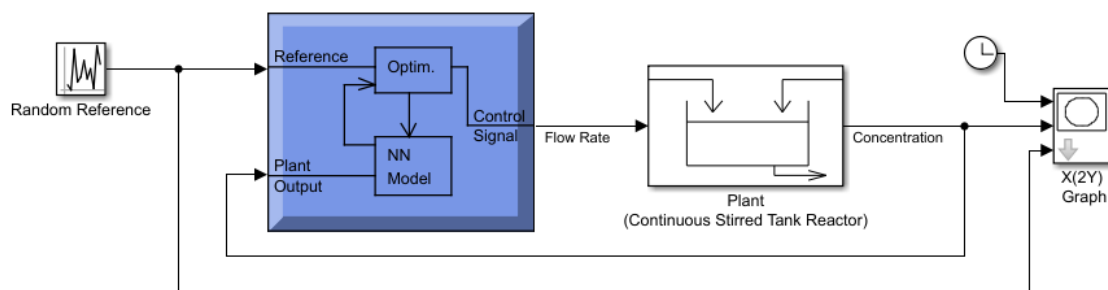


Рис. 2.25. Схема керування з використанням нейромережевого контролера з прогнозуванням

Тепер можемо почати навчання контролера. Для цього натискаємо на нього кілька разів. Після чого бачимо вікно налаштування, яке зображено на рис. 2.26. Так як тренування нейроконтролера вимагає чіткого розуміння методики виконання процесу, а також великого досвіду в цій справі, то ми скористаємося вже підібраними параметрами [31].

Neural Network Predictive Control

Cost Horizon (N2) 7 Control Horizon (Nu) 2	Control Weighting Factor (γ) 0.05 Search Parameter (λ) 0.001
Minimization Routine csrchbac	Iterations Per Sample Time 2
Plant Identification	OK Cancel Apply
Perform plant identification before controller configuration.	

Рис. 2.26. Графічний інтерфейс налаштувань Neural Network Predictive Control

У вікні Neural Network Predictive Control (Нейромережевий прогнозуючий контроль) задаємо наступні параметри [31]:

Горизонт або період витрат (Cost Horizon) = 7.

Горизонт або період керування (Control Horizon) = 2.

Коефіцієнт ваги керування (Control Weighting Factor) = 0.05.

Пошуковий параметр (Search Parameter) = 0.001.

Кількість ітерацій на часовий проміжок (Iterations Per Sample Time) = 2.

І задаємо метод мінімізації.

Далі натискаємо на Plant Identification (Ідентифікація об'єкта). Ми бачимо вікно налаштувань, зображене на рис. 2.27.

The screenshot shows the 'Plant Identification' window with the following settings:

- Network Architecture:**
 - Size of Hidden Layer: 7
 - Sampling Interval (sec): 0.2
 - No. Delayed Plant Inputs: 2
 - No. Delayed Plant Outputs: 2
 - ☐ Normalize Training Data
- Training Data:**
 - Training Samples: 8000
 - Maximum Plant Input: 4
 - Minimum Plant Input: 0
 - Maximum Interval Value (sec): 20
 - Minimum Interval Value (sec): 5
 - Limit Output Data: ☒
 - Maximum Plant Output: 23
 - Minimum Plant Output: 20
 - Simulink Plant Model: cstr
 - Buttons: Generate Training Data, Import Data, Export Data
- Training Parameters:**
 - Training Epochs: 200
 - Training Function: trainlm
 - Use Current Weights: ☒
 - Use Validation Data: ☒
 - Use Testing Data: ☐
 - Buttons: Train Network, OK, Cancel, Apply

A blue text note at the bottom states: 'Generate or import data before training the neural network plant.'

Рис. 2.27. Графічний інтерфейс налаштувань Plant Identification

У вікні Plant Identification (Ідентифікація об'єкта) ми задаємо наступні параметри [31]:

Архітектура мережі (Network Architecture)

Розмір прихованого шару (Size of Hidden Layer) = 7.

Інтервал дискретизації (Simpling Interval) = 0.05.

Вхідні сигнали без затримок на об'єкті (No.Delayed Plant Inputs) = 2.

Вихідні сигнали без затримок на об'єкті (No.Delayed Plant Outputs) = 2.

Дані для тренування(Training Data)

Навчальні вибірки (Training Samples) = 10000.

Максимальний вхідний сигнал (Maximum Plant Inputs) = 15.

Мінімальний вхідний сигнал (Minimum Plant Inputs) = -15.

Максимальне значення інтервалу (Maximum Interval Value) = 2.

Мінімальне значення інтервалу (Minimum Interval Value) = 0,1.

Задаємо обмеження вихідних даних (Limit Output Data):

Максимальний вихідний сигнал (Maximum Plant Inputs) = 3,1.

Мінімальний вихідний сигнал (Minimum Plant Inputs) = -3,1.

Параметри для тренування(Training Parameters)

Епохи тренування (Training Epochs) = 300.

Задаємо функцію тренування НМ, після чого натискаємо Generate Training Data (Генерація навчальних даних).

Тепер чекаємо, коли процес закінчиться (процес може займати значний проміжок часу).

На рис. 2.28 ми можемо побачити приклад результату генерації навчальних даних.

Отримавши результати, ми можемо або прийняти дані, якщо ми вважаємо, що отримані результати задовольняють встановлені критерії та придатні для використання у подальшій роботі, або відхилити дані, якщо дані не задовольняють встановлені вимоги, наприклад через помилки, шум, відхилення від норми тощо.

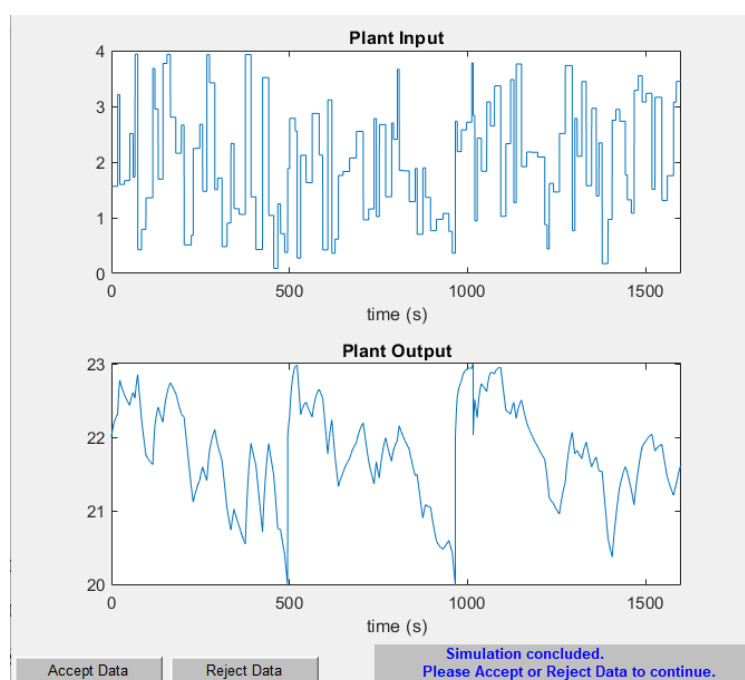


Рис. 2.28. Результат генерації навчальних даних об'єкта

Отримані результати вважаю прийнятними, тому натискаємо Accept Data (прийняти дані). Тепер можна запускати тренування мережі, для цього натискаємо відповідну кнопку Train Network. Процес тренування ми можемо бачити та контролювати у інтерфейсі, який зображено на рис. 2.29.

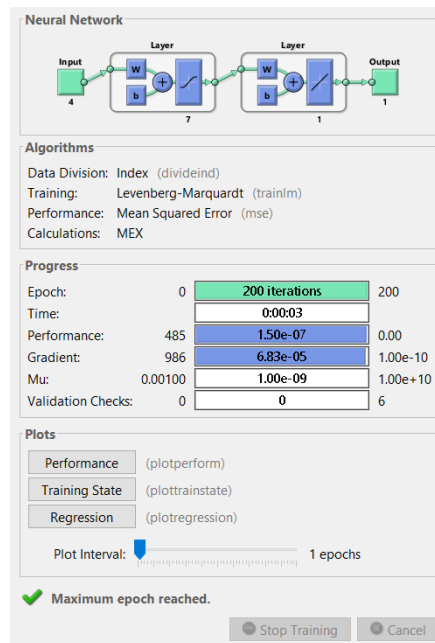


Рис. 2.29. Графічний інтерфейс для тренування нейронної мережі об'єкта

Після завершення процесу ми можемо бачити тренувальні дані (Training data), які зображені на рис. 2.30, та дані валідації (Validation data) які зображені на рис. 2.31.

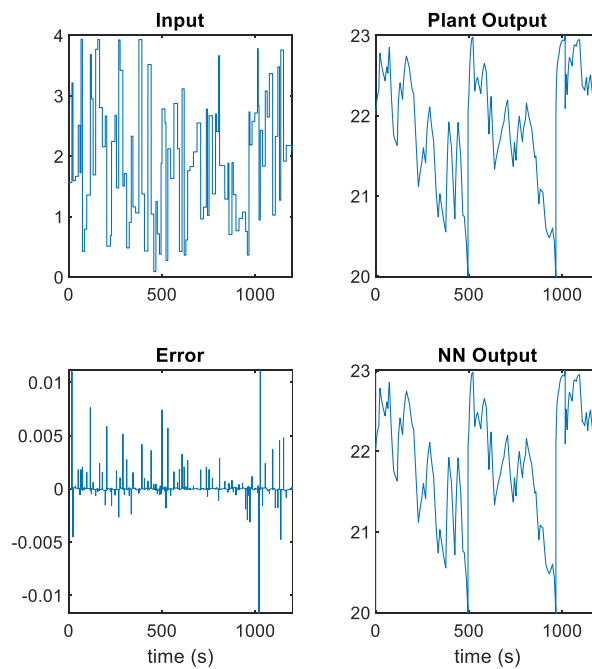


Рис. 2.30. Отримані тренувальні дані

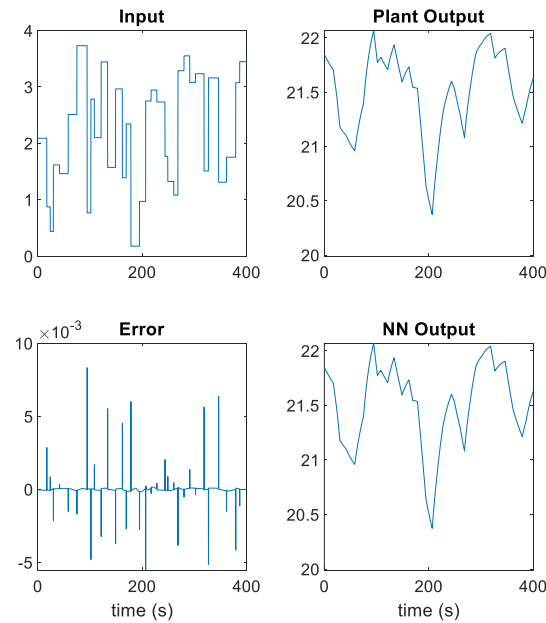


Рис. 2.31. Отримані дані валідації

Отримані результати нас задовольняють, якщо на виході ми маємо однакові результати, або якщо відхилення результатів є прийнятним.

Аналізуючи результати з рис. 2.30 та рис. 2.31, ми бачимо що тренування було проведено правильно, так як на обох рисунках вихідні графіки є ідентичними.

Так як результати є прийнятними, то зберігаємо усі налаштування, чим завершуємо тренування контролера.

2.3.3 Аналіз ефективності НМ регулятора після навчання

Тепер закриваємо усі вікна та повертаємося до нашої схеми. Задаємо час моделювання. Я обрала 300 с. та запускаємо моделювання системи. Чекаємо, поки завершиться процес моделювання. Після чого отримуємо результати, які зображено на рис. 2.32.

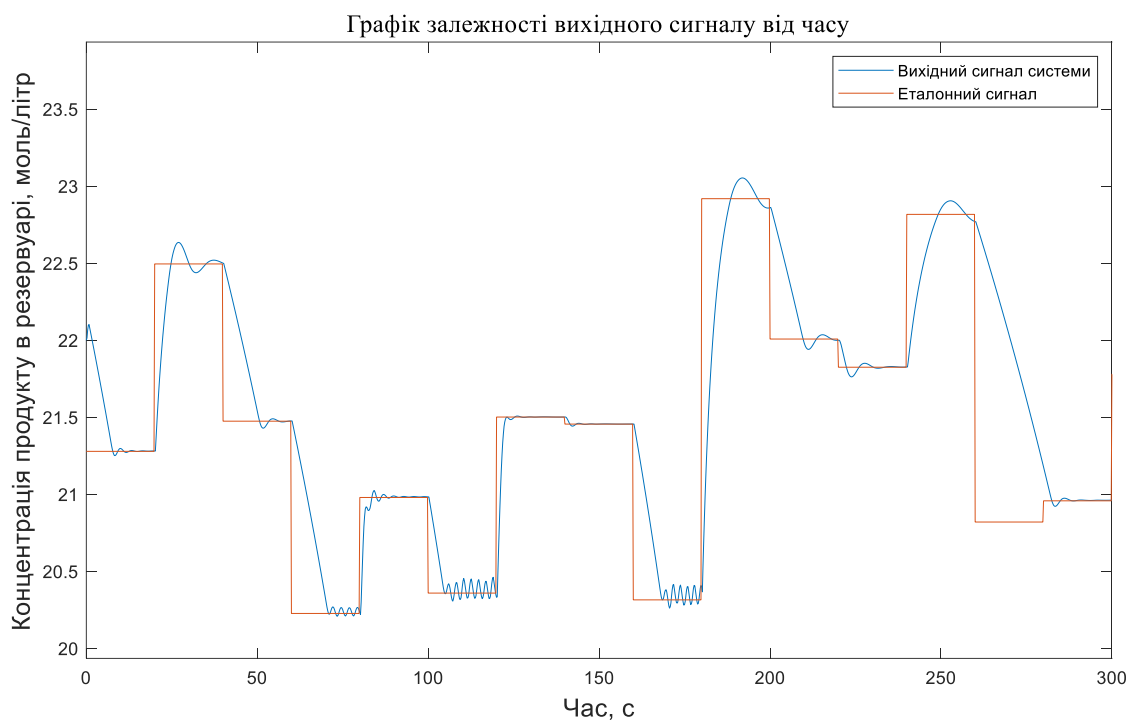


Рис. 2.32. Результати моделювання

Аналізуючи отримані результати з рис. 2.32, можна зробити висновок, що система постійно намагається досягти бажаного значення. Концентрація продукту у резервуарі (синій графік) постійно коливається навколо еталонного значення. Регулятор намагається забезпечити стабільну концентрацію продукту, контролюючи витрату. З часом графік стабілізується навколо еталонного значення і коливання стають меншими. Однак присутня невелика затримка між реакцією регулятора на зміни концентрації і досягнення стабільного стану.

Загалом, можна зробити висновок, що регулятор демонструє здатність підтримувати стабільність концентрації продукту у резервуарі. Хоча є місце деяким коливанням, але вони стають меншими з часом, натомість регулятор продовжує адаптуватися задля забезпечення еталонної концентрації продукту.

Висновки до розділу

У цьому розділі було розглянуто пакет Deep Learning Toolbox, які допомагають синтезувати нейронні мережі та регулятори на їхній основі.

Також розглянули нейромережевий регулятор на основі еталонної моделі, його характеристику та структуру. Для експерименту була обрана система керування одноланковою рукою робота, яка відповідає обраному контролеру. У програмному середовищі Simulink побудували схему та провели навчання регулятора. Отримані результати виявилися такими, на які ми і розраховували. Контролер намагається постійно наслідувати поведінку еталонної моделі.

Крім цього, також розглянули нейромережевий контролер з прогнозуванням. Склали коротку характеристику та визначили для яких систем він найбільш є підходящим. Для нашого моделювання як об'єкт керування було обрано систему керування реактором з резервуаром безперервної дії. Склали систему, після чого виконали тренування нейромережевого контролера. У результаті моделювання ми отримали результати, на які розраховували, адже з часом регулятор все краще прогнозує поведінку системи.

Отже, обидва регулятори мають свої переваги та застосування в системах керування. Вибір між ними буде залежати від конкретної задачі, вимог до системи та умов її функціонування. Враховуючи їхні відмінності та потенціал, важливо правильно вибрати регулятор, що найкраще відповідає потребам системи керування мобільним роботом.

ВИСНОВКИ

У даній дипломній роботі було досліджено різні аспекти робототехніки, а також застосування нейромережових регуляторів у системах керування мобільними роботами. Розглянуті були види та класифікація роботів, їх роль у повсякденному житті людини, а також системи руху та керування мобільними роботами.

У рамках дослідження було розглянуто нейромережові регулятори як потенційний інструмент для покращення керування роботами. Для цього було проведено синтез регуляторів на основі еталонної моделі та з прогнозуванням у середовищі Simulink. Обидва регулятори продемонстрували високу ефективність та здатність до стабільного керування, хоча вони працюють за різними принципами.

Наступним кроком було порівняння цих регуляторів, проте порівняння є вкрай складним завданням, оскільки вони мають різні підходи та характеристики. Тому при виборі регулятора для конкретного об'єкта керування необхідно ретельно проаналізувати його вимоги та очікувані результати.

Застосування нейромережових регуляторів у системах керування роботами має великий потенціал для покращення продуктивності, зниження витрат та підвищення рівня автономності. Це відкриває широкі можливості для їх використання в різних сферах, включаючи автоматизацію виробничих процесів, медичну робототехніку та роботи у важкодоступних середовищах.

Отже, на основі отриманих результатів можна зробити висновок, що застосування нейромережових регуляторів у системах керування мобільними роботами має великий потенціал та варто продовжувати їх дослідження та розвиток. Вони можуть стати потужним інструментом для створення ефективних та інтелектуальних робототехнічних систем, які здатні успішно працювати в різних умовах та досягати поставлених цілей.

ПЕРЕЛІК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Нестеренко, О. В. Інтелектуальні системи і технології. Ввідний курс : навч. посіб. / Нестеренко, О. В., Ковтунець, О. В., Фаловський, О.О. — К.: Національна академія управління, 2017. — 90 с.
2. Розетка. "Робот-пилосос Xiaomi Mi Robot Vacuum-Mop 2 Pro Black (BHR5204EU)" [Електронний ресурс].– Режим доступу до ресурсу:
<https://rozetka.com.ua/ua/xiaomi-894720/p333225100/>.– Дата доступу: 22 травня 2023.
3. Igrushki7km. "Робот Bambi Robowisdom 28091 на радіоуправлінні, акумулятор" [Електронний ресурс]. – Режим доступу до ресурсу:
<https://igrushki7km.com.ua/ua/robot-28091-r-u-na-radioupravlenii-67-programmiruemyh-funktsij-ekzoskeletonnaya-sistema-plavnogo-dvigeniya-sistema-infrakrasnyh-i-taktilnyh-datchikov-22133.html>.– Дата доступу: 22 травня 2023.
4. WaterLine. "Робот-пилосос для басейну dolphin s200" [Електронний ресурс].– Режим доступу до ресурсу:
https://wtl.com.ua/ua/p1606723736-robot-pylesoc-dlya.html?source=merchant_center .– Дата доступу: 22 травня 2023.
5. КЕМРРІ. "РОБОТ ДЛЯ ДУГОВОГО ЗВАРЮВАННЯ ARC MATE 120 ID" [Електронний ресурс].– Режим доступу до ресурсу:
https://kemppi.in.ua/catalog/svarochniy_robot/278.htm.– Дата доступу: 22 травня 2023.
6. DigiKey. "Use Compact Industrial Robots to Make Any Shop More Productive" [Електронний ресурс].– Режим доступу до ресурсу:
<https://www.digikey.fi/fi/articles/use-compact-industrial-robots-to-make-any-shop-more-productive>.– Дата доступу: 22 травня 2023.
7. Нанотехнолог – моя майбутня професія. "Нанороботи" [Електронний ресурс].– Режим доступу до ресурсу:
http://kornilov-profession.blogspot.com/2013/05/blog-post_1269.html.– Дата

доступу: 22 травня 2023.

8. SURFACE. "Scanning Probe Techniques" [Електронний ресурс].– Режим доступу до ресурсу:
https://www.surface-tec.com/nanolab_spm.php. – Дата доступу: 22 травня 2023.
9. Reeman. "Комерційний Виставковий зал Гуманоїдний службовий робот" [Електронний ресурс].– Режим доступу до ресурсу:
<http://ua.reemanrobot.com/humanoid-service-robot/the-exhibition-hall-humanoid-service-robot/commercial-the-exhibition-hall-humanoid.html>. – Дата доступу: 23 травня 2023.
10. Вікіпедія. "iCub" [Електронний ресурс].– Режим доступу до ресурсу:
<https://it.wikipedia.org/wiki/ICub>. – Дата доступу: 23 травня 2023.
11. Rubio. F. A review of mobile robots: Concepts, methods, theoretical framework, and applications / F. Rubio, F. Valero, C. Liopis-Albert // International Journal of Advanced Robotic Systems. – 2019. - №2. Р. 1-22.
12. Вікіпедія. "Марсохід" [Електронний ресурс].– Режим доступу до ресурсу:
<https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%81%D0%BE%D1%85%D1%96%D0%B4> . – Дата доступу: 23 травня 2023.
13. Вечірній Київ. "Техніка від іноземних партнерів допомогла київським рятувальникам ліквідувати пожежу у Борисполі" [Електронний ресурс].– Режим доступу до ресурсу:
<https://vechirniy.kyiv.ua/news/71217/> . – Дата доступу: 23 травня 2023.
14. Prom. "Купити Колеса на мотоблок 5.00-12 (6PR)" [Електронний ресурс].– Режим доступу до ресурсу:
<https://prom.ua/ua/p1169455557-kolesa-motoblok-500.html>. – Дата доступу: 24 травня 2023.
15. Tokopedia. "Sepatu Roda Flying Eagle F110X Inline Skates Black - 39" [Електронний ресурс].– Режим доступу до ресурсу:
https://www.tokopedia.com/joinsepaturoda/sepatu-roda-flying-eagle-f110x-inline-skates-black-39?utm_source=google&utm_medium=organic&utm_campaign=pdp-seo. – Дата доступу: 24 травня 2023.

16. PopSci+. "Weird Wheels Move Robot In Four Directions Without Twisting" [Електронний ресурс].– Режим доступу до ресурсу:
<https://www.popsci.com/weird-wheels-move-robot-in-four-directions-without-turning/>. – Дата доступу: 24 травня 2023.
17. Розетка. "Колеса з поворотним кронштейном з майданчиком 1021125РК Фрегат, діаметр 125мм, навантаження 120кг" [Електронний ресурс].– Режим доступу до ресурсу:
<https://rozetka.com.ua/ua/189855436/p189855436/>. – Дата доступу: 24 травня 2023.
18. Герасін О.С. Аналіз особливостей мобільних роботів багатоцільового призначення / О.С. Герасін // Наукові праці. Комп'ютерні технології. – 2014. - №238. – С. 25-31.
19. Збруцький О. В. Мобільні роботи: можливості, перспективи, проблеми / О. В. Збруцький, Ю. М. Савенко, Д. С. Мішкін // Механіка гіроскопічних систем : науково-технічний збірник. – 2013. – Вип. 26. – С. 112–120. – Бібліогр.: 5 назв.
20. Shutterstock. "Generic blue neuron cell model isolated on a white background with copy space" [Електронний ресурс].– Режим доступу до ресурсу:
<https://www.shutterstock.com/image-illustration/generic-blue-neuron-cell-model-isolated-1718854981>. – Дата доступу: 27 травня 2023.
21. Субботін, С. О. Нейронні мережі : теорія та практика : навч. посіб. / Субботін, С. О. – Житомир : Вид. Євенок, О. О. 2020. – 184 с.
22. Синтез нейромережевого регулятора для електромеханічної системи з пружними зв'язками в кінематичних передачах / Т.Ю. Василець, О.О. Варфоломійєв, В.С. Іщенко, С.Л. Ковальчук, О.О. Сусли // Системи обробки інформації. — 2018. — № 2. — С. 7-17.
23. Neural Network Predictive Controller в автоматизації теплового пункту / О.В. Степанець, Д.Т. Гритчук // Молодий вчений. — 2020. — № 4(80). — С. 235-239.
24. Нейромережева система наведення і стабілізації з регулятором на основі еталонної моделі Model Reference Controller / Б.І. Кузнецов, Т.Ю. Василець, О.О. Варфоломійєв // Електротехніка і Електромеханіка. — 2015. — № 4. — С.

35-39.

25. MATLAB. "MATLAB" [Електронний ресурс].– Режим доступу до ресурсу:
<https://www.mathworks.com/products/matlab.html>. – Дата доступу: 9 травня 2023.
26. MATLAB. "Deep Learning Toolbox" [Електронний ресурс].– Режим доступу до ресурсу:
https://www.mathworks.com/help/deeplearning/index.html?s_tid=CRUX_topnav. – Дата доступу: 9 травня 2023.
27. Порівняльний огляд регуляторів на основі нейронних мереж / К.Е. Молла// Збірник праць XVI Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування», 16-17 травня 2023 року, КПІ ім. Ігоря Сікорського, 2023. – С. 19-21.
28. ResearchGate. "Single-link robot arm" [Електронний ресурс].– Режим доступу до ресурсу:
https://www.researchgate.net/figure/single-link-robot-arm-where-ph-is-the-angle-of-the-arm-and-T-is-the-torque-supplied-by-a_fig1_340294825. – Дата доступу: 9 травня 2023.
29. Type-2 Fuzzy Logic Control of Continuous Stirred Tank Reactor / Khyati Sahu, G N Pillai // Advance in Electronic and Electric Engineering. — 2013. — № 2. — С. 169-178.
30. YouTube. "Model Reference Control System" [Електронний ресурс].– Режим доступу до ресурсу:
<https://www.youtube.com/watch?v=gandte90sXk>. – Дата доступу: 9 травня 2023.
31. YouTube. "Model Predictive Control System" [Електронний ресурс].– Режим доступу до ресурсу:
<https://www.youtube.com/watch?v=p-KBS-Neqoo>. – Дата доступу: 9 травня 2023.