

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА АЕРОКОСМІЧНИХ СИСТЕМ УПРАВЛІННЯ

ДОПУСТИТИ ДО
ЗАХИСТУ

Завідувач кафедри

_____ Мельник Ю.В.

« ____ » _____ 2024р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСНИКА ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО СТУПЕНЯ “БАКАЛАВР”

Тема: Адаптивний ПІД - регулятор на основі нейронної мережі

Виконавець: Біленький Максим Вікторович

Керівник: Безкоровайний Юрій Миколайович

Нормоконтролер: Безкоровайний Юрій Миколайович

Київ 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації електроніки та телекомунікацій

Кафедра аерокосмічних систем управління

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Мельник Ю. В.

“ _____ ” _____ 2024р.

ЗАВДАННЯ

На виконання кваліфікаційної роботи

Біленького Максима Вікторовича

1. Тема кваліфікаційної роботи Адаптивний ПІД- регулятор на основі нейронної мережі
затверджена наказом ректора від 01.04.2024 р. № 511/ст
2. Термін виконання роботи: з 13.04.2024 по 16.06.2024
3. Вихідні дані до роботи: дослідження архітектурного вибору для контролера на базі нейронної мережі, та його продуктивність у навчанні коефіцієнтів контролера в заданих умовах.
4. Зміст пояснювальної записки: Аналіз існуючих методів ПІД-регулювання, Дослідження нейронних мереж, Моделювання та симуляція, Дослідження ефективності методу PIDNN, Практичні застосування, Розширення досліджень.
5. Перелік ілюстративного матеріалу презентації: Сигмоїдальні функції, Нейронна мережа прямого поширення, Блок-схема генетичного алгоритму, Блок-схема оптимізації рою частинок, Графіки результатів симуляцій адаптивного ПІД-регулятора, Схеми та графіки експериментальних налаштувань PIDNN

6. Календарний план-графік

№ пор .	Завдання	Термін виконання	Відмітка про виконання
1.	Постановка завдання	13.04.2024	виконано
2.	Визначення мети та основних завдань дослідження	26.04.2024	виконано
3.	Огляд наявних методів	05.05.2024	виконано
4.	Теоретичне обґрунтування вирішення поставлених завдань	10.05.2024	виконано
5.	Моделювання системи та тестування її ефективності в лабораторних умовах	16.05.2024	виконано
6.	Підготовка пояснювальної записки	24.05.2024	виконано
7.	Складання презентації та підготовка ілюстраційних матеріалів	26.05.2024	виконано

7. Дата видачі завдання: 13.04.2024 р.

Керівник кваліфікаційної роботи: _____ Безкоровайний Юрій Миколайович

Завдання прийняв до виконання: _____ Біленький Максим Вікторович

Реферат

Текстова частина роботи: 55 сторінок, 19 рис.

Об'єкт дослідження - Процес автоматичного регулювання параметрів системи з використанням адаптивного ПД-регулятора на основі нейронної мережі.

Предмет дослідження - Розробка та дослідження адаптивного ПД-регулятора, що використовує нейронні мережі для автоматичного налаштування параметрів регулятора.

Мета роботи - Розробити адаптивний ПД-регулятор на основі нейронної мережі, що забезпечує ефективне автоматичне налаштування параметрів для різних умов експлуатації.

Методи дослідження - Аналіз існуючих методів ПД-регулювання для визначення базових принципів роботи та обмежень традиційних ПД-регуляторів. Дослідження нейронних мереж для розробки алгоритму адаптивного налаштування. Моделювання та симуляція для тестування ефективності запропонованого регулятора. Практичні експерименти для підтвердження результатів симуляцій.

Короткий зміст роботи - Теоретичні основи адаптивного ПД-регулятора: розглянуто основи ПД-регулювання, нейронних мереж та адаптивних систем. Оптимізація нейронної мережі та її впровадження в роботу ПД-регулятора: описано структуру адаптивного ПД-регулятора, методи оптимізації, алгоритми генетичного та ройового пошуку, адаптивний ПД-регулятор з використанням нейронних мереж. Дослідження методу PIDNN для налаштування ПД-регуляторів на основі нейронних мереж: розглянуто методи визначення параметрів, впровадження та налаштування ПД-регулятора, експериментальні налаштування та результати.

Ключові слова - АДАПТИВНИЙ ПІД-РЕГУЛЯТОР, НЕЙРОННІ МЕРЕЖІ, АВТОМАТИЧНЕ НАЛАШТУВАННЯ, МОДЕЛЮВАННЯ, СИМУЛЯЦІЯ, ОПТИМІЗАЦІЯ, ГЕНЕТИЧНИЙ АЛГОРИТМ, РОЙОВИЙ ПОШУК.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. Теоретичні основи адаптивного ПІД-регулятора.....	9
1.1. Поняття ПІД-регулятора	9
1.2. Коли можна використовувати ПІД-керування?.....	12
1.3. Нейронні мережі.....	15
1.4. Адаптивні системи регулювання.....	20
РОЗДІЛ 2.....	24
Оптимізація нейронної мережі та її впровадження в роботу ПІД-регулятора	
2.1. Структура адаптивного ПІД-регулятора.....	24
2.2. Оптимізаційні методи штучного інтелекту.....	25
2.3. Адаптивний PID-регулятор з використанням нейронних мереж.....	31
2.4. Впровадження нейронної мережі.....	35
2.5. Регулювання нейронних мереж.....	38
РОЗДІЛ 3.....	42
Дослідження методу PIDNN для налаштування ПІДрегуляторів на основі нейронних мереж.	
3.1. Методи визначення параметрів ПІД-регуляторів.....	42
3.2. Впровадження та налаштування ПІД-регулятора.....	45
3.3. Опис нейронної мережі ПІД-регулятора.....	47
3.4. Експериментальні налаштування та результати.....	50
Висновок.....	57
Список використаних джерел.....	59

						Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Вступ

Сучасний розвиток автоматизованих систем керування все більше уваги приділяє вдосконаленню методів регулювання технологічних процесів. Одним з найбільш поширених методів є пропорційно-інтегрально-диференційний (ПІД) регулятор, який, завдяки своїй простоті та ефективності, широко використовується в різних галузях промисловості. Проте традиційні ПІД-регулятори мають ряд обмежень, зокрема труднощі в налаштуванні параметрів для досягнення оптимальних результатів у динамічно змінних умовах. Це особливо актуально для сучасних систем, які потребують адаптивних методів регулювання для забезпечення стабільної та надійної роботи. У зв'язку з цим, використання нейронних мереж для автоматичного налаштування параметрів ПІД-регулятора є перспективним напрямом досліджень, що дозволяє підвищити ефективність і точність регулювання.

Метою роботи є розробка адаптивного ПІД-регулятора на основі нейронної мережі, що забезпечує автоматичне налаштування параметрів для різних умов експлуатації. Для досягнення цієї мети необхідно вирішити наступні завдання: провести аналіз існуючих методів ПІД-регулювання та їх обмежень; дослідити можливості використання нейронних мереж для адаптивного налаштування параметрів ПІД-регулятора; розробити алгоритм адаптивного ПІД-регулятора на основі нейронної мережі; виконати моделювання та симуляцію розробленої системи для оцінки її ефективності; провести експериментальні дослідження для підтвердження результатів моделювання.

На сьогоднішній день існує багато методів для налаштування параметрів ПІД-регуляторів, включаючи традиційні методи, такі як метод Ціглера-Нікольса, та сучасні підходи, що базуються на використанні штучного інтелекту, зокрема генетичних алгоритмів та оптимізації рою

						Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

частинок. Однак більшість цих методів не забезпечують достатньої гнучкості та адаптивності в умовах змінних параметрів системи.

Використання нейронних мереж для адаптивного регулювання є новітнім напрямком, що дозволяє значно покращити точність і стабільність керування.

						Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АДАПТИВНОГО ПІД-РЕГУЛЯТОРА

1.1 Поняття ПІД-регулятора

ПІД-регулятор (Пропорційно-Інтегрально-Диференціальний регулятор) є одним із найпоширеніших типів автоматичних регуляторів у системах управління. Його широке використання пояснюється простотою реалізації та ефективністю у керуванні різноманітними процесами. ПІД-регулятори використовуються в різних галузях промисловості, від керування технологічними процесами до застосування в автомобільній та авіаційній промисловості. Основні складові ПІД-регулятора: Пропорційна складова (P), Інтегральна складова (I), Диференціальна складова (D).

Розглянемо Кожну складову: Пропорційна складова визначається як добуток коефіцієнта пропорційності K_p і поточної помилки $q(t)$. Вона забезпечує миттєву реакцію регулятора на зміни в системі. Наприклад, якщо помилка зростає, вихід регулятора також збільшується пропорційно цій помилці, що дозволяє швидко реагувати на відхилення. Пропорційна складова є важливою для забезпечення стабільності системи. Якщо коефіцієнт K_p занадто високий, система може стати нестабільною і почати коливатися. З іншого боку, занадто низький коефіцієнт K_p може призвести до недостатньої реакції на зміни, що уповільнює реакцію системи на відхилення. [1].

Інтегральна складова визначається як добуток коефіцієнта інтеграції K_i та інтеграла помилки з часом. Вона допомагає усунути стале відхилення, акумулюючи помилку з часом і коригуючи вихід для досягнення точного значення встановленого параметра. Ця складова важлива для забезпечення

		АКСУ			Пояснювальна записка			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		Біленький М.В.			ТЕОРЕТИЧНІ ОСНОВИ АДАПТИВНОГО ПІД- РЕГУЛЯТОРА	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевірів</i>		Безкоровайний Ю.М.					9	5
<i>Н-контр.</i>		Безкоровайний Ю.М.				№ 151-403-СУ		
<i>Зав.каф.</i>		Мельник Ю.В.						

точності регулювання, особливо в системах, де присутнє постійне відхилення від заданої точки. Інтегральна складова дозволяє регулятору "запам'ятовувати" минулі значення помилок, що дозволяє системі поступово коригувати вихід для досягнення бажаного стану. Це особлив важливо в системах, де точність регулювання є критичною, наприклад, у температурних регуляторах або системах підтримки тиску [2].

Диференціальна складова визначається як добуток коефіцієнта диференціювання K_d та похідної помилки з часом. Вона забезпечує попередження та коригування виходу на основі швидкості зміни помилки, що дозволяє зменшити перерегулювання та покращити стабільність системи. Це особливо важливо для динамічних систем, де швидкість реакції має велике значення. Інтегральна складова дозволяє регулятору "запам'ятовувати" минулі значення помилок, що дозволяє системі поступово коригувати вихід для досягнення бажаного стану. Це особливо важливо в системах, де точність регулювання є критичною, наприклад, у температурних регуляторах або системах підтримки тиску [2].

Диференціальна складова реагує на швидкість зміни помилки, що дозволяє регулятору "передбачати" майбутні зміни та коригувати вихід завчасно. Це особливо корисно в системах з високою динамікою, де швидкість реакції є ключовою для підтримання стабільності, наприклад, в авіаційних системах керування польотом [2].

Принцип роботи ПД-регулятора:

ПД-регулятор обчислює помилку як різницю між заданим значенням і фактичним значенням системи. Далі, використовуючи пропорційну, інтегральну та диференціальну складові, регулятор генерує керуючий сигнал, який мінімізує цю помилку. Формула ПД-регулятора виглядає наступним чином:

Форм. 1. Принцип роботи ПД-регулятора

						Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

$u(t)$ – керуючий сигнал, $q(t)$ – помилка, K_p – коефіцієнт пропорційності, K_i – коефіцієнт інтеграції, K_d – коефіцієнт диференціювання.

Застосування ПІД-регулятора:

ПІД-регулятори знаходять застосування в різних галузях, таких як: Промисловість: для керування технологічними процесами, такими як температура, тиск, швидкість та рівень рідини [1].

ПІД-регулятори використовуються для підтримання стабільності та точності процесів, що є критичними для якості продукції та ефективності виробництва.

Автомобільна галузь: у системах круїз-контролю, стабілізації та гальмування. У системах круїз-контролю ПІД-регулятори допомагають підтримувати постійну швидкість автомобіля, автоматично коригуючи дросельну заслінку відповідно до зміни швидкості транспортного засобу [2].

Авіація: для керування польотом та стабілізації літаків. В авіаційних системах ПІД-регулятори використовуються для стабілізації польоту та керування автопілотом, що забезпечує безпечність та ефективність польотів.

Робототехніка: для забезпечення точного руху та позиціонування роботів. У робототехніці ПІД-регулятори допомагають досягати високої точності та стабільності при виконанні складних рухів та маніпуляцій [1].

						Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

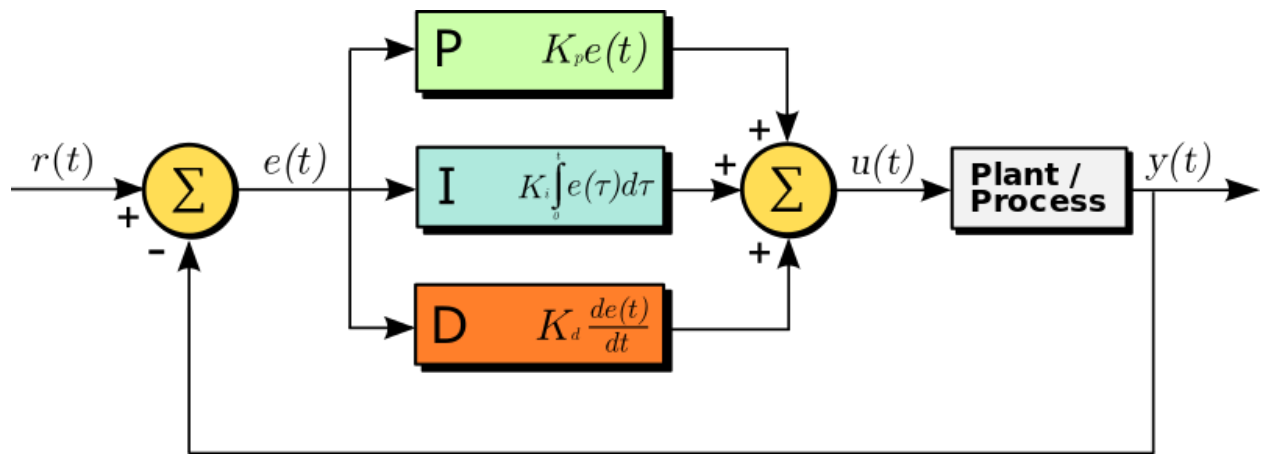


Рисунок 1. Блок-схема ПІД-регулятора

ПІД-регулятор має певні недоліки, такі як складність налаштування оптимальних значень коефіцієнтів та чутливість до змін параметрів системи. Налаштування ПІД-регулятора вимагає точного підбору коефіцієнтів K_p , K_i та K_d для досягнення оптимальної продуктивності. Невірне налаштування може призвести до нестабільності або недостатньої реакції системи [2].

Вибір та налаштування ПІД-регулятора:

Процес налаштування ПІД-регулятора включає визначення оптимальних значень коефіцієнтів K_p , K_i та K_d . Існують різні методи налаштування, такі як метод проб і помилок, метод Ціглера-Нікольсона та метод програмного моделювання. Кожен метод має свої переваги та недоліки, і вибір методу залежить від конкретних вимог та умов системи [1].

1.2. Коли можна використовувати ПІД-керування?

Вимоги до системи керування можуть включати багато факторів, таких як реакція на командні сигнали, нечутливість до шуму вимірювань та змін у процесі, а також відхилення збурень навантаження. Проектування системи керування також враховує аспекти динаміки процесу, насичення виконавчих механізмів та характеристики збурень. Може здатися дивним, що такий простий регулятор, як ПІД-регулятор, може працювати так добре.

Загальне емпіричне спостереження полягає в тому, що більшість промислових процесів можна керувати досить добре за допомогою

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

ПД-регулятора, за умови, що вимоги до продуктивності керування не є надто високими.

Коли П-керування є достатнім?

Диференційна дія часто не використовується. Це цікаве спостереження, що багато промислових регуляторів мають тільки П-дію, а в інших диференційна дія може бути (і часто вимикається).

Можна показати, що П-керування є адекватним для всіх процесів, де динаміка в основному є першого порядку (керування рівнем у одиночних баках, змішувальних реакторах з ідеальним змішуванням тощо). Досить легко з'ясувати, чи це так, шляхом вимірювання відгуку на ступінчастий сигнал або частотного відгуку процесу. Якщо відгук на ступінчастий сигнал виглядає як відгук системи першого порядку, або, точніше, якщо крива Найквіста лежить тільки в першому та четвертому квадрантах, то П-керування є достатнім. Інша причина полягає в тому, що процес був спроектований так, щоб його робота не вимагала жорсткого керування. Тоді, навіть якщо процес має динаміку вищого порядку, йому потрібна інтегральна дія для забезпечення нульового постійного відхилення і адекватного перехідного процесу за рахунок пропорційної дії.

Коли ПД-керування є достатнім?

Аналогічно, ПД-керування є достатнім для процесів, де домінуюча динаміка є другого порядку. Для таких процесів немає переваг у використанні більш складного регулятора.

Типовий випадок, коли диференційна дія покращує відгук, це коли динаміка характеризується часовими константами, що різняться за величиною. Тоді диференційна дія може бути використана для прискорення відгуку. Регулювання температури є типовим прикладом. Диференційне керування також корисне, коли потрібне точне керування системою вищого порядку. Динаміка вищого порядку обмежувала б кількість пропорційного підсилення для гарного керування. За допомогою диференційної дії

						Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечується покращене демпфування, тому можна використовувати більше пропорційне підсилення для прискорення перехідного процесу.

Коли потрібне більш складне керування?

Переваги використання більш складного регулятора, ніж ПІД-регулятор, демонструються на деяких прикладах нижче.

Процеси вищого порядку

Коли система має порядок вище другого, керування можна покращити, використовуючи більш складний регулятор, ніж ПІД-регулятор.

Системи з великим часом затримки

Керування системами з домінуючою затримкою часу є надзвичайно складним завданням. Це також тема, щодо якої існує багато різних думок про доцільність використання ПІД-регулятора. Здається, існує загальна згода, що диференційна дія не дуже допомагає при керуванні процесами з домінуючими затримками часу. Для стабільних процесів з відкритим контуром реакцію на командні сигнали можна суттєво покращити шляхом введення компенсації часу затримки. Відхилення збурень навантаження також можна покращити до певного ступеня, оскільки компенсатор часу затримки дозволяє використовувати більший коефіцієнт підсилення контуру, ніж ПІД-регулятор. Таким чином, системи з домінуючими затримками часу є кандидатами для більш складного керування.

Системи з коливальними режимами

Системи з коливальними режимами, які виникають через інерції та гнучкості, є ще одним випадком, коли ПІД-керування недостатньо. Існує кілька підходів до систем такого типу. У так званому підході з фільтром відсічки (notch filter) не робиться спроб загасити коливальні режими, але докладаються зусилля для зменшення передачі сигналу через регулятор за допомогою фільтра, який різко зменшує передачу сигналу на резонансній частоті. ПІД-регулятор може бути використаний, коли є лише один домінуючий коливальний режим. Дія фільтра відсічки може бути досягнута

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

шляхом обережного вибору параметрів регулятора. У цьому випадку параметри T_i і T_d повинні бути обрані таким чином, щоб чисельник мав комплексні корені.

Отже, Коли динаміка процесу проста, ПІД-регулятор є достатнім. Однак, зі збільшенням складності динаміки продуктивність керування можна покращити за допомогою складніших регуляторів. Наприклад, у системах з великими варіаціями параметрів можливо використовувати лінійні регулятори високого порядку для забезпечення роботи в широкому діапазоні параметрів.

Керування змінними, що впливають на якість процесу, часто вимагає вибору регулятора з урахуванням характеристик збурень, що може вимагати нетрадиційних стратегій, не типових для ПІД. Це особливо важливо для систем із затримками у часі.

ПІД-регулятори, через свою обмежену складність, не можуть ефективно моделювати складну або періодичну поведінку збурень.

1.3. Нейронні мережі

Штучні нейронні мережі є новим трендом, який дозволяє комп'ютеру мислити як людина, розпізнавати складні заплутані шаблони та дозволяє комп'ютерам імітувати людей. Нейронні мережі стали потужною технікою розпізнавання образів. Оскільки потреба в розпізнаванні образів виникає щоразу, коли комп'ютери взаємодіють із реальним світом, нейронні мережі широко корисні в ряді застосувань. Їхня оцінка для використання в конкретній області застосування повинна ґрунтуватися на знанні їхніх сильних і слабких сторін, типів завдань, у яких вони, ймовірно, досягнуть успіху, та конкретних завдань розробки. Нейронна мережа нагадує людський мозок у двох аспектах. Перший - це те, що знання набуваються мережею через процес навчання. Другий - це те, що сили зв'язків між нейронами, відомі як синаптичні ваги, використовуються для зберігання знань. Таким чином, ця технологія стала відома як система штучних нейронних мереж (Artificial Neural Network System, ANS).

						Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Система штучних нейронних мереж (ANS), яка виявилася корисною для вирішення проблем, що вимагають розпізнавання складних шаблонів і виконання нетривіальних функцій відображення, - це мережа зворотного поширення помилки (Back-Propagation Network). (Рис. 2).

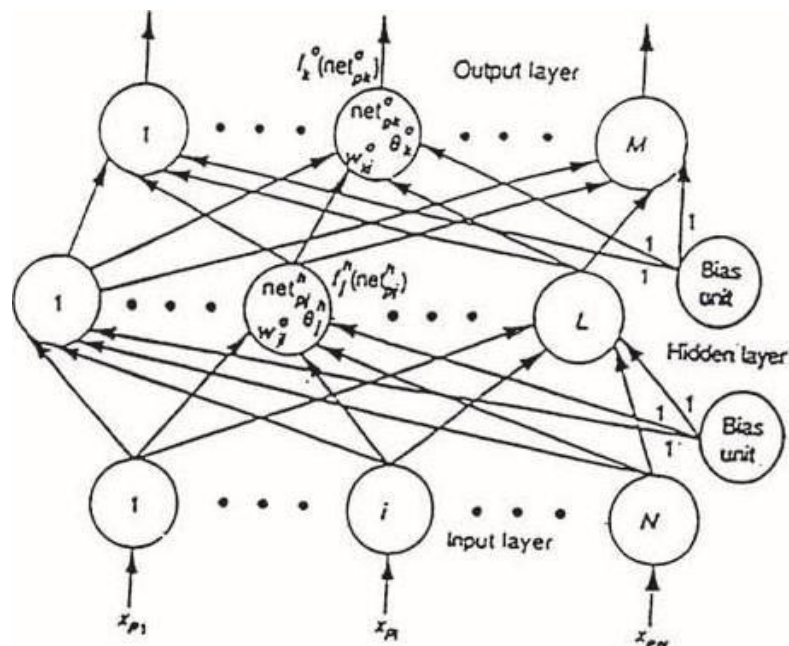


Рис. 2. Архітектура трьохшарової мережі зворотного поширення помилки (BPN)

Мережа зворотного поширення помилки (BPN) є багатошаровою, прямопроточною мережею, яка повністю взаємопов'язана шарами; тобто, немає зворотних зв'язків і зв'язків, що обходять один шар, переходячи безпосередньо до іншого шару. Багато потенційних комп'ютерних застосувань важко реалізувати через те, що багато проблем не підходять для вирішення послідовним процесом. Спочатку мережа навчається на попередньо визначеному наборі пар вхід-вихід, використовуючи двофазний цикл розповсюдження-оновлення. Після того як вхідний шаблон був поданий як стимул до першого шару мережевих одиниць, він розповсюджується через кожен верхній шар, поки не буде згенеровано вихід. Цей вихідний шаблон потім порівнюється з бажаним виходом. Сигнали помилок потім передаються назад від вихідного шару до кожного

вузла в проміжному шарі, який безпосередньо сприяє виходу. Однак кожна одиниця в проміжному шарі отримує лише частину загальної помилки, приблизно на основі відносного внеску, який одиниця зробила у початковий вихід. Цей процес повторюється, шар за шаром, поки кожен вузол у мережі не отримає сигнал помилки, що описує його відносний внесок у загальну помилку. На основі отриманого сигналу помилки ваги з'єднань оновлюються кожною одиницею, що змушує мережу наближатися до стану, який дозволяє закодувати тренувальний шаблон. Загальний ефект такої поведінки полягає в тому, що BPN забезпечує ефективний засіб для комп'ютерної системи аналізувати шаблони даних, які можуть бути неповними або шумними, і розпізнавати тонкі шаблони з часткових вхідних даних. Декілька дослідників показали, що під час тренування BPN розвиває внутрішні відносини між вузлами, щоб організувати тренувальні дані у класи шаблонів. Цю тенденцію можна екстраполювати на гіпотезу, що всі одиниці прихованого шару в BPN якимось чином пов'язані з певними ознаками вхідного шаблону в результаті тренування. Точно, що це за асоціація, може бути чи не бути очевидним для людського спостерігача. Важливо те, що мережа знайшла внутрішнє представлення, яке дозволяє їй генерувати бажані виходи при поданих тренувальних входах. Це ж внутрішнє представлення може бути застосоване до вхідних даних, які не використовувалися під час тренування. BPN буде класифікувати ці раніше невідомі вхідні дані відповідно до ознак, які вони мають спільно з тренувальними зразками. Одиниці у вхідному шарі цієї мережі не виконують перетворення даних на вхідний шаблон мережі. Вони просто утримують компоненти вектору входу у структурі мережі. Таким чином, процес тренування починається, коли зовні поданий вхідний шаблон подається на вхідний шар одиниць.[3]

Рівняння BPN можна представити в порядку, в якому вони використовувалися б під час тренування для однієї пари тренувальних векторів.

						Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- 1- Застосувати вхідний вектор, $= (\quad_1, \quad_2 \dots \dots \dots)$
- 2- Обчислити значення чистого вхідного сигналу для одиниці прихованого шару.

- 3- Обчислити вихід із прихованого шару:

- 4- Перейти до вихідного шару. Обчислити значення чистого вхідного сигналу для кожної одиниці.

- 5- Обчислити вихід:

- 6- Обчислити помилку для вихідних одиниць:

- 7- Обчислити помилки для прихованих одиниць:

- 8- Оновити ваги на вихідному шарі:

- 9- Оновити ваги на прихованому шарі:

Порядок оновлення ваг на окремих шарах не є важливим.

Помилка визначається як:

—

						Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки ця величина є мірою того, наскільки добре мережа навчається, коли помилка стає прийнятно малою для кожної пари тренувальних векторів, навчання можна припинити.

Використовується стільки даних, скільки маємо для навчання мережі, хоча не обов'язково використовувати їх усі. З доступних навчальних даних часто достатньо невеликої підмножини, щоб успішно навчити мережу. Решта даних можна використовувати для тестування мережі, щоб перевірити, чи може мережа виконувати бажане відображення на вхідних векторах, які вона ніколи не зустрічала під час навчання.

Переваги та недоліки BPN (мережі зворотного поширення помилки)

Мережа зворотного поширення помилки (BPN) добре підходить для узагальнення, що означає, що при поданні кількох різних вхідних векторів, які належать до одного класу, BPN навчиться виділяти значні подібності у вхідних векторах. Нерелевантні дані будуть ігноруватися. Якщо BPN неадекватно або недостатньо тренувана на конкретному класі вхідних векторів, подальше розпізнавання членів цього класу може бути ненадійним. Потрібно Переконатися, що тренувальні дані охоплюють весь очікуваний простір вхідних даних. У будь-якому випадку, не має резону тренувати мережу повністю на вхідних векторах одного класу, а потім переходити на інший клас. В значній мірі, її універсальність обумовлена загальною природою процесу навчання мережі. Узагальнення, яке забезпечується цим загальним процесом, дозволяє організацію та з'єднання окремих одиниць всередині мережі, що можуть значно відрізнятися. Аналогічно, завдяки різноманітності структур мереж, які можуть бути створені та успішно навчені за допомогою алгоритмів BPN, ця техніка навчання мереж може бути застосована до багатьох різних видів проблем.[4]

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

1.4. Адаптивні системи регулювання

Адаптивна система керування була розроблена для уникнення деградації динамічної продуктивності системи керування при виникненні змін у навколишньому середовищі. У той час як система зворотного зв'язку орієнтована на усунення впливу збурень стану, адаптивна система керування орієнтована на усунення впливу структурних збурень на продуктивність системи керування. Ці структурні збурення в основному спричинені зміною динамічних параметрів керованої установки.

Потреба в більш складних та високопродуктивних системах керування стала імпульсом для розвитку систематичної теорії керування. Навіть з розвитком такої систематичної теорії керування, зазвичай чогось не вистачає, коли необхідно спроектувати практичну систему керування.

Для реалізації високопродуктивних систем керування, коли динамічні характеристики установки мало відомі або коли відбуваються великі та непередбачувані зміни, слід розглянути клас систем керування, що називаються адаптивними системами керування. На рисунку 3 зображена блок-схема адаптивної системи керування [5].

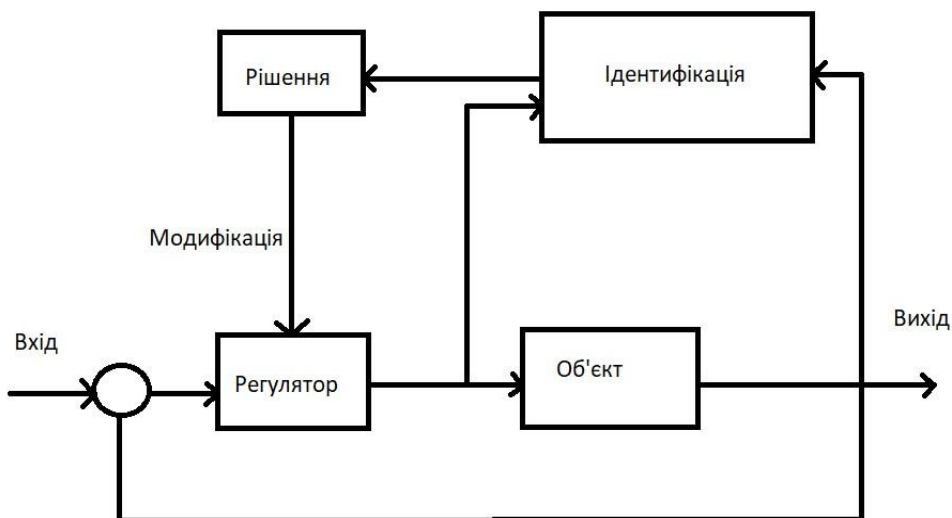


Рис. 3. Блок-схема адаптивної системи керування.

Адаптивне керування з використанням нейронних мереж

					Арк.
					20
Змн.	Арк.	№ докум.	Підпис	Дата	

Адаптивна система керування може складатися з наступних трьох функцій:

- Ідентифікація динамічних характеристик об'єкта.
- Прийняття рішень на основі ідентифікації об'єкта.
- Модифікація або активація на основі прийнятого рішення.

Традиційне адаптивне моделювання та керування поділяється на дві основні категорії: моделювання на основі перших принципів та статистичне моделювання [6].

Багато процесів мають характеристики, які ускладнюють існуючі методики моделювання та керування через:

1. Нелінійність об'єкта або невизначеність у моделі перших принципів.
2. Високівимірні багатовимірні сенсорні входи та керуючі змінні, які можуть демонструвати складні або шкідливі взаємодії.
3. Властиві змінні затримки у часі.
4. Недостатні та невизначені вимірювання.
5. Обмеження щодо безпеки об'єкта та навколишнього середовища.
6. Невимірювані або неконтрольовані зміни зовнішніх змінних, таких як сировина, зовнішня температура, барометричний тиск або вологість.

Концептуально, моделювання на основі перших принципів є найбільш привабливим через його основу в прикладній хімії та фізиці. Однак, навіть при використанні широких спрощувальних припущень для спрощення математичних розрахунків, цей підхід може вимагати багато років роботи для розробки адекватної моделі. Крім того, обмежена точність таких моделей означає, що вони можуть погано працювати на практиці через динаміку реальних об'єктів. З іншого боку, статистичне моделювання вимагає спеціалізованих знань як статистичних методів, так і методик для

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

проектування експериментів. Ці експерименти, зазвичай необхідні для збору даних для статистичної моделі, є дорогими і трудомісткими. Набагато кращим підходом може бути використання існуючих історичних даних для отримання уявлення про динаміку процесу до проведення спроектованих експериментів. Дійсно, багато з цих експериментів можуть бути непотрібними.

У певному сенсі біологічна нейронна мережа створює внутрішню модель функцій тіла і вчиться керувати тілом на основі цієї моделі. Подібна структура може бути корисною для управління виробничим заводом. Штучні нейронні мережі представляють собою набір дуже потужних математичних технік для моделювання, керування та оптимізації, які "вивчають" динаміку процесів безпосередньо з історичних даних. Основні характеристики, які роблять нейронні мережі ідеальною технологією для використання історичних даних, можна підсумувати наступним чином:

Нейронні мережі є алгоритмами нелінійної регресії, які можуть моделювати високовимірні системи та мають дуже простий, уніфікований інтерфейс користувача; вони добре працюють як для пакетних, так і для безперервних процесів, і можуть використовуватися як для статичного, так і для динамічного моделювання. Процес навчання нейронної мережі для прогнозування, оптимізації та керування процесами можна розділити на два етапи: побудова моделі та оптимізаційне керування. Перший етап еквівалентний класичній задачі прогнозування (або передбачення). Як простий приклад, припустимо, що завдання полягає в передбаченні майбутньої поведінки теплової установки на основі історичних даних змінних процесу. Щоб перевірити, наскільки добре працює схема прогнозування, використовується лише частина даних для навчання мережі.

Коли мережа завершує навчання на цьому наборі даних, її перевіряють на решті даних або "тестових даних", щоб визначити, наскільки добре вона може працювати з даними, яких вона раніше не бачила. Якщо вона точно передбачає тестові дані, вважається, що мережа

						Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

узагальнює. Якщо рівень узагальнення високий, то впевненість у тому, що прогнозуюча мережа захопила динаміку установки, також висока.

						Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Розділ 2 ОПТИМІЗАЦІЯ НЕЙРОНОЇ МЕРЕЖІ ТА ЇЇ ВПРОВАДЖЕННЯ В РОБОТУ ПІД-РЕГУЛЯТОРА

2.1. Структура адаптивного PID-регулятора

Більшість систем керування процесами прагнуть контролювати кілька основних параметрів.

До них відносяться час підйому (час, необхідний для того, щоб керовані параметри перейшли від 10 до 90% від кінцевих бажаних значень), час встановлення (час, необхідний для того, щоб загасаючі коливання перехідного процесу досягли та залишилися в межах $\pm 2\%$ від значення усталеної роботи) і максимальне перерегулювання (максимальна величина, на яку керовані змінні перевищують бажані значення). Сигнал керування PID-регулятора задається наступним рівнянням [7]:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

Де $u(t)$ — це керуючий сигнал, а $e(t)$ — це похибка, яка є різницею між бажаною уставкою та виміряною змінною процесу. Параметри регулятора включають пропорційне підсилення (K_p), інтегральний час (T_i), та диференціальний час (T_d) [7,8]

Під час розробки PID-регулятора значення інтегральної складової $K_i = K_p / T_i$ встановлюється для досягнення бажаної похибки в усталеному стані. Як показано на Рисунку 4, PID-регулятор має три основні терміни: пропорційну дію, при якій сигнал керування пропорційний до сигналу похибки; інтегральну дію, при якій сигнал керування пропорційний до часової інтегральної похибки; та диференційну дію $K_d = K_p T_d$, при якій сигнал керування пропорційний до часової похідної похибки. Значення трьох параметрів (K_p , T_i , T_d) повинні бути налаштовані так, щоб вхід

АКСУ					Пояснювальна записка							
Змн.	Арк.	№ докум.	Підпис	Дата	ОПТИМІЗАЦІЯ НЕЙРОНОЇ МЕРЕЖІ ТА ЇЇ ВПРОВАДЖЕННЯ В РОБОТУ ПІД-РЕГУЛЯТОРА			Літ.	Арк.	Акрушів		
Розробив		Біленький М.В.								24	5	
Перевірів		Безкоровайний Ю.М.						№ 151-403-СУ				
Н-контр.		Безкоровайний Ю.М.										
Зав.каф.		Мельник Ю.В.										

керування забезпечував прийнятну продуктивність.

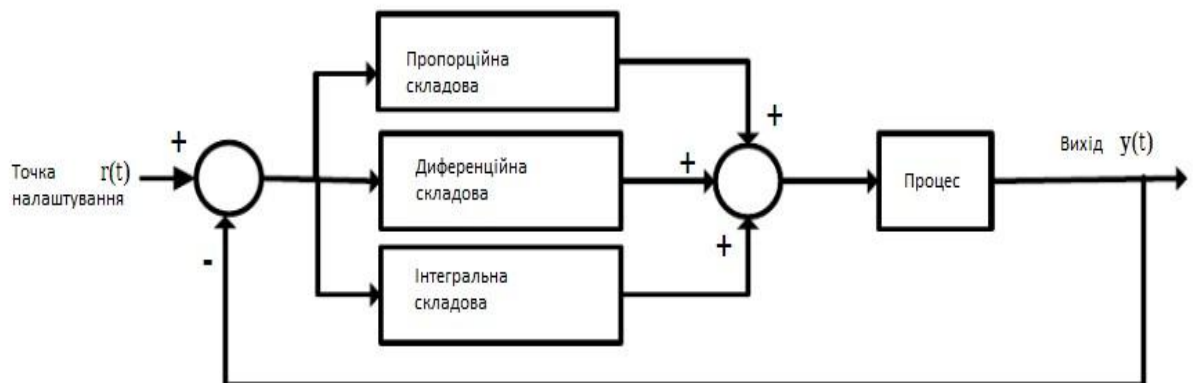


Рисунок 4: Система керування з ПІД-регулятором у замкнутому контурі

Налаштування ПІД-регулятора передбачає вибір параметрів , , які забезпечують задовільну роботу системи із замкнутим контуром. Ці параметри повинні бути обрані таким чином, щоб були дотримані характеристики: швидкість відгуку, час встановлення та відповідний коефіцієнт перерегулювання, що гарантує стабільність та продуктивність системи. Основний метод для цього базується на спробах і помилках, що є трудомістким процесом. Існують різні процеси для різних комбінацій пропорційного, інтегрального та диференційного коефіцієнтів. Завдання інженерії керування полягає в налаштуванні коефіцієнтів для одночасного досягнення зменшення похибки та динамічних відгуків. Передавальна функція ПІД-регулятора визначається наступним чином у рівнянні (2.2) [9]:

2.2. Оптимізаційні методи штучного інтелекту

Генетичні алгоритми (ГА)

Генетичні алгоритми (ГА) є стохастичними методами глобального пошуку, які імітують процес природної еволюції, створюючи сімейство обчислювальних моделей, натхненних еволюцією. Генетичний алгоритм починає роботу без знання правильної відповіді і повністю залежить від

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

відповідей свого середовища та операторів еволюції (тобто відтворення, кросовер і мутація), щоб досягти найкращого рішення. Починаючи з кількох незалежних точок і здійснюючи паралельний пошук, алгоритм уникає локальних мінімумів і збігається до субоптимальних рішень. Таким чином, ГА здатні знаходити високопродуктивні області в складних доменах без труднощів, пов'язаних з високою розмірністю, як це може статися з методами градієнтного спуску або методами, що залежать від інформації про похідну [10,11].

Рисунок 5 показує кроки створення та реалізації генетичного алгоритму. Послідовність можна проілюструвати наступними кроками:

Крок 1: Ініціалізувати популяцію індивідуумів, де кожен індивідуум представляє потенційне рішення для розв'язуваної проблеми.

Крок 2: Застосувати функцію придатності для оцінки якості кожного рішення.

Крок 3: Процес відбору застосовується в ітераціях для формування нової популяції. Процес відбору упереджений до більш придатних індивідуумів, щоб гарантувати їх участь у новій популяції.

Крок 4: Індивідууми змінюються за допомогою еволюційних операторів. Два найчастіше використовуваних еволюційних оператори — це мутація та кросовер:

- Мутація: Мутація вводить різноманітність у популяцію, вводячи нові гени в генетичний пул. Під час мутації індивідуальні агенти зазнають невеликих випадкових змін, що призводить до появи нових індивідуумів. Це допомагає зменшити можливість застрягання агентів у локальних оптимумах.
- Кросовер (або рекомбінація): Цей процес аналогічний спарюванню. Під час кросоверу два індивідууми об'єднуються для створення нащадка. Основна мета кросоверу полягає у вивченні нових областей у просторі пошуку.

						Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Крок 5: Вищезгадані кроки повторюються, доки рій не зійдеться до оптимального або субоптимального рішення [12].

Переваги генетичного алгоритму [13]:

- Оптимізація з безперервними або дискретними змінними.
- Інформація про похідні не потрібна.
- Одночасний пошук у широкому діапазоні поверхні витрат.
- Робота з великою кількістю змінних.
- Підходить для паралельних обчислень.
- Оптимізація змінних з надзвичайно складними поверхнями витрат (вони можуть виходити з локального мінімуму).
 - Забезпечення списку оптимальних змінних, а не лише одного рішення.
 - Кодування змінних так, щоб оптимізація здійснювалася з кодуванням змінних.
 - Робота з числовими даними, експериментальними даними або аналітичними функціями.

Ці переваги захоплюють і забезпечують кращі результати, коли традиційні підходи до оптимізації зазнають невдачі.

						Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

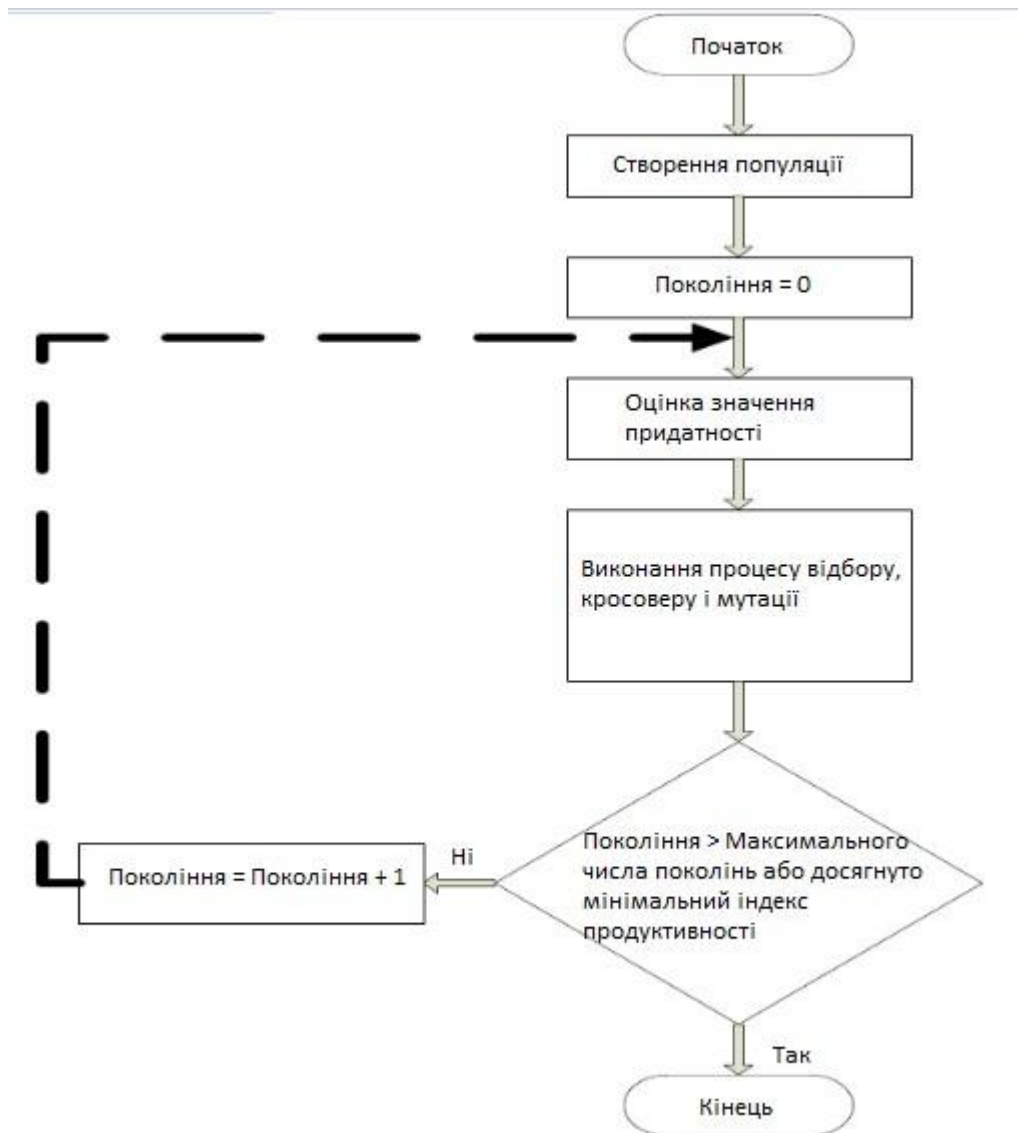


Рисунок 5: Блок-схема генетичного алгоритму

Оптимізація рою частинок (PSO)

Оптимізація рою частинок (PSO) була натхненна соціальною поведінкою зграї птахів. Комп'ютерні алгоритми імітують складну поведінку зграї птахів [12]. У системі PSO рій індивідуумів (які називаються частинками або інтелектуальними агентами) літає в пошуковому просторі. Кожна частинка представляє кандидатське рішення задачі оптимізації. Положення частинки залежить від найкращого положення, відвіданого нею самою (тобто її власного досвіду), і положення найкращої частинки у всій популяції [14]. Найкраще досягнуте положення називається глобальною найкращою частинкою. Продуктивність кожної

частинки вимірюється за допомогою функції придатності, яка змінюється залежно від задачі оптимізації. Зміна положення частинок реалізується за допомогою інформації про положення та швидкість відповідно до рівнянь (2.3) та (2.4) [15, 16]:

Де:

v_k : поточна швидкість частинки i на ітерації k ,

v_k^{+} : нова швидкість частинки i на наступній ітерації $k + 1$

c_1 : коефіцієнт когнітивного прискорення (впевненість у собі),

c_2 : коефіцієнт соціального прискорення (впевненість у роєві)

p_{best} : поточне положення частинки i на ітерації k

p_{best}^{+1} : нове положення частинки i на наступній ітерації $k + 1$

p_{best} : особистий найкращий результат частинки i ,

g_{best} : глобальний найкращий результат популяції.

Оптимізація рою частинок (PSO)

Рисунок 6 ілюструє загальну блок-схему техніки PSO. Послідовність кроків можна описати наступним чином [17]:

Крок 1: Генерація початкових умов для кожної частинки.

Початкові точки пошуку (x_i) і швидкості (v_i) кожної частинки зазвичай генеруються випадковим чином у межах допустимого діапазону.

Поточна точка пошуку встановлюється як p_{best} для кожної частинки.

Найкраще оцінене значення p_{best} встановлюється як g_{best} , і номер частинки з найкращим значенням зберігається.

Крок 2: Оцінка точки пошуку кожної частинки. Обчислюється цільова функція для кожної частинки. Якщо значення краще за поточне значення p_{best} частинки, тоді p_{best} замінюється поточним значенням.

Якщо найкраще значення p_{best} краще за поточне g_{best} , значення g_{best}

						Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

замінюється найкращим значенням, і номер частинки з найкращим значенням зберігається.

Крок 3: Модифікація кожної точки пошуку.

Відмінності між PSO та генетичним алгоритмом (GA):

- PSO загалом швидший, більш надійний і працює краще, ніж GA, особливо коли розмірність проблеми збільшується.
- Продуктивність PSO менш чутлива до розміру популяції (хоча розмір популяції не повинен бути занадто малим). PSO з меншими розмірами роїв працює порівняно краще, ніж GA з більшими популяціями [12].

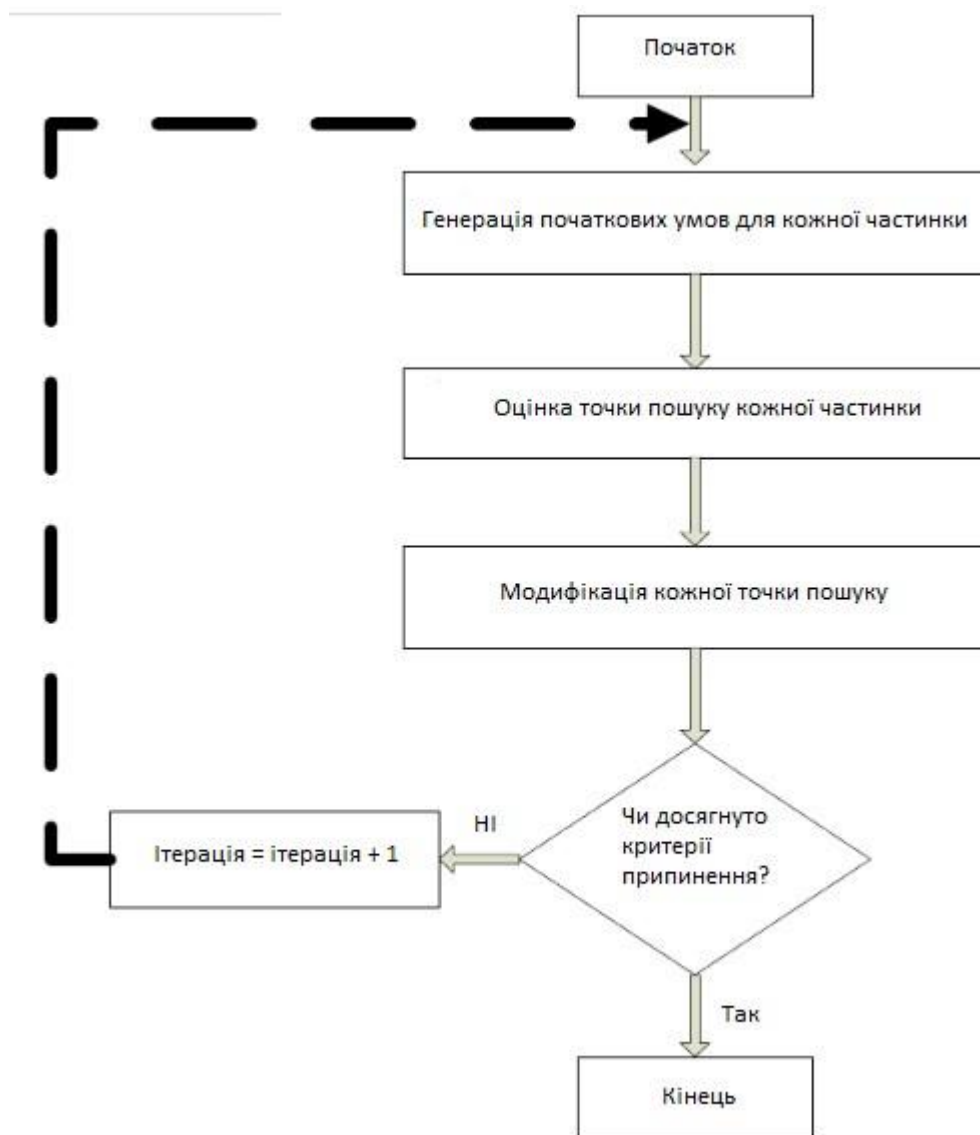


Рисунок 6: Блок-схема оптимізації рою частинок [18]

2.3 Адаптивний PID-регулятор з використанням нейронних мереж

Пропорційно-інтегрально-диференційний (ПІД) регулятор є однією з найдавніших технік керування і досі широко використовується в промисловості завдяки простоті впровадження, надійній продуктивності та простоті фізичного принципу. ПІД-регулятор є лінійною методологією керування з дуже простою структурою керування. Цей тип регулятора працює безпосередньо на сигналі похибки, який є різницею між бажаним виходом і фактичним виходом, і генерує сигнал керування, що керує процесом. Для досягнення відповідної роботи замкнутого контуру необхідно налаштувати три параметри ПІД-регулятора. Методи налаштування параметрів ПІД-регулятора класифікуються як традиційні та методи штучного інтелекту (АІ). Традиційні методи, такі як метод Ціглера-Нікольса, не забезпечують оптимальних параметрів налаштування ПІД-регулятора і зазвичай призводять до відгуків замкнутого контуру, які характеризуються коливаннями та великим перерегулюванням. Підходи штучного інтелекту, такі як генетичний алгоритм та оптимізація рою частинок, успішно застосовуються для вирішення задачі оптимізації налаштування параметрів ПІД-регулятора для досягнення кращої продуктивності.

У багатьох застосуваннях звичайний ПІД-регулятор не задовольняє специфікації керування, і потрібні більш чутливі регулятори для таких застосувань. Налаштування таких складних регуляторів є досить складним завданням. Регулятор, налаштований на одних умовах роботи, може потребувати повторного налаштування на інших умовах роботи, інакше він буде працювати неоптимально. Погано налаштований складний регулятор може погіршити керування, тому бажано мати якусь автоматичну настройку параметрів регулятора при зміні умов роботи об'єкта [19].

Рисунок 7 представляє побудовану ПІД-систему керування з використанням нейронних мереж для налаштування параметрів регулятора, де ідентифікація та розрахунок параметрів є більш точними, ніж у

						Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

звичайного ПІД-регулятора, і більш придатними для онлайн-застосувань. Нейронна частина є адаптацією контуру регулятора. Нейронна мережа повинна виявити будь-які зміни у виході процесу та налаштувати регулятор відповідно до одного з доступних критеріїв налаштування [20].

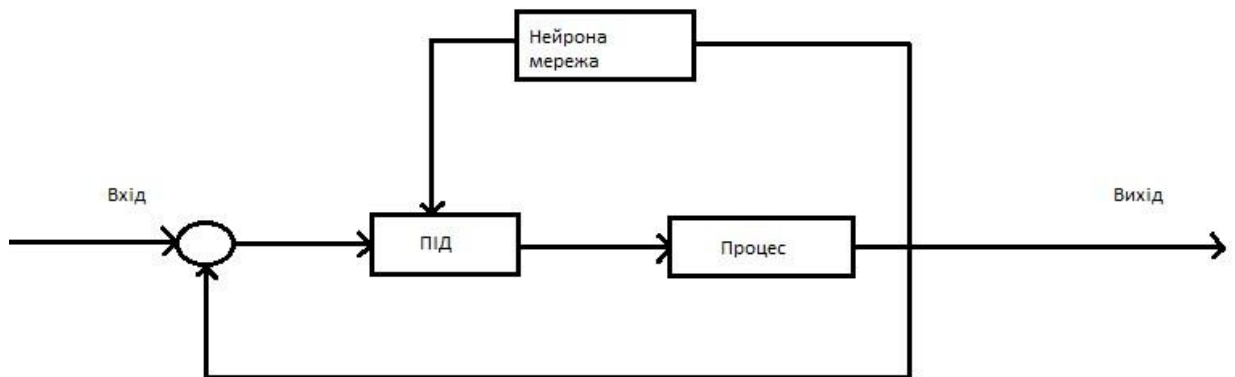


Рис. 7. Система налаштування ПІД

Для лінійних регуляторів існує кілька критеріїв налаштування.

1. Інтегральна абсолютна помилка.
2. Демпфування чверті амплітуди.
3. Інтегральна квадратна помилка.

Нейронна мережа повинна навчитися певним шаблонам виходу процесу і мати змогу отримати його модель у термінах коефіцієнта підсилення процесу, часу затримки та часу запізнювання. Вона також повинна навчитися певним шаблонам моделі процесу і відповідним налаштованим параметрам PID.

Припускається, що процес моделюється наступним рівнянням.

Де $G(S)$ є передаточною функцією процесу: $G(S) = c/m$

де m - вихід регулятора, а c - вихід процесу.

Незалежно від часу затримки, наведене вище рівняння може бути розв'язане будь-якими чисельними методами: методом Ейлера, покращеним методом Ейлера або методом Рунге-Кутти. Ефект часу затримки реалізується за допомогою масиву, розмір якого дорівнює значенню часу затримки, поділеного на час вибірки. Цей масив спочатку ініціалізується до нуля, і вихід Рунге-Кутти подається до цього масиву. Шляхом зсуву елементів масиву включається ефект часу затримки.

Рунге-Кутта сам по собі отримує вихід регулятора. Але якщо користувач вирішує ввести збурення, вхід до цього модуля буде вихід регулятора, від якого віднімається значення збурення. Насправді модуль процесу включає внутрішньо два модулі: Рунге-Кутта і Час затримки. Рунге-Кутта відповідає за розв'язання диференціального рівняння, а Час затримки відповідає за включення ефекту часу затримки у модуль процесу.

Модуль PID-регулятора

Цей модуль використовується для імітації ПІД-регулятора. Вхід регулятора є сигналом помилки, а вихід регулятора позначається як m , де:

Де, K_p — є статичним коефіцієнтом підсилення p

- p – пропорційна смуга
- e - вхідна помилка для регулятора
- I - час інтеграції (відновлення)
- D - час диференціювання

Рис. 8. - це блок-схема, що показує функцію модуля регулятора.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

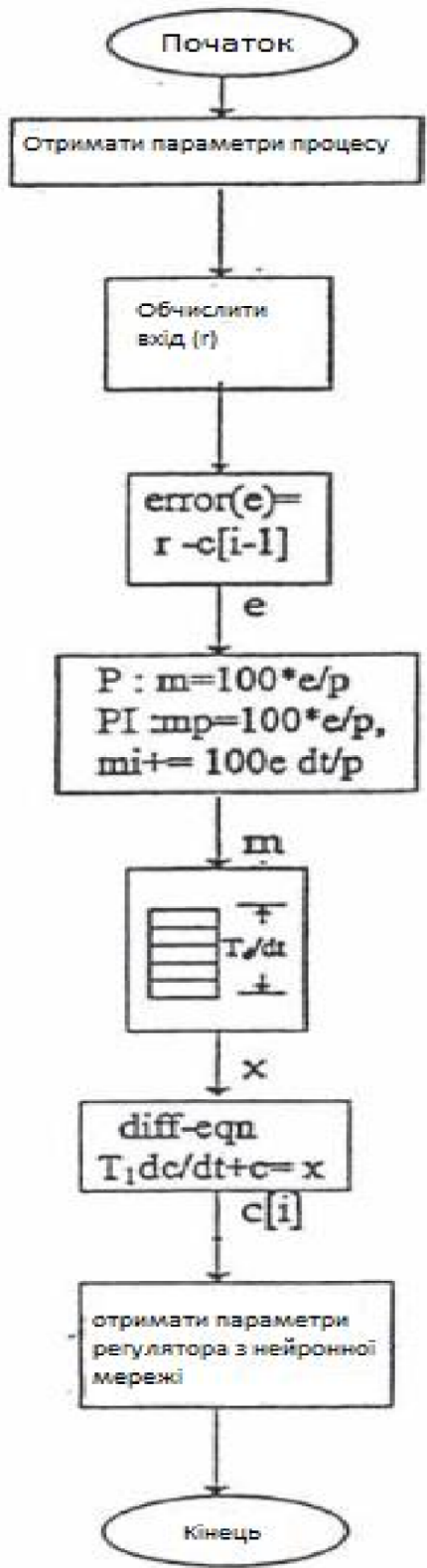


Рис. 8. Функція модуля регулятора

Змн.	Арк.	№ докум.	Підпис	Дата

2.4 Впровадження нейронної мережі

Більшість практичних регуляторів, навіть для розподілених систем керування, є ПІД-регуляторами. Це пов'язано з простотою та ефективністю цієї техніки. Випадки, де ПІД-регулятори мають деякі недоліки, не варті складності впровадження більш витончених технік. Тому іноді найкращим рішенням є покращення ПІД-регулятора шляхом налаштування певних ефективних методів налаштування, серед яких є нейронні мережі.

Використання нейронних мереж для моделювання та керування системами має велике значення в області керування. Нейронна мережа — це система взаємопов'язаних елементів, змодельована за зразком людського мозку [21]. Зростання інтересу до нейронних мереж пов'язане з успіхами в алгоритмах навчання мереж і нещодавніми досягненнями в апаратному забезпеченні, які зробили можливим використання надзвичайно швидких і відносно недорогих комп'ютерів для реалізації цих алгоритмів. Нейронні мережі представляють масово паралельну розподілену обробку, здатну до постійного покращення продуктивності шляхом динамічного навчання. Класичне лінійне керування забезпечує надійність у відносно малому діапазоні невизначеності. Адаптивні методи керування були розроблені для систем, які повинні працювати у великих діапазонах невизначеності через великі зміни значень параметрів, умов навколишнього середовища та вхідних сигналів. Ці адаптивні методи зазвичай включають другий контур зворотного зв'язку, який знаходиться поза першим контуром зворотного зв'язку. Цей другий контур зворотного зв'язку може мати здатність відслідковувати параметри системи, умови навколишнього середовища та характеристики вхідних даних; зворотний зв'язок тоді може змінювати параметри в елементах компенсації внутрішнього контуру, щоб підтримувати прийнятні характеристики продуктивності.

Мета розробки інтелектуальної системи керування схожа на мету адаптивної системи керування [22]. Однак є різниця. Для інтелектуальної системи керування діапазон невизначеності може бути значно більшим, ніж

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

той, який можуть витримати алгоритми адаптивних систем. Завдання інтелектуального керування полягає в розробці системи з прийнятними характеристиками продуктивності в дуже широкому діапазоні невизначеності. У типовому застосуванні керування, наприклад у робототехніці, потрібно вирішувати ряд проблем, які також є загальними для проектування регуляторів для великих динамічних систем. Одним з трьох підходів, що мають потенціал для інтелектуального керування, є нейронні мережі як елементи компенсації в системах керування. З нейронними мережами проблема керування може розглядатися як проблема розпізнавання образів [23], де образи, що розпізнаються, є "сигналами змін", які перетворюються на "сигнали дій" для заданої продуктивності системи. Регулятор повинен в реальному часі розпізнавати і виділяти образи змін та "вчитися" робити це швидко. Властивості розпізнавання і відображення з постійним покращенням самоорганізації та прийняття рішень є деякими з потенційних переваг використання штучних нейронних мереж для розробки та впровадження інтелектуальних регуляторів. Один з найбільш цікавих результатів нейронних мереж був той, що з компенсацією зворотного/прямого зв'язку, реалізованою нейронною мережею, в процесі навчання компенсація прагне переміщуватися на шлях прямого зв'язку з незначною компенсацією зворотного зв'язку.

Нейронна мережа включає три шари: вхідний шар, один прихований шар і один вихідний шар. Кожен шар з'єднаний з іншим за допомогою ваг. Кожен вузол у вхідному шарі з'єднаний з усіма вузлами прихованого шару, і кожен вузол у прихованому шарі з'єднаний з усіма вузлами вихідного шару через інші синаптичні ваги. Синаптичні ваги з'єднують кожні два шари. Кожен вузол з'єднаний синаптичними вагами з усіма вузлами наступного шару. Ваги зберігаються у файлах.

Впровадження нейронної мережі включає три фази:

- 1 - Фаза введення даних.
- 2 - Фаза навчання.

						Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

3 -Фаза тестування.

Фаза введення приймає вектор вхідних-вихідних даних N, M, L, P, NP, Rate та Eges як вхідні дані.

- NP - кількість панер.
- N - кількість вхідних вузлів.
- L - кількість вузлів прихованого шару.
- M - кількість вихідних вузлів.
- P - кількість прихованих шарів.
- Rate - швидкість навчання.
- Eges - точність помилки.

Фаза навчання приймає вектор вхідних-вихідних даних та значення ваг як вхідні дані, потім зберігає ці значення у файл. Фаза навчання розділена на дві частини:

Навчання в режимі офлайн: включає дві функції: функцію ідентифікації та функцію налаштування. Воно приймає кожен патерн вхідних і вихідних даних до модуля процесу для навчання на ньому та представляє приблизні значення параметрів процесу (/ /) а потім використовує їх для отримання параметрів PID.

Навчання в режимі онлайн: частина онлайн використовується, коли користувачеві потрібно змінити параметри системи. У цей час мережа починає навчатися новим значенням і починає імітувати контур. Третя фаза, яка є тестовою, використовує два файли: файл ваг і файли даних. Користувач може ввести патерн для тестування або використовувати файли даних і ваг для тестування мережі. Файл даних містить вхідну інформацію, а файл ваг містить збережені ваги у формі масиву.

Модуль ідентифікації: цей модуль приймає адресу патерна як вхід, змінює параметри системи для імітації контура керування, а потім зберігає вхідні та вихідні дані процесу у масив у двійковому файлі та записує необхідні дані (параметри системи для кожного патерна) у двійковий файл.

						Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Потім навчає мережу комбінаціям ідентифікації. Нарешті, він генерує параметри системи (. ,) як вихідні дані.

Модуль налаштування: Після того як ідентифікація розпізнала параметри системи, викликається модуль налаштувань регулятора для розрахунку налаштувань PID, відповідних розпізнаним параметрам системи.

Модуль ВРН: Тут обчислюється чисте значення вхідного сигналу для одиниць прихованого шару, обчислюємо помилку для прихованих одиниць, оновлюємо ваги на вихідному шарі, оновлюємо ваги на прихованому шарі та обчислюємо помилку для зупинки навчання, коли досягається необхідна точність.

2.5. Регулювання нейронних мереж

Прості нелінійності можуть ефективно використовуватися в системах управління. Ключовою ідеєю використання деяких технік на основі нелінійностей є використання функцій від кількох змінних. Такі функції нелегко охарактеризувати простим способом. На перший погляд ці методи можуть здаватися досить складними, але насправді це лише нелінійні функції.

Нейронні мережі виникли як спроби створити прості моделі нейронної активності в мозку та пристрої, які могли б розпізнавати шаблони та виконувати прості навчальні завдання. Нижче наведено короткий опис, який передає основну ідею.

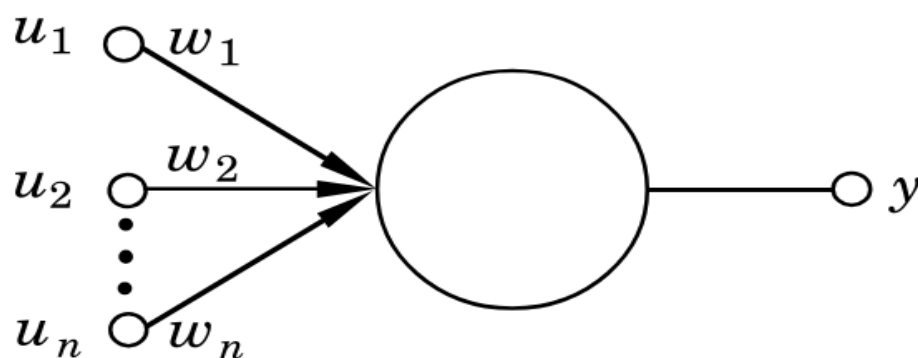


Рис.9. Схематична діаграма простого нейрона

Простий нейрон

На рисунку 9 показана схематична діаграма простого нейрона. Система має багато входів і один вихід. Якщо вихід u , а входи x_1, x_2, \dots, x_n , то відношення вхід-вихід описується як:

де числа w_1, w_2, \dots, w_n називаються вагами. Функція f є так званою сигмоїдною функцією, проілюстрованою на рисунку 10. Така функція може бути представлена як:

$$f(x) = \frac{1}{1 + e^{-\alpha x}}$$

де α є параметром. Таким чином, ця модель нейрона є просто нелінійною функцією. Деякі спеціальні класи функцій можуть бути апроксимовані рівнянням (2.7)

Більш складні моделі можна отримати, з'єднуючи нейрони разом, як показано на рисунку 11. Така система називається нейронною мережею або нейронною сіткою. Часто додається прикметник "прямого поширення", щоб вказати, що нейрони з'єднані у напрямку прямого поширення. Існують також інші типи нейронних мереж. У мережі з прямим поширенням вхідні нейрони з'єднані з шаром нейронів, виходи нейронів першого шару з'єднані з нейронами другого шару і так далі, поки не отримаємо виходи. Проміжні шари в мережі називаються прихованими шарами.

						Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

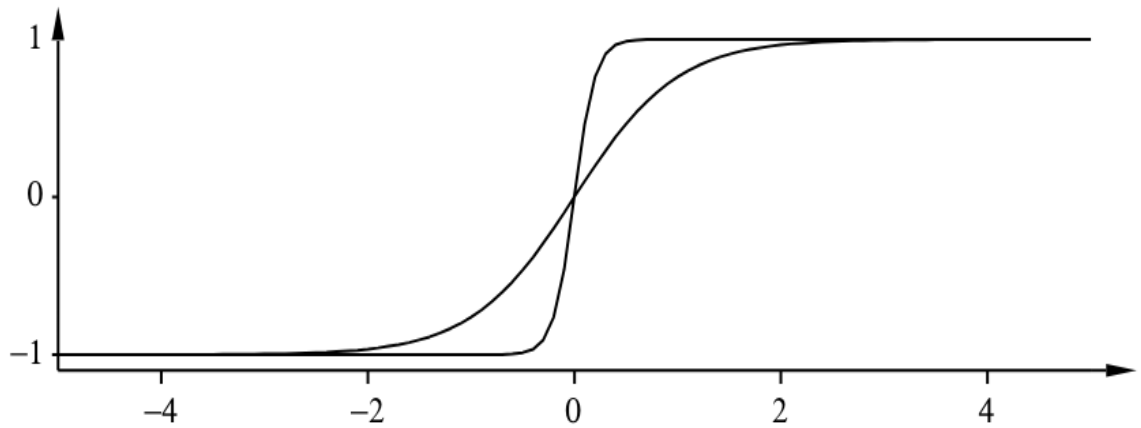


Рис. 10 Сигмоїдальні функції

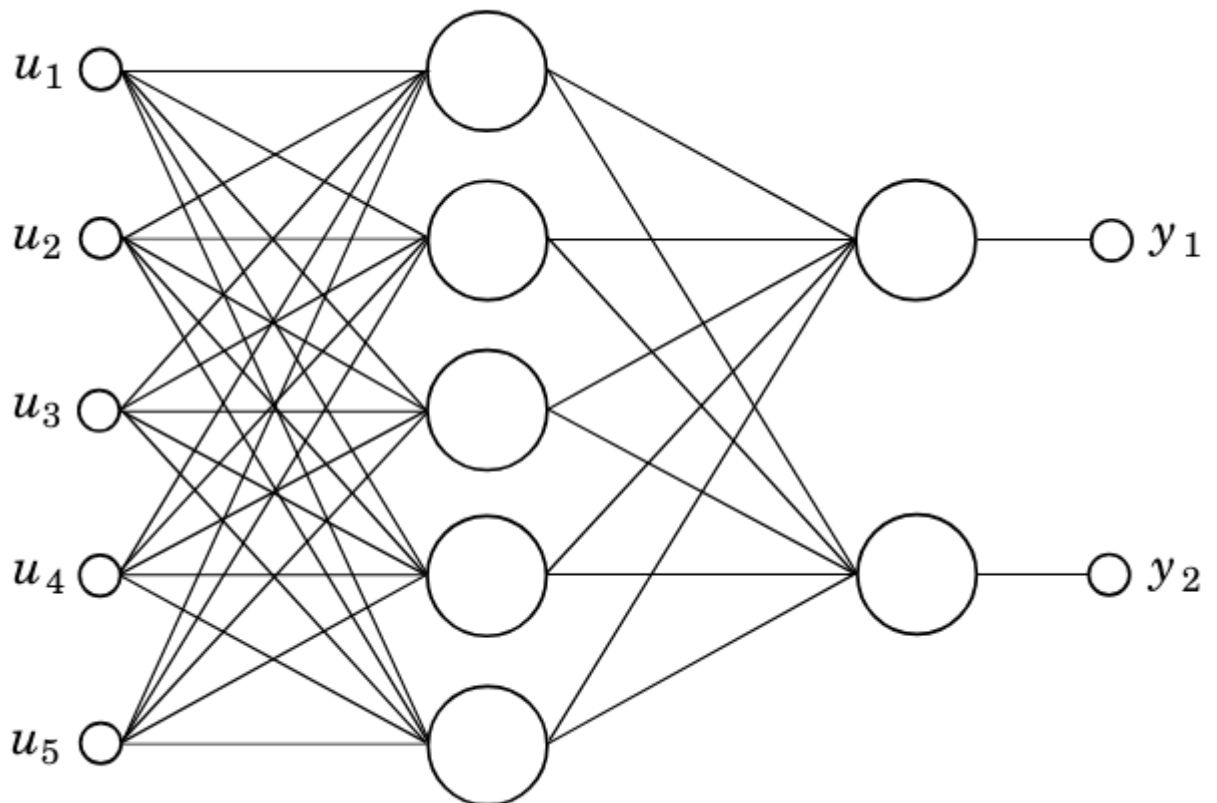


Рис. 11 Нейронна мережа прямого поширення.

Кожен нейрон описується рівнянням (2.7). Відношення вхід-вихід нейронної мережі таким чином є нелінійною статичною функцією. І навпаки, нейронна мережа розглядається як один із способів побудови нелінійної функції від кількох змінних. Представлення нейронної мережі передбачає, що нелінійна функція від кількох змінних будується з двох компонентів: однієї нелінійної функції, сигмоїдної функції (2.8), яка є скалярною функцією однієї змінної; та

лінійних операцій. Це простий спосіб побудови нелінійності з простих операцій. Одна з причин, чому нейронні мережі цікаві, полягає в тому, що практично всі неперервні функції можна апроксимувати нейронними мережами з одним прихованим шаром. Було виявлено, що практично використовувати більше прихованих шарів, оскільки тоді можна використовувати менше ваг.

Важливо звернути увагу, що в нейронній мережі багато параметрів (ваг). Припускаючи, що в шарі є n нейронів, якщо всі нейрони з'єднані, потрібно параметрів для опису зв'язків між двома шарами. Ще однією цікавою властивістю нейронної мережі є так звані процедури навчання. Це алгоритм, який дозволяє знаходити параметри (ваги) так, щоб функція відповідала заданим значенням вхід-вихід. Параметри зазвичай отримуються рекурсивно, задаючи вхідне значення функції та бажане вихідне значення. Ваги потім налаштовуються, щоб дані відповідали. Процедура повторюється, поки не буде отримано добрий збіг для розумного набору даних. Ця процедура називається навчанням мережі. Популярний метод навчання мережі прямого поширення називається зворотним поширенням. З цієї причини мережу прямого поширення іноді називають мережею зворотного поширення.

Нейронна мережа прямого поширення є нічим іншим, як нелінійною функцією від кількох змінних з процедурою навчання. Функція має багато параметрів (ваг), які можуть бути налаштовані за допомогою процедури навчання, щоб функція відповідала заданим даним. Навіть якщо це надзвичайно спрощена модель реального нейрона, це дуже корисний компонент системи. У процесі управління часто ефективно використовуються нелінійні функції. Один із прикладів - калібрування датчиків. Існує багато ситуацій, коли прилад має багато різних датчиків, виходи яких повинні бути нелінійно поєднані для отримання бажаного вимірюваного значення. Нелінійні функції також можуть використовуватися для розпізнавання шаблонів.

						Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Розділ 3. Дослідження методу PIDNN для налаштування ПД-регуляторів на основі нейронних мереж.

3.1. Методи визначення параметрів ПД-регуляторів

Два класичні методи визначення параметрів ПД-регуляторів були представлені Ціглером і Нікольсом у 1942 році. Ці методи досі широко використовуються, як у їх оригінальній формі, так і в деяких модифікаціях. Вони часто є основою для процедур налаштування, які використовують виробники регуляторів та процесна промисловість. Методи базуються на визначенні деяких характеристик динаміки процесу. Параметри регулятора потім виражаються у вигляді простих формул на основі цих характеристик регуляторів

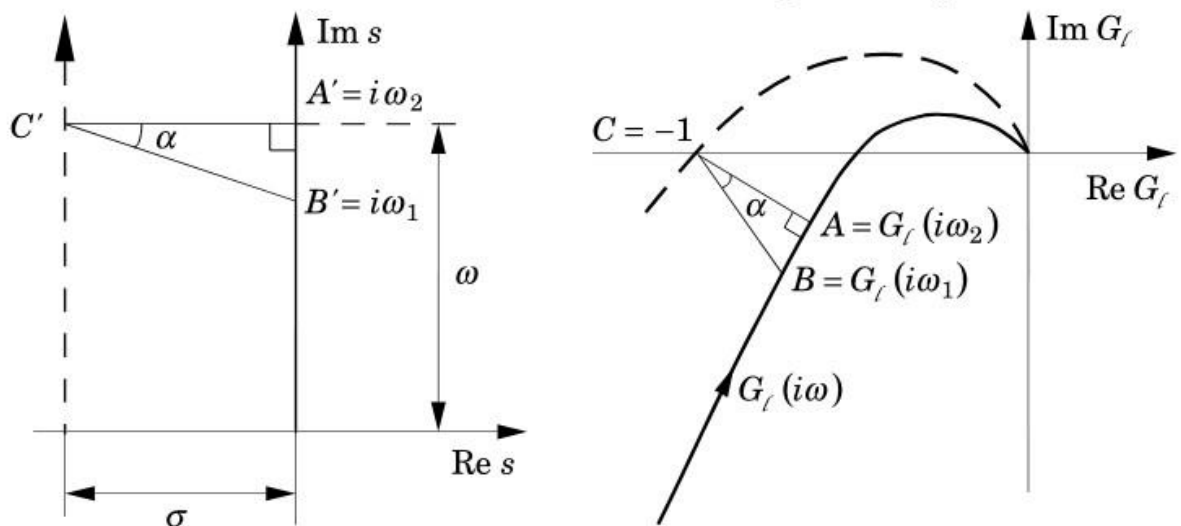


Рис. 12. Подання функції передачі контуру у вигляді карти комплексних площин.

		АКСУ				Пояснювальна записка		
Змн.	Арк.	№ докум.	Підпис	Дата				
Розробив		Біленький М.В.			Дослідження методу PIDNN для налаштування ПД-регуляторів на основі нейронних мереж.	Літ.	Арк.	Акрушів
Перевірів		Безкоровайний Ю.М.					42	5
Н-контр.		Безкоровайний Ю.М.				№ 151-403-СУ		
Зав.каф.		Мельник Ю.В.						

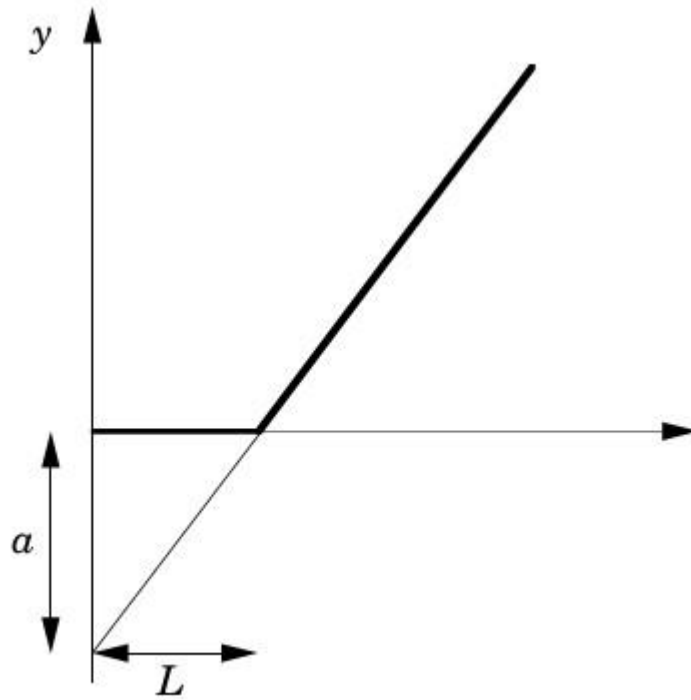


Рис. 13. Характеризація відгуку на крок у методі Ціглера-Нікольса для крокового відгуку.

Метод крокового відгуку

Перший метод проектування, представлений Ціглером і Нікольсом, базується на реєстрації відгуку системи на одиничний стрибок у режимі відкритого контуру, який характеризується двома параметрами. Параметри визначаються з одиничного крокового відгуку процесу, як показано на Рисунку 13.

Спочатку визначається точка, де нахил крокового відгуку має максимум, і в цій точці проводиться дотична. Перетини між дотичною та координатними осями дають параметри

a та L . Це відповідає моделюванню процесу за допомогою інтегратора та часової затримки. Ціглер і Нікольс безпосередньо дали параметри ПІД-регулятора як функції від a та L .

Метод частотної характеристики

Цей метод також базується на простій характеристиці динаміки процесу. Проектування базується на знанні точки на кривій Найквіста функції передачі процесу $G(s)$, де крива Найквіста перетинає негативну дійсну вісь. ця точка характеризується параметрами які називаються кінцевим підсиленням і кінцевим періодом.

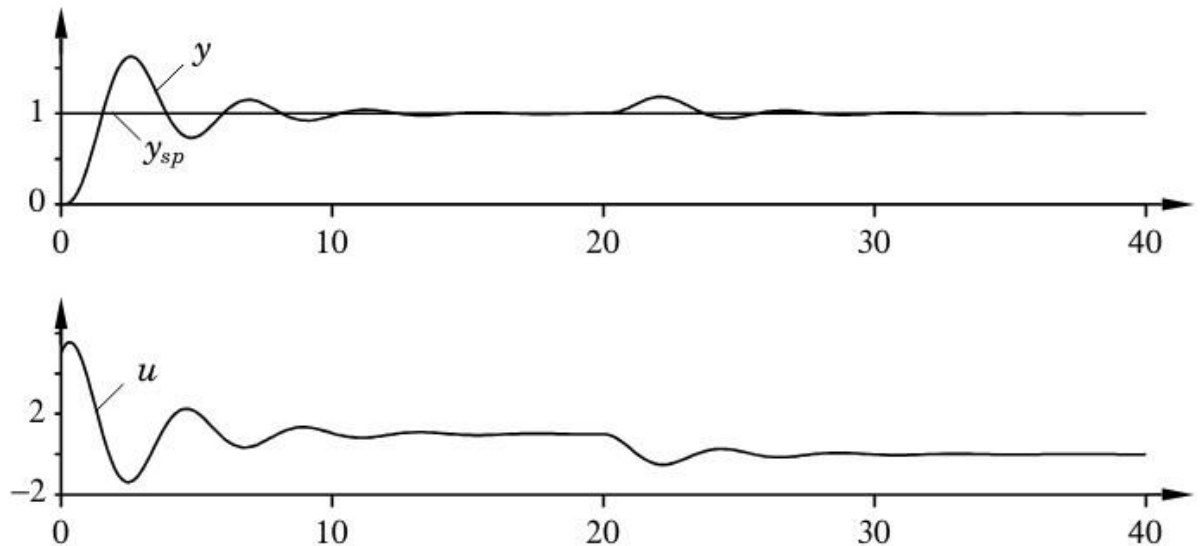


Рисунок 14. Відгук на задане значення і на збурення навантаження процесу з передавальною функцією $1/(s+1)^3$, керованого ПІД-регулятором, налаштованим за методом крокового відгук Ціглера-Нікольса.

Діаграми показують задане значення y_{sp} , вихід процесу y та керуючий сигнал u

Оцінка налаштувань Ціглера-Нікольса

Процедури налаштування Ціглера-Нікольса є простими та інтуїтивно зрозумілими. Вони вимагають мінімальних знань про процес і можуть бути застосовані з незначними зусиллями. Ось кілька причин, чому вони так широко використовуються. Основним недоліком є те, що базовий критерій проектування полягає в отриманні системи із замкнутим контуром з коефіцієнтом затухання амплітуди чверті ($d = 0.25$). Це забезпечує хороше придушення збурень навантаження, але також створює систему із замкнутим контуром, яка дуже погано демпфована і має погані стабільні

						Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

межі. Коефіцієнт підсилення із замкнутим контуром зазвичай у 2-3 рази вищий, ніж необхідно. Метод частотної характеристики є більш надійним, ніж метод крокового відгуку. Одна з причин цього полягає в тому, що кінцеве підсилення однозначно визначене, але є багато способів визначити явну мертву зону. Метод крокового відгуку зазвичай також дає дещо вищі коефіцієнти підсилення.

Карти налаштування

Оскільки методи Ціглера-Нікольса надають лише приблизні значення, необхідно здійснювати ручне налаштування для досягнення бажаної продуктивності. Для керування ручним налаштуванням було розроблено пристрій під назвою карти налаштування. Метою цих карт є надати інтуїтивне розуміння того, як зміни в параметрах регулятора впливають на поведінку системи із замкнутим контуром. Карти налаштування являють собою двовимірні масиви перехідних відгуків або частотних характеристик, організовані систематичним чином.

3.2. Впровадження та налаштування ПД-регулятора

ПД-регулятор зазвичай є найкращим вибором при проектуванні системи керування через простоту його впровадження та проектування. Його стандартна архітектура зазвичай є найкращим вибором, оскільки вона складається з лінійної комбінації трьох компонентів: пропорційної, інтегральної та диференційної дій, які визначаються наступним чином:

(3.1)

де u – це вихід керування, а e – компоненти помилки (дії), які визначаються наступним чином:

—

де e - вхідна помилка.

Широке розповсюдження ПД-регулятора пояснюється його низькою вартістю (у порівнянні з вартістю мікропроцесора), а також можливістю

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

використання без точного знання керованого промислового процесу. Цю останню характеристику не можна недооцінювати, враховуючи велику різноманітність промислових процесів. Можливо, що регулятори, розроблені спеціально для конкретного промислового процесу, які можуть включати модель або симуляцію процесу, могли б бути кращими за ПІД-регулятор, але це вимагало б значного збільшення витрат на проектування та реалізацію.

Компоненти рівняння (3.1) є основними діями для лінійного незмінного в часі регулятора, який забезпечує пропорційну дію (відповідний коефіцієнт підсилення сигналу), інтегральну дію (конвергенція помилки до нуля з часом) та диференціальну дію (швидкість корекції, яка змінюється з часом). Не обов'язково, щоб усі дії відбувалися одночасно. Насправді, часто використовуються лише пропорційні (P) та пропорційно-інтегральні (PI) регулятори.

Техніки налаштування ПІД-регулятора, запропоновані Ціглером і Нікольсом (ZN) [24], є одними з класичних методів налаштування ПІД-регуляторів. Вони мають евристичний характер і базуються на попередній оцінці установки для визначення початкових значень регулятора. Перша техніка використовує апроксимацію установки до системи першого порядку з явною затримкою для розрахунку параметрів регулятора. Друга техніка стосується різних типологій установок вищого порядку, на яких виконується тест із замкнутим контуром із чистим пропорційним підсиленням. Це дозволяє встановити пропорційне підсилення, яке приводить систему на межу її стабільності. Потім це початкове значення допомагає отримати оптимізовані цільові коефіцієнти.

Техніки Ціглера, Нікольсона зазвичай підходять для багатьох типів систем, але можуть спричиняти значні коливання та перерегулювання через відносно низьке демпфування.

Іншою класичною технікою налаштування ПІД-регуляторів є метод екстремального пошуку (ES) [25], який широко використовується для

						Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

налаштування в реальному часі. Цей метод не вимагає знань про систему або функцію помилки. Однак він припускає, що функція помилки відповідає передумовам теореми градієнта, оскільки експериментальний пошук використовує мінімізацію градієнта для розрахунку трьох коефіцієнтів ПДрегулятора. Цей метод можна реалізувати апаратно з послідовністю фільтра, модулятора та інтегратора, що вказує на тенденцію помилки та оцінку параметрів. Обидва підходи, Ціглера-Нікольса та експериментальний пошук, значною мірою перевершені більш складними техніками, заснованими на ідентифікації систем або методах оптимізації, але вони залишаються важливими методами для порівняння продуктивності нових методів налаштування ПД-регуляторів.

3.3. Опис нейронної мережі ПД-регулятора

У парадигмі навчання з учителем штучний агент отримує набір мічених прикладів, що складаються з пар <входи, вихід>, де вихід є міткою класифікації, яку потрібно передбачити. Вихідні значення створюються невідомою функцією керування f . Завданням агента є навчання функції f , яка, згідно з термінологією машинного навчання, називається гіпотезою, шляхом пошуку в просторі гіпотез функції f , де передбачається існування принаймні хорошої апроксимації f . Якість або продуктивність знайденої функції f потім оцінюється на окремому наборі прикладів, що називається тестовим набором, який не зустрічався в наборі прикладів, використаних для знаходження f . Як зазначено вище, вибір простору гіпотез визначає можливість знайти рішення f , яке узгоджується з навчальним набором. Прийнятне рішення f може бути не ідеальним, але воно пояснюватиме більшість навчальних прикладів і добре працюватиме на тестовому наборі.

Нейронна мережа (NN) [26] є нелінійною математичною структурою, що складається з набору взаємопов'язаних нейронів або вузлів. Властивості нейронної мережі визначаються її топологією та характеристиками вузлів. Нейронні мережі прямого поширення можуть бути представлені як набір рівнів вузлів, де ми можемо виділити вхідний рівень, підмножину вузлів,

						Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

що отримують входи, вихідний рівень, набір вузлів, що генерують виходи нейронної мережі, та проміжні рівні, тобто всі інші вузли. У мережі прямого поширення нейронна мережа проходить тільки в одному напрямку від вхідного шару до вихідного шару, вона не містить циклів і, таким чином, не має внутрішньої пам'яті.

Існують й інші типи нейронних мереж: наприклад, у рекурсивній нейронній мережі виходи знову подаються на вхідні вузли, що створює ефект пам'яті, коли попередні входи/виходи можуть також впливати на поточні виходи разом із поточними входами. Оскільки в межах цього дослідження нас цікавить моделювання ПІД-регулятора, який є динамічною моделлю, ми розглядатимемо рекурсивні нейронні мережі.

Опишемо ПІД нейронну мережу (PIDNN), використану в експерименті. PIDNN використовує три спеціальні типи нейронів: P-нейрон, I-нейрон і D-нейрон, які реалізують три фундаментальні дії керування. Мережа побудована з трьох рівнів. Вхідний рівень складається з двох нейронів, які приймають як вхідні дані вихід установки та еталонний вхід. Другий рівень складається з P-, I- та D-нейронів. Третій рівень об'єднує три компоненти керуючої дії у керуючу дію, яку потрібно застосувати до установки (рис. 15).

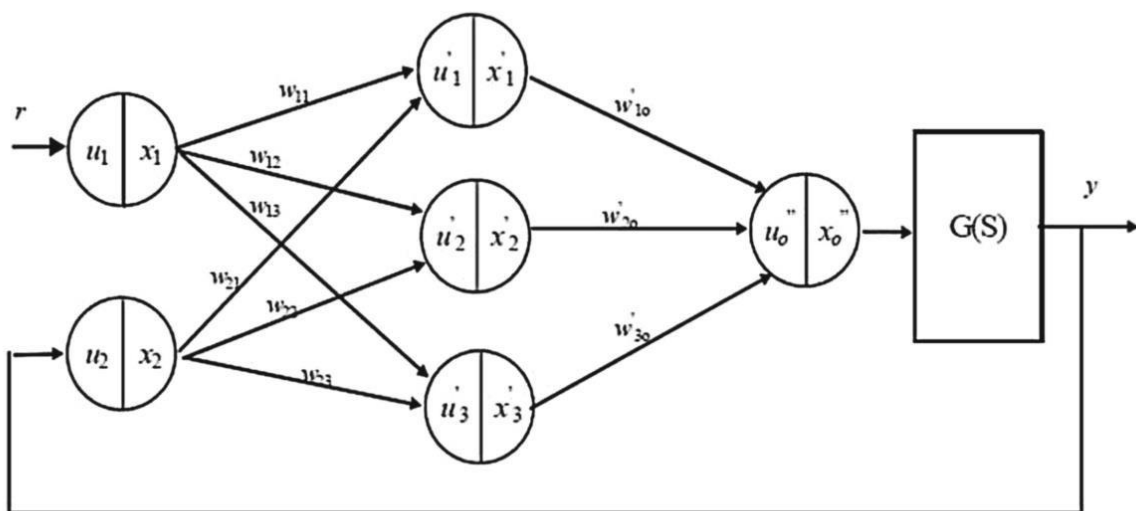


Рис. 15. PIDNN, де $G(s)$ - це система, яку потрібно керувати.

Функції нейронів визначаються наступним чином:

—

Функції інтеграції та диференціювання можуть бути реалізовані по-різному: було обрано чисельне диференціювання (яке можна застосувати навіть у найпростіших вбудованих системах), тоді як функція інтеграції полягає у виконанні інтеграції до поточного часу.

Ці вибори були керовані результатами, отриманими під час попередніх експериментів. помічено, що вибір реалізації впливає на ефективність PIDNN, оскільки вони вимагають різних методів дискретизації вхідних сигналів.

Ваги визначаються наступним чином:

Проходячи через мережу від вхідного шару до вихідного шару:

— —

—

						Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

3.4. Експериментальні налаштування та результати

Обраною моделлю для установки есперементальної оцінки є:

$$G(s) = \frac{(1 - 5s)}{(1 + 10s)(1 + 20s)}$$

Ця система є системою з немінімальною фазою з нулем, що має додатню дійсну частину. Крім того, вона має два малих полюси, хоча асимптотично стабільні, вона дійсно повільна. Обрана система є лінійна інваріантна в часі для початку дослідження, хоча методологія, заснована на нейронних мережах, також повинна дозволити синтезувати регулятор для нелінійної системи.

Приклад робочого простору в Simulink, використаного для проведення експериментів, показаний на рис. 16.

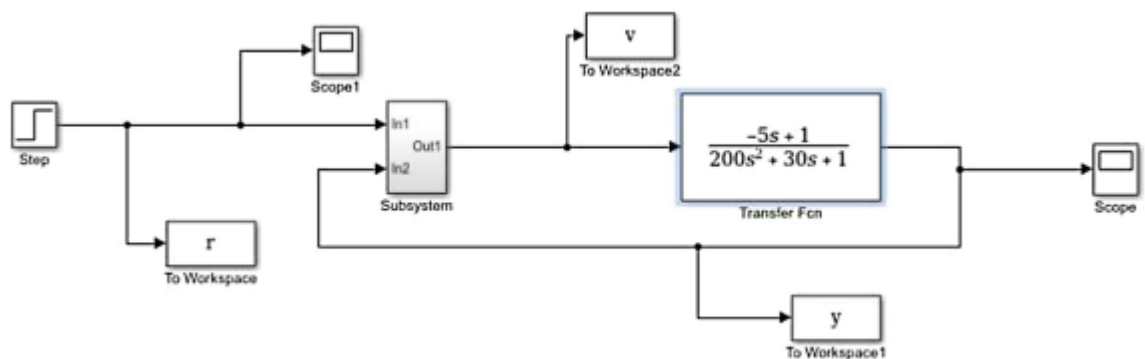


Рис 16. Робочий простір для симуляцій

Це метод для автономного налаштування регулятора в режимі off-line, заснований на повторних спробах покращити характеристики регулятора. Під час цих спроб не гарантовано, що регулятор досягне збіжності, тому проведення таких випробувань на реальній установці було б неможливим. На рис. 15 показана архітектура ПД-регулятора на основі нейронної мережі, використана в наших експериментах. Вона буде включена в архітектуру на рис. 14 в підсистемний блок.

У першій серії експериментів алгоритм зворотного поширення помилки, написаний у Matlab, змінює ваги зв'язків, ініціалізованих фіксованими значеннями. Початкові значення ваг безпосередньо взяті з другої техніки Ціглера-Нікольса, яка забезпечує такі початкові значення: $w_{11} = 3.53$, $w_{12} = 0.2101$, $w_{13} = 14.8260$ для ваг на входних краях вихідних нейронів (3.5, 3.6, 3.7), залишаючи початкові значення 1 і -1 для інших. Алгоритм працює наступним чином.

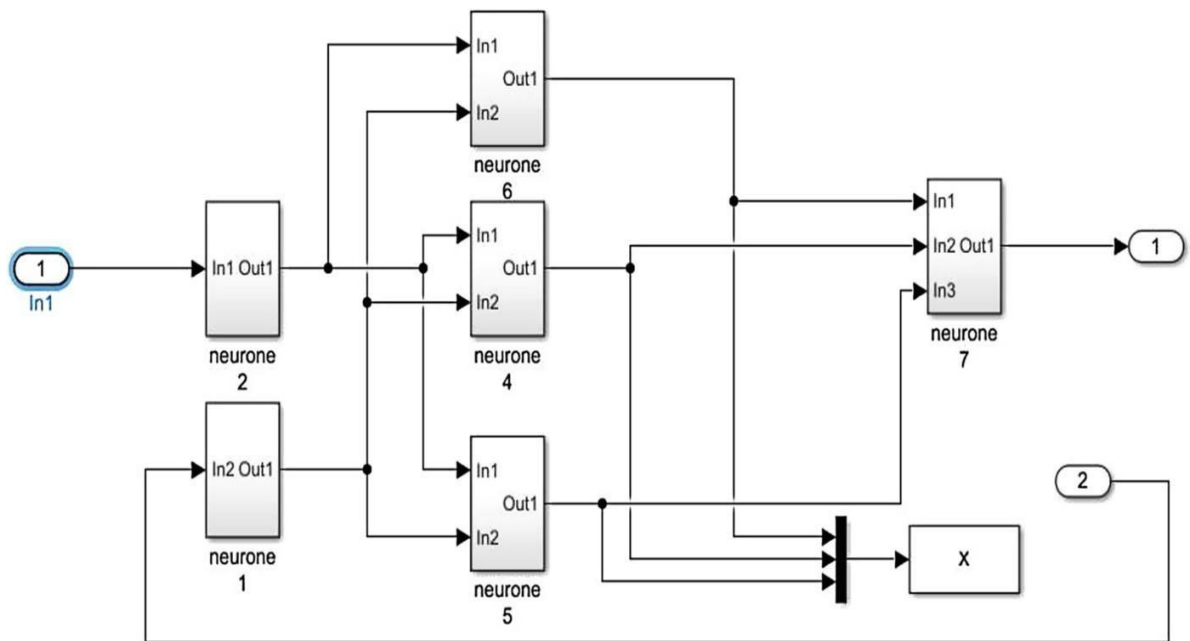


Рис. 17. PIDNN, реалізована в Simulink. PIDNN реалізована в підсистемному блоці на рис. 16.

// Глобальні змінні

// m - поточний зразок сигналів

// η_1 - швидкість навчання для першого шару

// η_2 - швидкість навчання для другого шару

// Визначення входних параметрів

// r - еталонний вхід установки

// y - реальний вихід системи із замкнутим контуром

// v - керуючий вхід установки

// x - вектор виходів нейронів PID

					Арк.
					51
Змн.	Арк.	№ докум.	Підпис	Дата	

Метод зворотного поширення помилки (r, y, v, x, K)

$J = 0$ // вектор оновлень, початково встановлений на нуль

$y_1 = y(m + 1)$ // наступний вихід

$y_2 = y(m)$ // поточний вихід

$v_1 = v(m - 1)$ // попередній керуючий вхід

$v_2 = v(m)$ // поточний керуючий вхід

$x_1 = x(m)$ // вектор вихідного сигналу на поточний момент часу для

трьох нейронів

$x_2 = x(m + 1)$ // вектор вихідного сигналу на наступний момент

часу для трьох нейронів

$$\Delta = 2 * (y_1 - y_2) * (y_2 - y_1) / (y_1 - y_2)$$

$$\Delta_1 = \Delta * k_i * (x_2(1) - x_1(1)) / ((r - y_2) - (r - y_1))$$

$$\Delta_2 = \Delta * k_i * (x_2(2) - x_1(2)) / ((r - y_2) - (r - y_1))$$

$$\Delta_3 = \Delta * k_d * (x_2(3) - x_1(3)) / ((r - y_2) - (r - y_1))$$

//Обчислення вектора помилок

For Each i From 0 To m

Do

$$(1) = J(1) - (\Delta * x_1(1, 1) / m)$$

$$(2) = J(2) - (\Delta * x_1(1, 2) / m)$$

$$(3) = J(3) - (\Delta * x_1(1, 3) / m)$$

$$(4) = J(4) - (\Delta_1 * r / m) - (\Delta_2 * v_1 / m) - (\Delta_3 * v_2 / m)$$

$$(5) = J(5) - (\Delta_1 * y_1 / m) - (\Delta_2 * y_1 / m) - (\Delta_3 * y_1 / m)$$

End Do

//оновлення коефіцієнтів для регулятора

$$k_i = k_i - ni * J(1)$$

$$k_i = k_i - ni * J(2)$$

$$k_d = k_d - ni * J(3)$$

$$1(1) = kj1(1) - ni1 * J(4)$$

$$2(1) = kj2(1) - ni1 * J(5)$$

					Арк.
					52
Змн.	Арк.	№ докум.	Підпис	Дата	

$$= [k_p, k_i, \dots, 1(1), k_j 2(1)]$$

$$= m + 1$$

(K)

Запропонований алгоритм контролює вихід системи, вихід регулятора та виходи окремих нейронів, порівнюючи результати з еталонним входом. Потім алгоритм навчається, дискретизуючи ці сигнали, кількість яких залежить від частоти дискретизації. Швидкості навчання η і η_l є основою цього процесу, оскільки вони встановлюють швидкість навчання PIDNN окремо для вхідних і вихідних нейронів.

Вплив вибору значень для швидкостей навчання буде досліджено в майбутній роботі, однак ми вже спостерігали, що відповідний вибір може забезпечити надійний або високоефективний регулятор, і також, що за межами деяких діапазонів PIDNN не буде збігатися, що призведе до створення регулятора, який спричиняє нестабільність у системі.

Під час попередніх експериментів було виявлено, що одним з факторів, що впливають на нестабільність у системі/регуляторі, є висока частота дискретизації. Насправді, висока частота дискретизації створює дискретизацію сигналу, початкова частина якого не має значення для кінцевої частини.

Для виконаних експериментів була обрана частота дискретизації 0,1 секунди та використаний наведений вище алгоритм паралельно на різних частинах кривої сигналу, усереднивши отримані коефіцієнти для отримання остаточних. Наведений нижче код показує, дію.

Ініціалізація коефіцієнтів ПІД-регулятора

$$= [\dots, 1, -1]$$

Отримання сигналу з симуляції

$m = 0;$

for each j in range of 1 and Кількість_Ітерацій

do

					Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	53

for each i in range of 1 and $\text{Кількість_Ітерацій}$

do

$m = i$

$K1 = \text{BackPropagation}(r, y, v, x, K(1), K(2), K(3), K(4), K(5))$

$m = \text{Кількість_Ітерацій} / 5 + i$

$K2 = \text{BackPropagation}(r, y, v, x, K(1), K(2), K(3), K(4), K(5))$

$m = 2 * \text{Кількість_Ітерацій} / 5 + i$

$K3 = \text{BackPropagation}(r, y, v, x, K(1), K(2), K(3), K(4), K(5))$

$m = 3 * \text{Кількість_Ітерацій} / 5 + i$

$K4 = \text{BackPropagation}(r, y, v, x, K(1), K(2), K(3), K(4), K(5))$

$m = 4 * \text{Кількість_Ітерацій} / 5 + i$

$K5 = \text{BackPropagation}(r, y, v, x, K(1), K(2), K(3), K(4), K(5))$

$K(1) = (K1(1) + K2(1) + K3(1) + K4(1) + K5(1)) / 5$

$K(2) = (K1(2) + K2(2) + K3(2) + K4(2) + K5(2)) / 5$

$K(3) = (K1(3) + K2(3) + K3(3) + K4(3) + K5(3)) / 5$

$K(4) = (K1(4) + K2(4) + K3(4) + K4(4) + K5(4)) / 5$

$K(5) = (K1(5) + K2(5) + K3(5) + K4(5) + K5(5)) / 5$

${}_1() = (1)$

${}_2() = (2)$

${}_3() = (3)$

${}_4() = (4)$

						Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

$$s(\) = (5)$$

// запуск симуляції та отримання нових сигналів

end

f = f+1

end

Повторення алгоритму на одному вибіркового сигналі виявилось неефективним, тому було вирішено паралелізувати дію алгоритму на п'ять вибірок сигналу. Таким чином кожен шматок сигналу робить більший внесок у визначення остаточних коефіцієнтів.

Остаточний результат показує значні покращення в порівнянні з продуктивністю, спостереженою при використанні методів Ціглера-Нікольса та експериментальний пошук, зі скороченням часу врегулювання на 60 і 20 секунд відповідно, як показано на рис. 18.

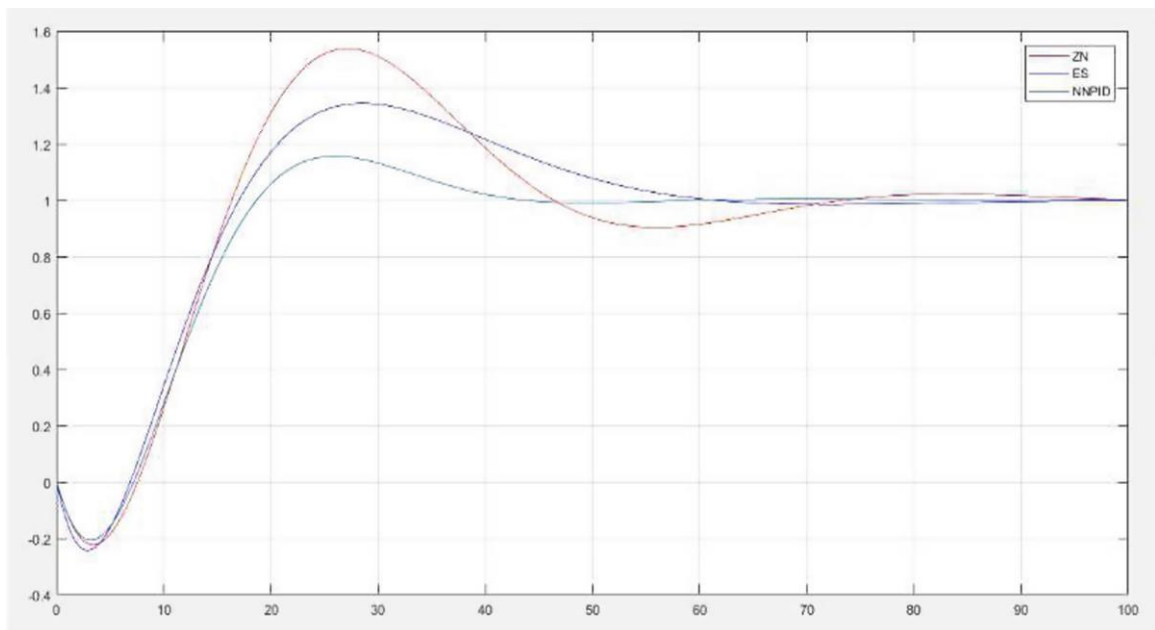


Рис. 18. Порівняльна реакція для технік PIDNN, ЦН та ЕС

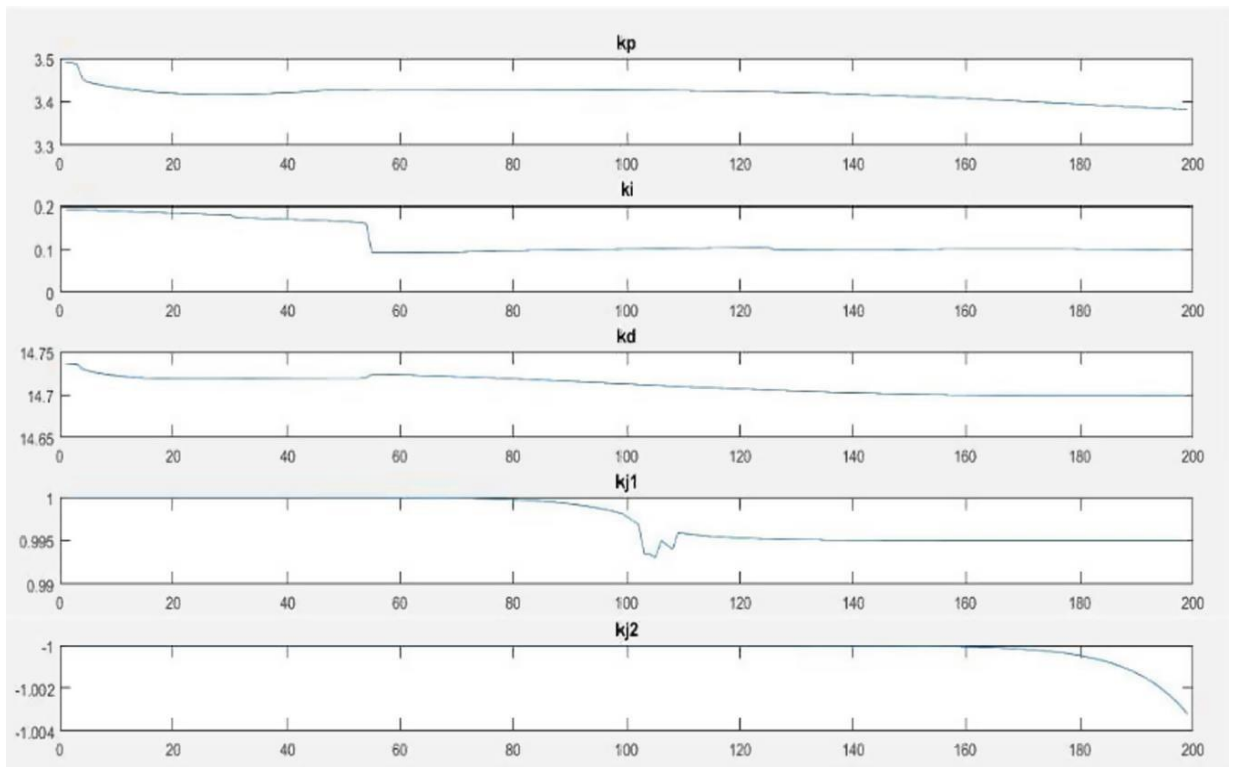


Рис. 19. Швидкість навчання мережі PIDNN

Як видно з наведених результатів, знайдене рішення також покращує undershoot на кілька відсоткових пунктів і, головне, overshoot, який на 30% і 10% менший, ніж у перших двох методів. Для навчання мережі було використано 1000 Кількість Ітерацій, після чого спостерігалось явище перенавчання, тому продовжувати було безглуздо. Криві навчання, використані для оцінки швидкості збіжності для швидкостей навчання, наведені на рис. 17.

Де k_p , k_i та k_d є коефіцієнтами для регулятора, а k_{j1} k_{j2} є вагами для передніх і зворотних зв'язків відповідно.

Висновок

Метою цієї роботи є дослідження нового методу PIDNN для налаштування ПІД-регуляторів на основі нейронних мереж. Досліджувалося як архітектурний вибір для контролера на основі нейронних мереж, так і його продуктивність у навчанні коефіцієнтів контролера в заданих умовах. Емпіричні результати показують, що метод PIDNN є дуже перспективним з точки зору швидшого навчання коефіцієнтів контролера порівняно з традиційними методами, такими як ZN і ES. Детально описано використані алгоритми, а також продемонстровано поведінку коефіцієнтів контролера під час навчання PIDNN.

У ході дослідження були розв'язані наступні задачі: Аналіз існуючих методів ПІД-регулювання та їх обмежень. Дослідження можливостей використання нейронних мереж для адаптивного налаштування параметрів ПІД-регулятора. Розробка алгоритму адаптивного ПІД-регулятора на основі нейронної мережі. Виконання моделювання та симуляції розробленої системи для оцінки її ефективності. Проведення експериментальних досліджень для підтвердження результатів моделювання.

Значення отриманих результатів для науки полягає в демонстрації можливостей використання нейронних мереж для адаптивного налаштування ПІД-регуляторів, що дозволяє значно покращити точність і стабільність регулювання. Практичне значення роботи полягає в розробці ефективного методу автоматичного налаштування ПІД-регуляторів, який може бути використаний у різних галузях промисловості, де потрібне точне автоматичне регулювання параметрів.

Метод PIDNN для налаштування ПІД-регуляторів на основі нейронних мереж є ефективним і перспективним, забезпечуючи швидше навчання коефіцієнтів контролера порівняно з традиційними методами. Розроблений алгоритм адаптивного ПІД-регулятора може бути успішно застосований у різних промислових системах для покращення їх продуктивності та стабільності. Використання нейронних мереж для автоматичного налаштування параметрів

						Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

ПІД-регулятора дозволяє знизити витрати на ручне налаштування та підвищити ефективність виробничих процесів.

Для наукового та практичного використання отриманих результатів рекомендується подальше дослідження можливостей застосування методу PIDNN у більш складних системах, таких як нелінійні системи та системи з багатьма входами і виходами. Також важливо дослідити вплив різних рівнів шуму та збурень на стабільність і збіжність PIDNN за певний проміжок часу. Іншим напрямком досліджень є вивчення альтернативних методів штучного інтелекту, таких як моделювання на основі агентів, для моделювання поведінки регуляторів і оптимізації їх параметрів.

						Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Список використаної літератури

1. Огородніков С.І. Автоматичне управління та регулювання. - Київ: НАУ, 2012.
2. Князєв В.Д., Іванов А.С. Системи автоматичного керування: підручник. - Харків: ХНУРЕ, 2018.
3. Wassennan P., (1989) Neural computing: Theory and practice, Van Nostrand Reinhold Company.
4. Werbos P.I., (1994) The roots of backpropagation: From ordered derivatives to Neural Networks and political forecasting, John Wiley & Sons.
5. Lee A.Y. and Sins C.S., (1973) "Adaptive estimation and stochastic control for uncertain models", Int. J. Control, Vol. 19, No. 3, pp. 625- 639.
6. Harris C, and Brown M., (1994) Neurofuzzy adaptive Modeling and Control, Prentice Hall.
7. Ali Marzoughi, Hazlina Selamat, Mohd Fua'ad Rahmat and Herlina Abdul Rahim, "Optimized proportional integral derivative (PID) controller for the exhaust temperature control of a gas turbine system using particle swarm optimization", International Journal of the Physical Sciences Vol. 7(5), pp. 720 - 729, 30 January, 2012.
8. Katsuhiko Ogata, University of Minnesota, "Modern Control Engineering, Fourth Edition", Publisher: Aeeizb, 2002.
9. A. Jalilvand, A. Kimiyaghalam, A. Ashouri, H. Kord, "Optimal Tuning of PID Controller Parameters on a DC Motor Based on Advanced Particle Swarm Optimization Algorithm", ISSN 2077-3528, IJTPE Journal, 2011.
10. A. Zilouchian, M. Jamshidi, "Intelligent Control Systems Using Soft Computing Methodologies", by CRC Press LLC, 2001.
11. Mohammed Obaid Ali, S. P. Koh, K. H. Chong, S.K.Tiong and Zeyad Assi Obaid, "Genetic Algorithm Tuning Based PID Controller for Liquid-Level Tank System", Proceedings of the International Conference on Man-Machine Systems (ICoMMS) 11 – 13 October 2009, Batu Ferringhi, Penang, MALAYSIA, 2009.

						Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

12. N. Pillay, "A Particle Swarm Optimization Approach for Tuning of SISO PID Control Loops", Durban University of Technology, 2008.

13. A.A. Javadi, R. Farmani, T.P. Tan, "A hybrid intelligent genetic algorithm", www.elsevier.com/locate/aei, Advanced Engineering Informatics 19 (2005) 255–262, 2005.

14. Andries P. Engelbrecht, "Computational intelligence: an introduction-2nd Ed.", John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2007.

15. K.E. Parsopoulos and M.N. Vrahatis, "Recent approaches to global optimization problems through Particle Swarm Optimization", Academic Publishers. Natural Computing 1: 235–306, 2002.

16. János M'adar, János Abonyi and Ferenc Szeifert, "Interactive Particle Swarm Optimization", Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005.

17. Dian Palupi Rini, Siti Mariyam Shamsuddin and Siti Sophiyati Yuhaniz, "Particle Swarm Optimization: Technique, System and Challenges", International Journal of Computer Applications (0975 – 8887) Volume 14– No.1, January 2011.

18. A. Marzoughi, H. Selamat, M. F. Rahmat, H. Abdul Rahim, "Optimized Proportional Integral Derivative (PID) Controller for the Exhaust Temperature Control of a Gas Turbine System Using Particle Swarm Optimization", International Journal of the Physical Sciences Vol. 7(5), pp. 720-729, 2012.

19. Medsker L., Liebowitz 1., (1994) DeSign and development of expert systems and neural network, Macrnillan Press.

20. Saab S., (1995) "A Discrete time learning control algorithm for a class of linear time - invariant systems", IEEE Trans. Automatic Control, vol. 40, pp. 1138-1142.f

21. Fausett L., (1994) Fundamentals ofneural networks: Architectures. algorithms, and applications. Prentice Hall. 2005 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					60

22. Astrom KJ, (1983) "Theory and applications of adaptive control a survey,"
Automatica, 19, 471-486

23. Freeman I. A. and Skapura D.M., (1991) Neural networks algorithms,
applications and programming techniques, Addison - Wesley.

24. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers.
Trans. ASME 64, 759–768 (1942)

25. Draper, C., Li, Y.: Principles of optimizing control systems and an
application to the internal combustion engine. Optimal and Selfoptimizing Control
(1951)

26. Rumelhart, D.E., Widrow, B., Lehr, M.A.: The basic ideas in neural networks.
Commun. ACM 37, 87–92 (1994)

						Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		