

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ математики і інформатики _____
(повна назва)
Кафедра _____ Інформатики _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий(магістерський) _____

ПРОЕКТ ВЕБ-СЕРВІСУ З РОЗРАХУНКУ РАДІАЛЬНИХ
КУЛЬКОВИХ ПІДШИПНИКІВ

(тема)

Виконав:

студент 2 курсу, групи _____ МФ-62 _____

_____ Глушков Артем Андрійович _____

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика

(повна назва освітньої програми)

Керівник проф. Фролов В.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

_____ (підпис)

_____ (прізвище, ініціали)

2024 р.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	4
Вступ.....	5
1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ДЛЯ РОЗРАХУНКУ РАДІАЛЬНИХ КУЛЬКОВИХ ПІДШИПНИКІВ	9
1.1 Основні поняття та параметри.....	9
1.2 Класичний метод розрахунку ресурсу підшипника	13
1.3 Визначення еквівалентного динамічного навантаження.....	14
1.4 Розрахунок підшипника за умовами статичного навантаження....	15
1.5 Стандарти та нормативна документація підшипників.....	15
1.6 Новітні методи та інновації: Огляд сучасних підходів та нових технологій у розрахунку підшипників.....	17
1.6.1 Новітні методи та інновації: Огляд сучасних підходів та нових технологій у розрахунку підшипників.....	17
1.7 Порівняльний аналіз методів: Детальний аналіз переваг і недоліків різних методів розрахунку	18
2 Реалізація вебзастосунку.....	21
2.1 Вибір технологій та інструментів розробки.....	21
2.1.1 Вибір технологій Vue.js 3.....	21
2.1.2 Вибір технологій NestJS	22
2.1.3 Вибір технологій TypeScript	23
2.1.4 Вибір технологій MongoDB.....	24
2.1.4 Вибір технологій ml-regression.....	24
2.2 Архітектура застосунку.....	25
2.2.1 Бекенд-архітектура з NestJS.....	27
2.2.2 Модульна структура NestJS	28
2.2.3 Модульна архітектура в веб-розробці	29
2.3 Проектування бази даних.....	30
2.4 Реалізація математичної моделі	31
2.5 Реалізація фронтенду з Vue.js 3.....	33

	3
2.5.1 Компоненти інтерфейсу	33
2.6 Обґрунтування вибору середовища програмної реалізації	34
2.6.1 Інтегроване середовище розробки WebStorm.....	35
2.6.2 Додаткові інструменти та технології.....	35
3 Использование регрессионного анализа для автоматизации инженерных расчетов.....	39
3.1 Поліном: визначення та основні властивості.....	39
3.2 Сучасний стан справ в області математичного моделювання за допомогою поліномів	41
3.3 Аналіз точності поліноміальних моделей різного порядку для kt	43
3.4 Метод найменших квадратів	47
3.5 Аналіз апроксимації X та Y	48
3.6 Довірчі інтервали	53
Висновки	55
Перелік джерел посилання	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

NPM – Node Package Manager (Менеджер пакетів вузлів)

ORM – Object-Relational Mapping (Об'єктно-реляційне відображення)

DI – Dependency Injection (Ін'єкція залежностей)

IOC – Inversion of Control (Інверсія керування)

IDE – Integrated Development Environment (Інтегроване середовище розробки)

JSON – JavaScript Object Notation (Нотація об'єктів JavaScript)

BSON – Binary JSON (Бінарний JSON)

ВСТУП

Радіальні кулькові підшипники є одним із найбільш поширених типів підшипників, що використовуються в різних галузях промисловості, таких як машинобудування, енергетика, авіабудування та автомобільна промисловість. Їх правильний вибір впливає на ефективність, надійність і довговічність роботи механізмів. Разом з тим, складність технічних вимог до обладнання та велика кількість варіантів підшипників, доступних на ринку, створюють значні виклики для інженерів і конструкторів у процесі проектування та експлуатації.

Метою даної роботи є розробка веб-сервісу для автоматизованого розрахунку радіальних кулькових підшипників, який дозволить інженерам та конструкторам швидко і точно визначати необхідні параметри для різних умов експлуатації. У сучасному машинобудуванні підшипники відіграють критично важливу роль, оскільки вони забезпечують плавність руху, знижують тертя та підвищують довговічність механічних систем. Правильний вибір та розрахунок підшипників впливає на ефективність роботи обладнання, зменшує ризик виходу з ладу та знижує витрати на обслуговування.

Для досягнення поставленої мети необхідно виконати ряд завдань. По-перше, провести детальний аналіз існуючих методів і алгоритмів розрахунку радіальних кулькових підшипників. Це включає вивчення стандартів, технічних характеристик, а також особливостей різних типів підшипників. Такий аналіз дозволить визначити найефективніші підходи та виявити можливі недоліки, які можна усунути в новому сервісі.

По-друге, розробити математичну модель для розрахунку підшипників, яка враховує різні фактори навантаження та умов роботи. Це означає, що модель повинна бути достатньо універсальною, щоб застосовуватися в різних галузях промисловості, де використовуються підшипники, та враховувати специфічні вимоги до них. Важливо забезпечити точність розрахунків, щоб

отримані результати могли бути безпосередньо використані в практичних проектах.

По-третє, створити веб-інтерфейс, який забезпечить зручний доступ до сервісу та його функціоналу. Інтерфейс повинен бути інтуїтивно зрозумілим, щоб користувачі без спеціальної підготовки могли легко вводити необхідні дані та отримувати результати. Використання веб-технологій дозволить забезпечити доступність сервісу з будь-якого пристрою з інтернет-з'єднанням, що значно розширить його аудиторію.

Нарешті, необхідно провести тестування та валідацію сервісу на реальних інженерних завданнях. Це допоможе перевірити точність та надійність розробленої моделі, виявити можливі помилки та недоліки, а також отримати зворотний зв'язок від потенційних користувачів для подальшого вдосконалення сервісу.

Актуальність даної теми обумовлена зростаючою потребою в інструментах, які спрощують та автоматизують інженерні розрахунки. У контексті швидкого розвитку технологій та підвищення вимог до ефективності виробництва, можливість швидко отримувати точні розрахунки підшипників стає важливою конкурентною перевагою. Більшість існуючих рішень або є платними, або потребують встановлення складного програмного забезпечення, що може бути недоступним для малих та середніх підприємств. Створення веб-сервісу вирішує ці проблеми, надаючи доступний та ефективний інструмент широкому колу фахівців.

У ході дослідження було проведено огляд сучасних методів розрахунку підшипників, включаючи аналіз літератури та існуючих програмних рішень. Виявлено, що багато з них не враховують усіх необхідних факторів або є надто складними у використанні. Розроблена математична модель враховує як статичні, так і динамічні навантаження, вплив температури, вібрацій та інших експлуатаційних умов, що підвищує точність розрахунків.

Новизна отриманих результатів полягає в інтеграції складних інженерних розрахунків у зручний веб-сервіс, доступний з будь-якого місця та

в будь-який час. Сервіс не тільки автоматизує процес розрахунку, але й надає рекомендації щодо вибору оптимального типу підшипника, що враховує специфіку конкретного застосування. Це дозволяє інженерам швидше приймати обґрунтовані рішення, підвищуючи ефективність роботи та знижуючи ризики помилок.

Таким чином, розроблений веб-сервіс має потенціал стати цінним інструментом для інженерів та конструкторів, сприяючи розвитку галузі та впровадженню інноваційних технологій у виробничі процеси.

На сьогодні існують різні програмні продукти та онлайн-інструменти для розрахунку підшипників, такі як каталоги виробників з вбудованими калькуляторами та спеціалізовані програми для механічного проектування. Проте більшість з них мають обмежений функціонал, вимагають встановлення додаткового програмного забезпечення або не враховують специфічних умов експлуатації.

У ході роботи було розроблено веб-сервіс, який об'єднує сучасні методи розрахунку радіальних кулькових підшипників з інтуїтивним та доступним інтерфейсом. Сервіс дозволяє користувачам вводити необхідні параметри, такі як навантаження, швидкість обертання, умови експлуатації, та отримувати детальні результати розрахунків, включаючи рекомендовані типорозміри підшипників та прогнозований термін служби.

Новизна роботи полягає в поєднанні глибоких інженерних розрахунків з веб-технологіями, що забезпечує широкодоступність та зручність використання сервісу без необхідності встановлення додаткового програмного забезпечення. Крім того, розроблений сервіс враховує більш широкий спектр факторів, ніж аналогічні інструменти, що підвищує точність та надійність розрахунків.

Деякі дослідники пропонують алгоритми та моделі для більш точного прогнозування довговічності та надійності підшипників, враховуючи такі фактори, як динамічні навантаження, вібрації та температурні режими. Однак

інтеграція цих моделей у доступні та зручні для користувача інструменти залишається недостатньо розвиненою.

Підшипники є критично важливими компонентами в машинобудуванні, авіабудуванні, енергетиці та багатьох інших галузях. Їх правильний вибір і розрахунок мають суттєвий вплив на надійність і довговічність роботи обладнання. На сучасному ринку існує велика кількість моделей підшипників з різними характеристиками, що ускладнює процес їх вибору для інженерів і технічних спеціалістів.

Впровадження автоматизованого веб-сервісу для розрахунку параметрів радіальних кулькових підшипників дозволить значно спростити цей процес, зменшити ймовірність помилок та підвищити ефективність роботи спеціалістів. Крім того, використання такого сервісу може бути корисним для навчальних цілей, наприклад, для студентів технічних спеціальностей.

Проект спрямований на задоволення попиту на сучасні інструменти, що інтегрують розрахункові алгоритми, бази даних та зручний інтерфейс для взаємодії з користувачем, що робить його актуальним як для індустрії, так і для освітнього середовища.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ДЛЯ РОЗРАХУНКУ РАДІАЛЬНИХ КУЛЬКОВИХ ПІДШИПНИКІВ

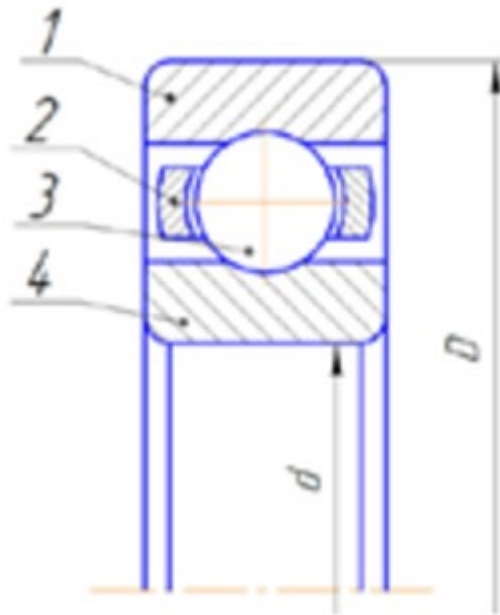
1.1 Основні поняття та параметри

Радіальні кулькові підшипники є одним із найважливіших елементів у конструкції різноманітних механічних систем, забезпечуючи підтримку обертових валів та зменшуючи тертя між рухомими частинами. Розуміння основних понять та параметрів, пов'язаних з їхньою роботою, є критично важливим для правильного розрахунку, вибору та експлуатації підшипників.

1.1.1 Конструкція радіальних кулькових підшипників

Радіальний кульковий підшипник складається з наступних основних компонентів:

1. Внутрішнє кільце. монтується на вал і обертається разом із ним. Має доріжку кочення на зовнішній поверхні.
2. Зовнішнє кільце. встановлюється у корпусі і зазвичай залишається нерухомим. Має доріжку кочення на внутрішній поверхні.
3. Тіла кочення (кульки). розташовані між внутрішнім та зовнішнім кільцями. Кульки забезпечують кочення замість ковзання, зменшуючи тертя.
4. Сепаратор (клітка). тримає кульки на рівних відстанях одна від одної, запобігаючи їх контакту та забезпечуючи рівномірний розподіл навантаження.



1 – зовнішнє кільце; 2 – сепаратор; 3 – тіла кочення; 4 – внутрішнє кільце.

Рисунок 1.1 – Конструкція підшипника кочення

1.1.2 Основні параметри підшипника



Рисунок 1.2 – Основні параметри підшипника

При розрахунку та виборі радіального кулькового підшипника враховуються наступні параметри:

1. Розміри підшипника. Внутрішній діаметр (d). діаметр отвору внутрішнього кільця, відповідає діаметру вала; Зовнішній діаметр (D). діаметр

зовнішньої поверхні зовнішнього кільця; Ширина підшипника (B) відстань між торцевими поверхнями кілець;

2. Номінальне динамічне навантаження (C). Нормоване навантаження, при якому номінальний ресурс підшипника становить 1 мільйон обертів. Визначається за стандартними методиками і залежить від геометрії підшипника, матеріалу та якості виготовлення;

3. Номінальне статичне навантаження (C0). Максимальне навантаження, яке підшипник може витримати в стані спокою без виникнення постійних деформацій в точках контакту. Важливе при розрахунку підшипників, що піддаються великим статичним або ударним навантаженням.

4. Еквівалентне динамічне навантаження (P). Умовне постійне навантаження, еквівалентне фактичному змінному навантаженню з точки зору впливу на довговічність підшипника. Розраховується з урахуванням фактичних радіальних та осьових навантажень.

5. Ресурс підшипника (L10). Статистичний показник, який відображає очікуваний термін служби підшипника до появи перших ознак втомного руйнування. Визначається як кількість обертів, яку витримує 90% підшипників при заданих умовах навантаження.

6. Коефіцієнти навантаження (X та Y). Коефіцієнти, що використовуються при розрахунку еквівалентного навантаження, залежать від типу підшипника та співвідношення між радіальним та осьовим навантаженнями.

7. Швидкість обертання (n). Частота обертання вала, виражена в обертах за хвилину (об/хв). Впливає на розрахунок ресурсу підшипника в годинах та на вибір мастила.

8. Коефіцієнт надійності (a1). Враховує необхідний рівень надійності підшипника. Базова надійність складає 90%, але в деяких випадках потрібен вищий рівень надійності, що враховується через цей коефіцієнт.

9. Коефіцієнт умов роботи (a_2). Враховує вплив специфічних умов експлуатації, таких як якість змащення, температура, забруднення, вібрації тощо.

10. Коефіцієнт матеріалу (a_3). Враховує особливості матеріалу підшипника, його якість та характеристики.

1.1.3 Навантаження на підшипник

Розрахунок підшипника передбачає точне визначення діючих на нього навантажень.

1. Радіальне навантаження (F_r). Сила, що діє перпендикулярно до осі обертання. Може виникати внаслідок ваги ротора, центробіжних сил, сил різання тощо.

2. Осьове навантаження (F_a). Сила, що діє вздовж осі обертання. Може бути результатом дії гвинтових передач, теплових розширень, монтажних похибок.

1.1.3 Еквівалентне навантаження та його розрахунок

Еквівалентне динамічне навантаження визначається за формулою:

$$P = XF_r + YF_a \quad (1.1)$$

де X – коефіцієнт радіального навантаження;

Y – коефіцієнт осьового навантаження;

Коефіцієнти X та Y залежать від відношення осьового до радіального навантаження (F_a/F_r) та кута контакту підшипника. Для радіальних кулькових підшипників з нормальним кутом контакту (0°) стандартні значення такі:

Якщо $F_a/F_r \leq e$

$$X = 1, Y = 0 \quad (1.2)$$

Якщо $Fa/Fr > e$

$$X = 0.56, Y = 1.6 \quad (1.3)$$

де e – коефіцієнт, який залежить від конструкції підшипника;

1.2 Класичний метод розрахунку ресурсу підшипника

Класичний метод розрахунку ресурсу підшипника базується на емпіричних формулах, розроблених для оцінки довговічності підшипників при заданих умовах експлуатації. Один із найвідоміших та найчастіше використовуваних підходів у цій сфері — формула Лундберга-Пальмгрена, яка дозволяє визначити номінальний ресурс підшипника L_{10} . Цей метод був запропонований норвезькими інженерами Еріхом Лундбергом та Вернером Пальмгренем у середині XX століття і залишається основою для багатьох сучасних розрахунків ресурсів підшипників.

1.2.1 Теоретичні основи формули Лундберга-Пальмгрена

Формула Лундберга-Пальмгрена виражає залежність між ресурсом підшипника L_{10} , номінальним динамічним навантаженням C та еквівалентним динамічним навантаженням P . Вона формулюється наступним чином:

$$L_{10} = \left(\frac{C}{P} \right)^p \quad (1.4)$$

де L_{10} — номінальний ресурс підшипника, що означає кількість обертів, яку витримає 90% підшипників без втомних руйнувань;

C — номінальне динамічне навантаження, яке підшипник може витримати при ресурсі L_{10} ;

P — еквівалентне динамічне навантаження, що враховує реальні умови навантаження;

p — показник степеня, який для кулькових підшипників.

Ця формула припускає, що ресурс підшипника обертовий і що навантаження на нього є постійним та радіальним.

1.3 Визначення еквівалентного динамічного навантаження

Еквівалентне динамічне навантаження P для радіальних кулькових підшипників визначається за формулою:

$$P = XF_r + YF_a \quad (1.5)$$

де F_r — радіальне навантаження (Н);

F_a — осьове навантаження (Н);

X та Y — коефіцієнти, що залежать від співвідношення F_a / F_r та конструкції підшипника.

Для чисто радіального навантаження ($F_a=0$):

$$P = F_r \quad (1.6)$$

Коефіцієнти X та Y визначаються за таблицями, наведеними в стандартах або каталогах виробників.

1.4 Розрахунок підшипника за умовами статичного навантаження

Перевірка підшипника на статичну міцність здійснюється за допомогою коефіцієнта статичної безпеки S_0 :

$$S_0 = \frac{C_0}{P_0} \quad (1.7)$$

де C_0 – номінальне статичне навантаження (Н);

P_0 – еквівалентне статичне навантаження (Н).

Еквівалентне статичне навантаження визначається за формулою:

$$P_0 = X_0 F_r + Y_0 F_a \quad (1.8)$$

де X_0, Y_0 – коефіцієнти для статичного розрахунку.

1.5 Стандарти та нормативна документація підшипників

У процесі розробки, вибору та розрахунку радіальних кулькових підшипників особливу увагу слід приділяти стандартам та нормативним документам, які регулюють вимоги до їхніх характеристик, виготовлення та експлуатації. Дотримання цих стандартів є критично важливим для забезпечення надійності, безпеки та взаємозамінності підшипників у різних галузях промисловості.

Одним із ключових міжнародних стандартів у сфері підшипникової техніки є стандарти Міжнародної організації зі стандартизації ISO. Стандарти ISO встановлюють загальні вимоги до розмірів, допусків, матеріалів та методів випробувань підшипників кочення. Зокрема, стандарт ISO 15 визначає основні

граничні розміри та маркування радіальних кулькових підшипників, що забезпечує їхню взаємозамінність незалежно від виробника та країни походження.

Стандарт ISO 281 надає методику розрахунку номінального ресурсу підшипників кочення з урахуванням навантажень, умов експлуатації та надійності. Цей стандарт є основою для інженерних розрахунків довговічності підшипників і дозволяє оцінити їхній термін служби в конкретних умовах роботи.

Для розрахунку статичних навантажень та перевірки підшипників на статичну міцність застосовується стандарт ISO 76, який встановлює методи визначення еквівалентного статичного навантаження та номінального статичного навантаження підшипників кочення. Він допомагає запобігти перевантаженням, що можуть призвести до деформацій та передчасного виходу з ладу підшипників.

У сфері точності та допусків розмірів підшипників важливим є стандарт ISO 492, який визначає класи точності підшипників кочення та відповідні допуски на їхні геометричні параметри. Це особливо актуально для високошвидкісних або високоточних застосувань, де відхилення від номінальних розмірів можуть суттєво вплинути на роботу механізму.

Важливість дотримання стандартів та нормативної документації полягає в наступному:

1. Забезпечення якості та надійності. Стандарти встановлюють мінімальні вимоги до характеристик підшипників, що гарантує їхню відповідність вимогам безпеки та експлуатаційним параметрам.

2. Взаємозамінність та уніфікація. Уніфіковані розміри та допуски дозволяють використовувати підшипники різних виробників без необхідності внесення конструктивних змін у обладнання.

3. Оптимізація процесів проектування та виробництва. Стандартизовані методики розрахунку та вимоги спрощують інженерні розрахунки та скорочують час на розробку нових виробів.

4. Міжнародна співпраця та торгівля. Визнання міжнародних стандартів полегшує експорт та імпорт підшипникової продукції, сприяючи розвитку міжнародних економічних відносин.

У сучасних умовах, коли технології швидко розвиваються, стандарти також регулярно оновлюються та доповнюються, відображаючи нові досягнення науки та техніки. Тому фахівцям у галузі машинобудування важливо постійно стежити за актуальністю нормативної бази та впроваджувати нові стандарти у практику.

1.6 Новітні методи та інновації: Огляд сучасних підходів та нових технологій у розрахунку підшипників

У сфері розрахунку радіальних кулькових підшипників постійно з'являються нові методи та інноваційні технології, що сприяють підвищенню точності, ефективності та надійності цих компонентів. Сучасні підходи охоплюють як теоретичні, так і практичні аспекти, використовуючи передові інструменти та технології для покращення процесу розрахунку та експлуатації підшипників.

1.6.1 Новітні методи та інновації: Огляд сучасних підходів та нових технологій у розрахунку підшипників

Комп'ютерне моделювання стало невід'ємною частиною інженерних розрахунків, включаючи розрахунок підшипників. Методи кінцевих елементів (МКЕ) дозволяють детально аналізувати напружено-деформаційний стан підшипника під різними умовами навантаження. Це сприяє більш точному визначенню меж експлуатації та прогнозуванню ресурсів підшипника.

Приклади застосування:

1. Прогнозування виходу підшипника з ладу на основі історичних даних.
2. Оптимізація параметрів підшипника для конкретних умов експлуатації.
3. Адаптивне управління технічним обслуговуванням.

Переваги:

1. Збільшення точності прогнозів.
2. Можливість автоматизації процесів аналізу.
3. Підвищення ефективності експлуатації обладнання.

Недоліки:

1. Висока складність впровадження.
2. Необхідність наявності великих обсягів даних для навчання моделей.

1.7 Порівняльний аналіз методів: Детальний аналіз переваг і недоліків різних методів розрахунку

Для ефективного вибору методу розрахунку радіальних кулькових підшипників важливо провести порівняльний аналіз існуючих підходів, оцінюючи їхні переваги та недоліки. Це дозволяє визначити найбільш підходящий метод для конкретних умов експлуатації та вимог до підшипника.

Класичний метод розрахунку ресурсу підшипника, запропонований Бренданом, Гілфордом та іншими, базується на статистичному аналізі даних експлуатації підшипників. Він використовує концепцію еквівалентного динамічного навантаження для визначення ресурсу підшипника. Основними перевагами цього методу є його простота застосування, зрозумілість формул, широке використання та підтвердженість практикою, а також відсутність потреби в складних обчислювальних ресурсах. Проте, класичний метод має

обмежену точність при нестандартних умовах навантаження, не враховує вплив таких факторів, як температура та вібрації, і найбільш підходить для статичних або циклічних навантажень без значних коливань.

Методи кінцевих елементів (МКЕ) дозволяють моделювати поведінку підшипника під різними умовами навантаження, враховуючи складні геометричні та матеріальні властивості. Ці методи характеризуються високою точністю розрахунків, можливістю врахування складних умов експлуатації та геометрій, а також дозволяють проводити детальний аналіз напружень та деформацій. Однак, методи кінцевих елементів вимагають значних обчислювальних ресурсів, спеціалізованого програмного забезпечення та знань, а також є часозатратними через складність процесу моделювання та аналізу.

Методи машинного навчання використовують великі обсяги даних для навчання моделей, які можуть прогнозувати ресурс підшипника або виявляти аномалії в його роботі. Серед їхніх переваг – здатність обробляти великі обсяги даних та виявляти складні закономірності, висока точність прогнозів при достатній кількості даних, а також можливість автоматизації процесів аналізу та прогнозування. Недоліками є високі вимоги до обсягу та якості даних, необхідність експертизи в галузі штучного інтелекту та машинного навчання для розробки моделей, а також ризик перенавчання моделей та недостатньої узагальнюваності.

Метод статистичного аналізу використовує ймовірнісні підходи для оцінки ресурсів підшипників, базуючись на статистичних даних експлуатації. Цей метод простий у застосуванні та розумінні, дозволяє оцінювати ймовірнісні характеристики ресурсу і має широкий спектр застосувань у різних галузях. Проте, його точність обмежена при відсутності достатніх даних, він не враховує складні взаємодії між різними факторами навантаження і найкраще підходить для стандартних умов експлуатації.

Інтегровані методи поєднують кілька підходів, наприклад, методи кінцевих елементів та машинного навчання, для досягнення більш високої

точності та надійності розрахунків. Перевагами таких методів є поєднання сильних сторін різних підходів, підвищена точність та надійність результатів, а також можливість врахування складних умов експлуатації. Однак, інтегровані методи відзначаються високою складністю впровадження та обробки даних, потребують використання складних інструментів та методологій, а також вимагають значних витрат часу та ресурсів на розробку інтегрованих моделей.

Метод	Переваги	Недоліки
Класичний метод	Простота, зрозумілість, широке застосування	Обмежена точність, неврахування деяких факторів
Метод кінцевих елементів	Висока точність, можливість моделювання складних умов	Високі вимоги до ресурсів, складність застосування
Метод машинного навчання	Обробка великих даних, висока точність прогнозів	Високі вимоги до даних, складність розробки моделей
Статистичний аналіз	Простота, можливість оцінки ймовірнісних характеристик	Обмежена точність, неврахування складних взаємодій
Інтегровані методи	Поєднання переваг різних методів, висока точність	Складність впровадження, високі витрати на розробку

Таблиця 1.1 Порівняння методів

Таким чином, вибір методу розрахунку ресурсу підшипника залежить від конкретних умов експлуатації, наявних ресурсів та вимог до точності та надійності розрахунків. Класичні методи залишаються популярними завдяки своїй простоті та перевірності, тоді як сучасні підходи, такі як методи кінцевих елементів та машинного навчання, пропонують більш детальний та точний аналіз, але вимагають більших ресурсів та спеціалізованих знань.

2 РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

2.1 Вибір технологій та інструментів розробки

У процесі розробки веб-сервісу для розрахунку радіальних кулькових підшипників було здійснено ретельний аналіз доступних технологій та інструментів з метою вибору оптимального стеку, який забезпечить високу продуктивність, масштабованість, зручність розробки та подальшої підтримки застосунку. Враховуючи вимоги до функціональності, інтерфейсу користувача, обробки даних та математичного моделювання, були обрані наступні технології та інструменти:

1. Vue.js 3;
2. NestJS;
3. TypeScript;
4. MongoDB;
5. ml-regression.

2.1.1 Вибір технологій Vue.js 3

Vue.js 3 — це сучасний прогресивний JavaScript-фреймворк для створення користувацьких інтерфейсів. Він забезпечує реактивність даних, компонентний підхід до розробки та легкість інтеграції з іншими бібліотеками або існуючими проектами. Переваги використання Vue.js 3:

1. Реактивність та продуктивність. Нова версія Vue.js використовує Composition API та новий механізм реактивності, що підвищує продуктивність застосунку та полегшує керування станом компонентів;
2. Компонентна архітектура. Забезпечує повторне використання коду, спрощує тестування та підтримку застосунку;

3. Низький поріг входу. Простий синтаксис та відмінна документація роблять Vue.js доступним для розробників різного рівня;

4. Активна спільнота та екосистема. Велика кількість плагінів, бібліотек та інструментів підтримують розвиток проекту.

Для фронтенд-частини веб-сервісу необхідно було вибрати фреймворк, який забезпечить інтуїтивний та інтерактивний інтерфейс користувача з високою продуктивністю та простотою розробки. Vue.js 3 відповідає цим вимогам завдяки своїй реактивності та компонентному підходу. Крім того, його легка інтеграція з іншими технологіями та можливість використання TypeScript роблять його ідеальним вибором для цього проекту.

2.1.2 Вибір технологій NestJS

NestJS — це прогресивний Node.js фреймворк для створення ефективних, надійних та масштабованих серверних додатків. Він побудований на основі Express.js та використовує TypeScript, що додає йому переваг у типізації та структурі коду. Переваги використання NestJS:

1. Модульна архітектура. Створення застосунків за принципом модулів, контролерів та сервісів сприяє масштабованості та підтримованості коду;

2. Використання TypeScript. Забезпечує статичну типізацію, що допомагає уникати багатьох помилок на етапі компіляції;

3. Інверсія управління (IoC) та ін'єкція залежностей (DI). Спрощує тестування та підвищує гнучкість застосунку;

4. Вбудована підтримка бібліотек та інструментів. Легка інтеграція з MongoDB, підтримка WebSockets, GraphQL та інших технологій.

Бекенд веб-сервісу потребує потужного та масштабованого фреймворку, який забезпечить структуровану архітектуру та спростить розробку складної бізнес-логіки. NestJS відповідає цим вимогам завдяки своїй

модульності та використанню TypeScript. Крім того, він забезпечує високий рівень абстракції, що дозволяє зосередитися на бізнес-логіці, а не на низькорівневих деталях реалізації.

2.1.3 Вибір технологій TypeScript

TypeScript — це надмножина JavaScript, яка додає статичну типізацію та об'єктно-орієнтовані можливості до мови. Це дозволяє писати більш надійний та підтримуваний код, який легше масштабувати та тестувати. Переваги використання TypeScript:

1. Статична типізація. Допомогає виявляти помилки на ранніх етапах розробки;
2. Покращене автодоповнення та рефакторинг. Підвищує продуктивність розробників за рахунок кращої інтеграції з IDE;
3. Модульність та об'єктно-орієнтовані можливості. Спрощує структурування великих проектів;
4. Сумісність з JavaScript. Можливість поступового переходу з JavaScript на TypeScript.

Використання TypeScript як на фронтенді, так і на бекенді забезпечує узгодженість коду та полегшує спільну розробку. Статична типізація зменшує кількість помилок, покращує якість коду та полегшує підтримку великого проекту. Це особливо важливо для проектів, які містять складні алгоритми та моделі, як у випадку з математичними розрахунками підшипників.

2.1.4 Вибір технологій MongoDB

MongoDB — це документно-орієнтована нереляційна база даних NoSQL, яка зберігає дані у форматі BSON (бінарний JSON). Вона забезпечує гнучкість у моделюванні даних та високу продуктивність при роботі з великими обсягами інформації. Переваги використання MongoDB:

1. Гнучкість у структурі даних. Можливість зберігати документи зі змінною структурою, що полегшує еволюцію схеми бази даних;
2. Висока продуктивність та масштабованість. Підтримка горизонтального масштабування та розподіленого зберігання даних;
3. Легка інтеграція з Node.js. Наявність офіційного драйвера та підтримка ORM-бібліотек, таких як Mongoose;
4. Потужні можливості для запитів та агрегації даних. Дозволяє виконувати складні запити та обробку даних безпосередньо на рівні бази даних.

Для зберігання інформації про параметри підшипників, результати розрахунків та користувачів необхідна база даних, яка забезпечує гнучкість та високу продуктивність. MongoDB відповідає цим вимогам завдяки своїй документно-орієнтованій моделі, яка дозволяє легко змінювати та розширювати структуру даних без необхідності складних міграцій. Крім того, вона добре інтегрується з NestJS та TypeScript через бібліотеку Mongoose, що спрощує роботу з базою даних.

2.1.4 Вибір технологій ml-regression

ml-regression — це легка бібліотека машинного навчання для Node.js, яка надає реалізації різних алгоритмів регресії, таких як лінійна регресія, багаточленна регресія та інші. Переваги використання ml-regression:

1. Простота використання. Легко інтегрується в Node.js додатки, має простий API;
2. Підтримка різних алгоритмів регресії. Дозволяє вибирати оптимальний метод для конкретного завдання;
3. Відсутність залежностей від зовнішніх модулів: Полегшує розгортання та підтримку застосунку;
4. Відкрите вихідне код. Можливість модифікації та адаптації під потреби проекту.

Математичне моделювання розрахунків підшипників вимагає використання алгоритмів регресії для прогнозування ресурсу та інших параметрів на основі вхідних даних. ml-regression надає необхідні інструменти для реалізації цих алгоритмів без зайвих складнощів та дозволяє інтегрувати їх безпосередньо у серверну частину на Node.js. Це усуває потребу в використанні окремих сервісів або мов програмування для машинного навчання, таких як Python, і спрощує архітектуру застосунку.

2.2 Архітектура застосунку

Архітектура веб-застосунку для розрахунку радіальних кулькових підшипників базується на клієнт-серверній моделі, яка забезпечує чіткий розподіл функціональності між фронтендом та бекендом. Такий підхід дозволяє незалежно розробляти, тестувати та масштабувати кожну частину застосунку.

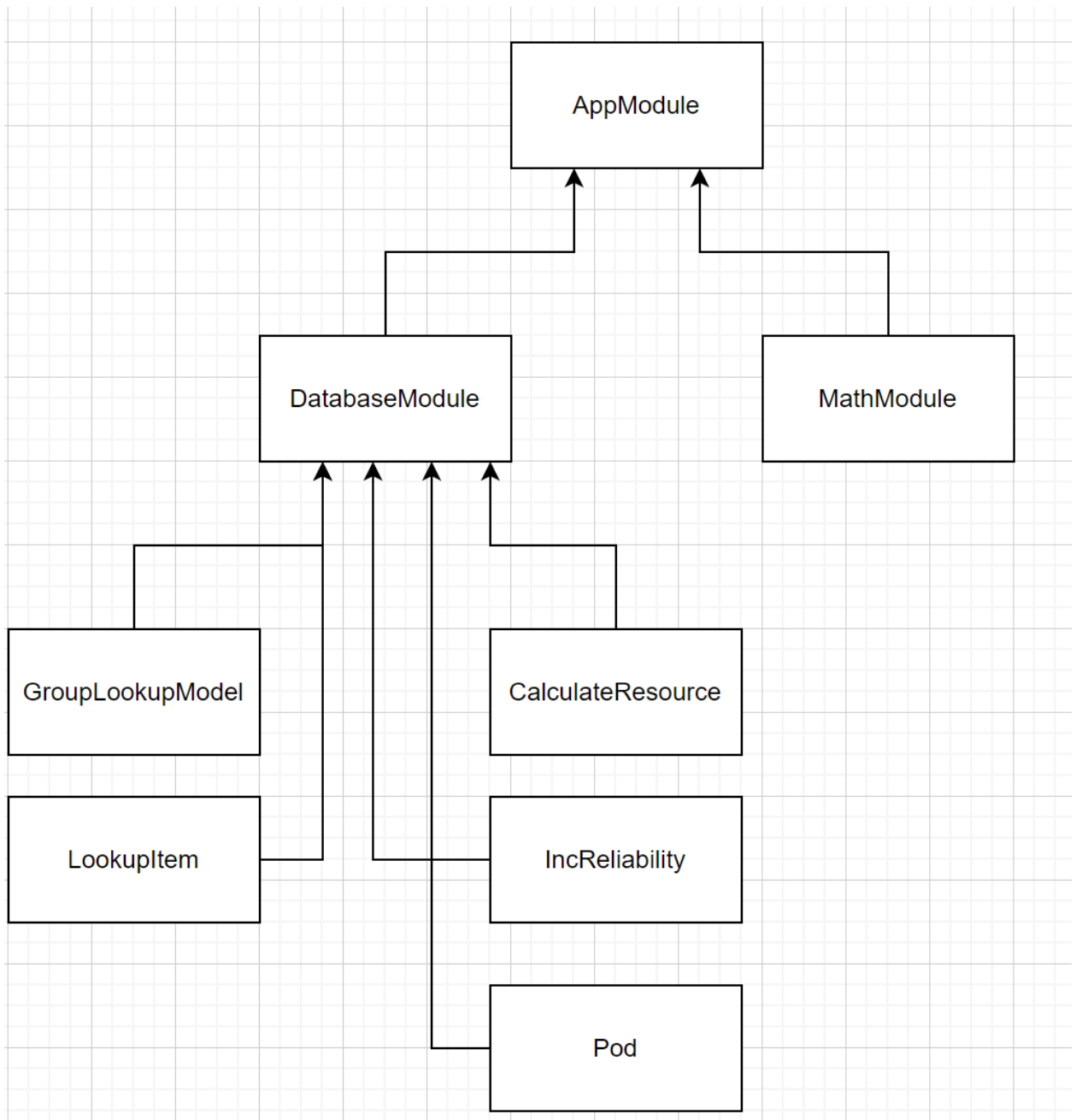


Рисунок 2.1 – діаграма зв'язків модулів

На поданій діаграмі представлена структурна схема взаємодії модулів у програмному забезпеченні. Вона демонструє взаємозв'язки між окремими компонентами системи, підкреслюючи модульність та ієрархію архітектури. У верхній частині схеми розташований основний модуль AppModule, який виступає як головний координатор або точка входу в систему. До нього підключені два важливі підсистеми: MathModule і DatabaseModule.

MathModule відповідає за математичні обчислення або операції, що забезпечують функціонування алгоритмів, які застосовуються в інших компонентах системи.

DatabaseModule, у свою чергу, є центральною складовою для роботи з даними. Він виконує функції збереження, отримання та обробки даних для інших компонентів. Усі ключові модулі нижчого рівня взаємодіють із DatabaseModule, що підкреслює його важливість у загальній архітектурі.

Серед компонентів, які залежать від DatabaseModule, виділяються такі:

1. GroupLookupModel, що забезпечує пошук і управління групами даних.
2. CalculateResource, таблиця в бд відповідає за розрахунковий ресурс
3. IncReliability – таблиця в бд відповідає за коефіцієнти для розрахунку підвищеної надійності

2.2.1 Бекенд-архітектура з NestJS

Бекенд застосунку реалізований на основі фреймворку NestJS, який використовує модульну архітектуру та забезпечує чітку організацію коду. Основні принципи бекенд-архітектури:

1. Модульність. Розбиття застосунку на логічні модулі, кожен з яких відповідає за певну функціональність (наприклад, модуль розрахунків, модуль користувачів, модуль підшипників).
2. Контролери. Відповідають за обробку вхідних HTTP-запитів, маршрутизують їх до відповідних сервісів та повертають відповіді клієнту.
3. Сервіси. Містять бізнес-логіку застосунку, взаємодіють з базою даних, виконують розрахунки та обробку даних.

4. Ін'єкція залежностей. Використання вбудованого механізму ін'єкції залежностей для спрощення тестування та управління залежностями між компонентами.

5. Об'єктно-реляційне відображення (ORM). Використання бібліотеки Mongoose для роботи з MongoDB, що дозволяє моделювати структуру даних та виконувати CRUD-операції.

2.2.2 Модульна структура NestJS

Відповідно до документації NestJS, застосунок організований за принципом модуль-контролер-сервіс, що забезпечує масштабованість та підтримуваність коду. Модулі є основними будівельними блоками застосунку. Вони об'єднують пов'язані компоненти та визначають область видимості залежностей. Основні модулі застосунку:

1. AppModule. Головний модуль, який імпортує всі інші модулі та налаштовує глобальні залежності.
2. DatabaseModule. Модуль, який налаштовує підключення до бази даних MongoDB та надає доступ до моделей.
3. MathModule. Модуль який відповідає за математичні розрахунки

Контролери обробляють HTTP-запити, приймають вхідні дані від клієнта та повертають відповіді. Вони визначають маршрути та типи запитів (GET, POST, PUT, DELETE) і взаємодіють із відповідними сервісами.

Сервіси містять бізнес-логіку та здійснюють обробку даних. Вони взаємодіють з базою даних через моделі та можуть використовувати інші сервіси.

2.2.3 Модульна архітектура в веб-розробці

Основою архітектури NestJS є модулі, які дозволяють структурно організувати додаток на логічно пов'язані блоки. Кожен модуль відповідає за певну функціональність і може містити контролери, сервіси та інші провайдери. Такий підхід сприяє кращій організації коду, полегшує його підтримку та масштабування, оскільки додаток можна легко розширювати, додаючи нові модулі без впливу на існуючі.

Контролери в NestJS відповідають за обробку вхідних HTTP-запитів та повернення відповідей клієнтам. Вони визначають маршрути та зв'язують запити з відповідними сервісами, які містять бізнес-логіку додатка. Це дозволяє розділити відповідальності між різними компонентами та забезпечує чисту архітектуру, де кожен елемент виконує свою чітко визначену роль.

Сервіси, або провайдери, є ключовими компонентами, які містять основну бізнес-логіку додатка. Вони можуть бути інжектовані в інші компоненти завдяки системі впровадження залежностей (Dependency Injection), яка є однією з сильних сторін NestJS. Це дозволяє легко керувати залежностями між класами та сприяє створенню гнучкого та тестованого коду.

Впровадження залежностей в NestJS полегшує процес тестування та заміни компонентів, оскільки залежності можуть бути легко замінені на макети або інші реалізації. Це особливо корисно при розробці великих додатків, де необхідно забезпечити високий рівень надійності та підтримуваності коду.

Декоратори є ще однією важливою особливістю NestJS. Вони використовуються для додавання метаданих до класів, методів та властивостей, що робить код більш декларативним та читабельним. З їх допомогою можна визначати маршрути, впроваджувати залежності та налаштовувати різні аспекти додатка без необхідності писати додатковий код.

Крім основних компонентів, NestJS підтримує використання проміжного програмного забезпечення (middleware), гвардів (guards),

інтерсепторів (interceptors) та фільтрів виключень (exception filters). Це надає розробникам додаткові інструменти для контролю та модифікації потоку запитів та відповідей. Наприклад, проміжне ПЗ може використовуватися для автентифікації або логування запитів, гварди дозволяють реалізувати авторизацію та контроль доступу, а інтерсептори можуть модифікувати або трансформувати дані перед їх поверненням клієнту.

Фільтри виключень дозволяють централізовано обробляти помилки та виключення, що виникають під час виконання запитів. Це забезпечує кращий користувацький досвід, оскільки користувачі отримують зрозумілі та структуровані повідомлення про помилки, а не стандартні системні повідомлення.

2.3 Проектування бази даних

При розробці серверного додатка важливо обрати відповідну базу даних, яка забезпечить ефективне зберігання та доступ до необхідної інформації. У цьому проекті було обрано MongoDB завдяки її гнучкості у роботі з нереляційними даними та простоті інтеграції з додатками на базі Node.js.

MongoDB є документно-орієнтованою базою даних, яка зберігає дані у форматі BSON (бінарний JSON). Це дозволяє працювати зі складними та динамічними структурами даних без необхідності суворого визначення схеми. Такий підхід є особливо корисним при зберіганні інформації про параметри підшипників, результати розрахунків та дані користувачів, де структура даних може змінюватися або розширюватися з часом.

Для взаємодії з MongoDB у середовищі NestJS використовується бібліотека Mongoose, яка забезпечує можливості Object Data Modeling (ODM). Mongoose дозволяє визначати схеми даних, створювати моделі та надає

зручний API для виконання CRUD (Create, Read, Update, Delete) операцій. Це спрощує процес розробки та забезпечує високу продуктивність додатка.

Хоча MongoDB є нереляційною базою даних, важливо забезпечити логічні зв'язки між даними. Для цього використовуються посилання між документами за допомогою ідентифікаторів. Наприклад, документ розрахунку містить посилання на відповідний підшипник та користувача. Це дозволяє легко отримувати пов'язані дані при виконанні запитів та підтримувати цілісність інформації.

З метою підвищення продуктивності було впроваджено індекси на ключових полях, що часто використовуються у запитах. Це дозволяє зменшити час виконання пошукових операцій та забезпечує швидкий доступ до даних навіть при збільшенні обсягу інформації.

2.4 Реалізація математичної моделі

Математична модель є основою для проведення розрахунків параметрів підшипників та прогнозування їх ресурсу. У попередніх розділах було детально розроблено цю модель, яка враховує різні фактори, такі як навантаження, швидкість обертання, температура та інші експлуатаційні умови.

Для практичної реалізації математичної моделі було обрано бібліотеку `ml-regression`, яка надає широкий спектр алгоритмів регресії. Зокрема, ми вирішили використовувати поліноміальну регресію, яка дозволяє моделювати нелінійні залежності між вхідними параметрами та прогнозованим ресурсом підшипника.

Поліноміальна регресія є розширенням лінійної регресії і дозволяє моделювати складніші, нелінійні взаємозв'язки між змінними. У контексті нашого проекту це особливо важливо, оскільки залежність між технічними характеристиками підшипників та їхнім ресурсом не завжди є лінійною.

Використовуючи поліноміальну регресію, ми можемо врахувати вплив вищих степенів вхідних змінних, що підвищує точність прогнозу. Це дозволяє більш точно відображати реальні процеси зношування та зносу підшипників в умовах експлуатації. Процес розробки моделі:

1. Збір та підготовка даних. Ми зібрали великий обсяг даних про різні типи підшипників, їх технічні характеристики та фактичний термін служби. Дані були очищені від аномалій та пропусків, нормалізовані та підготовлені для використання в алгоритмі поліноміальної регресії. Вибір алгоритму. На основі аналізу даних було обрано найбільш підходящий алгоритм регресії, який забезпечує оптимальний баланс між складністю моделі та точністю прогнозу.

2. Вибір ступеня поліному. Одним з ключових етапів було визначення оптимального ступеня поліному. Занадто високий ступінь може призвести до перенавчання моделі, тоді як занадто низький — не врахувати важливі залежності. Після серії експериментів та валідації було обрано ступінь поліному, який забезпечує найкращий баланс між складністю моделі та її узагальнювальною здатністю.

3. Навчання моделі. За допомогою бібліотеки `ml-regression` та обраного ступеня поліному ми навчили модель на підготовлених даних. Алгоритм підганяє поліноміальну функцію до даних таким чином, щоб мінімізувати помилку прогнозу.

4. Тестування та валідація. Модель була протестована на тестовому наборі даних, що не використовувався під час навчання. Ми оцінили точність прогнозів за допомогою метрик, таких як середньоквадратична помилка та коефіцієнт детермінації. Результати показали високу точність та надійність моделі.

Після успішної розробки та тестування модель була інтегрована у сервісну частину бекенду. Це дозволяє автоматично виконувати розрахунки при запитах від клієнтської частини додатка. Користувач вводить необхідні

параметри, система передає ці дані до моделі, яка повертає прогнозований ресурс підшипника.

2.5 Реалізація фронтенду з Vue.js 3

У сучасних веб-додатках фронтенд відіграє ключову роль у забезпеченні зручного та інтуїтивно зрозумілого інтерфейсу користувача. У нашому проєкті для розробки клієнтської частини було обрано фреймворк Vue.js 3. Цей фреймворк поєднує в собі високу продуктивність, гнучкість та простоту використання, що робить його ідеальним вибором для створення інтерактивних та масштабованих веб-додатків.

Фронтенд забезпечує інтуїтивний інтерфейс для введення вхідних даних, відображення результатів розрахунків та взаємодії з користувачем. Основна мета — створити зручний та функціональний інструмент, який дозволить користувачам легко виконувати необхідні дії та отримувати потрібну інформацію.

2.5.1 Компоненти інтерфейсу

Для побудови інтерфейсу були розроблені наступні основні компоненти:

1. `App.vue`. Це головний компонент на якому будується весь вебзастосунок
2. `NumberLine.vue`. Це компонент який являється вхідним полем для числових значень
3. `LookupLine`. Це компонент для відображення довідкових значень
4. `GroupLookupLine`. Це компонент для відображення групованих довідкових значень

5. `TextAreaLine.vue`. Цей компонент відповідає за багаторядкове поле.

2.6 Обґрунтування вибору середовища програмної реалізації

При розробці програмного забезпечення надзвичайно важливо обрати оптимальне середовище програмування, яке забезпечить ефективність роботи, зручність та продуктивність. У рамках даного дипломного проєкту було прийнято рішення використовувати операційну систему Windows та інтегроване середовище розробки WebStorm від компанії JetBrains. Цей вибір обумовлений низкою факторів, які сприяли успішній реалізації поставлених завдань.

Операційна система Windows є однією з найпоширеніших ОС у світі. Її використання у проєкті має декілька переваг:

1. Широка підтримка програмного забезпечення. Windows сумісна з великою кількістю інструментів та середовищ розробки, що дозволяє легко інтегрувати різні технології.
2. Зручний інтерфейс користувача. інтуїтивний графічний інтерфейс спрощує навігацію та управління системою, що особливо важливо при роботі з багатьма програмами одночасно.
3. Підтримка апаратного забезпечення. Windows добре працює з різноманітним апаратним обладнанням, забезпечуючи стабільну та продуктивну роботу системи.
4. Спільнота та підтримка. велика кількість користувачів Windows означає наявність розширеної спільноти, де можна знайти рішення для різних технічних питань.

2.6.1 Інтегроване середовище розробки WebStorm

WebStorm було обрано як основне інтегроване середовище розробки (IDE) для роботи над проєктом з наступних причин:

1. Потужний редактор коду. WebStorm підтримує широкий спектр мов програмування та фреймворків, включаючи JavaScript, TypeScript, Vue.js 3, Node.js та NestJS, що є основними технологіями у проєкті.
2. Інтелектуальні функції. автоматичне завершення коду, рефакторинг, навігація по коду та вбудовані інструменти аналізу допомагають писати чистий та якісний код, зменшуючи кількість помилок.
3. Вбудована підтримка систем контролю версій. інтеграція з Git дозволяє ефективно керувати версіями коду, здійснювати команди, злиття та вирішення конфліктів безпосередньо в IDE.
4. Налагодження та тестування. WebStorm надає зручні інструменти для налагодження додатків, встановлення точок зупину, перегляду змінних та виконання юніт-тестів.
5. Розширюваність. можливість встановлення додаткових плагінів та тем дозволяє налаштувати середовище під власні потреби та підвищити продуктивність роботи.

2.6.2 Додаткові інструменти та технології

Окрім основного середовища розробки, у проєкті використовувалися наступні інструменти:

1. Node.js. серверне середовище виконання, яке дозволяє запускати JavaScript-код поза браузером. Використання Node.js забезпечує високу продуктивність та масштабованість серверної частини додатка.

2. Npm. менеджери пакетів, які спрощують встановлення та оновлення залежностей проєкту, забезпечуючи стабільність та контроль версій використовуваних бібліотек.

3. MongoDB Compass. графічний інтерфейс для роботи з базою даних MongoDB, який полегшує візуалізацію даних, виконання запитів та керування колекціями.

4. Postman. інструмент для тестування API, який дозволяє відправляти HTTP-запити до серверу, перевіряти відповіді та налагоджувати взаємодію між клієнтом і сервером.

5. GitHub. платформа для хостингу репозиторіїв, спільної роботи над кодом та відстеження проблем (issues). Використання GitHub сприяє організованій командній роботі та зберіганню історії проєкту.

Вибір Rest API для взаємодії між клієнтом і сервером обґрунтований наступними причинами: Rest API є загальноприйнятим і широко використовуваним стандартом для створення веб-сервісів. Він забезпечує простоту та зручність у взаємодії завдяки використанню протоколу HTTP та стандартних методів передачі даних. Rest API дозволяє обмінюватися інформацією у популярних форматах, таких як JSON або XML, що спрощує обробку та передачу даних між клієнтом і сервером. Цей підхід сприяє гнучкому розподілу функціональності між клієнтом і сервером, що полегшує масштабування системи та розвиток окремих компонентів незалежно один від одного. Крім того, Rest API підтримує кешування результатів запитів, що допомагає знизити навантаження на сервер і покращити продуктивність системи. Використання Rest API дозволяє забезпечити взаємодію з різними типами клієнтів, включаючи веб-додатки, мобільні додатки, IoT-пристрої та інші, що робить його універсальним і гнучким рішенням для комунікації між клієнтом і сервером. Таким чином, Rest API є популярним вибором для реалізації взаємодії між клієнтом і сервером завдяки своїй стандартності,

простоті використання, можливості масштабування та підтримці різноманітних типів клієнтів.

Для створення веб-інтерфейсу були застосовані HTML і CSS. HTML є основною мовою розмітки, яка використовується для побудови структури та вмісту веб-сторінок. Вона дозволяє визначати різні елементи, такі як заголовки, абзаци, таблиці, форми, зображення та посилання. CSS використовується для оформлення та стилізації веб-сторінок. За його допомогою можна змінювати кольори, шрифти, розміри, відступи, фонові зображення та інші візуальні властивості HTML-елементів. CSS також дозволяє розташовувати елементи на сторінці, встановлювати їх позиціонування, створювати анімації та забезпечувати адаптивність до різних пристроїв і розмірів екранів. Використання HTML і CSS дає змогу розробникам створювати привабливі, структуровані та зручні для користувачів веб-інтерфейси. HTML відповідає за структуру та семантику сторінки, тоді як CSS забезпечує її візуальне оформлення та вигляд. Разом ці технології дозволяють створювати естетично привабливі та функціональні веб-інтерфейси для взаємодії з користувачами. Вебзастосунок використовував фреймворк Tailwind CSS для побудови свого користувацького інтерфейсу. Tailwind CSS є сучасним utility-first CSS-фреймворком, який дозволяє швидко та ефективно створювати індивідуальні дизайни без необхідності писати власний CSS-код з нуля. На відміну від традиційних CSS-фреймворків, таких як Bootstrap чи Bulma, які надають готові компоненти та стилі, Tailwind CSS пропонує набір низькорівневих утилітарних класів. Ці класи можна комбінувати безмежно, що дає змогу створювати унікальні та гнучкі інтерфейси користувача. Фреймворк підтримує адаптивний дизайн через використання мобільних запитів, вбудованих безпосередньо в утилітарні класи. Це дозволяє легко створювати інтерфейси, що добре виглядають на будь-яких пристроях, від мобільних телефонів до настільних комп'ютерів. Крім того, Tailwind CSS підтримує псевдокласи для станів елементів (наприклад, `:hover`, `:focus`), що робить можливим додавання інтерактивності

без додаткового CSS. Однією з ключових особливостей Tailwind CSS є можливість видалення невикористаних стилів при побудові проєкту для продакшн середовища. Це здійснюється за допомогою інструментів, таких як PurgeCSS, і дозволяє значно зменшити розмір фінального CSS-файлу, покращуючи швидкість завантаження сторінки. Використання Tailwind CSS у проєкті значно скоротило час розробки та полегшило процес стилізації компонентів. Завдяки утилітарним класам, розробники могли швидко прототипувати інтерфейси та вносити зміни без глибокого занурення в CSS. Це забезпечило високу якість та узгодженість дизайну, а також спростило командну роботу над кодом.

3 ИСПОЛЬЗОВАНИЕ РЕГРЕССИОННОГО АНАЛИЗА ДЛЯ АВТОМАТИЗАЦИИ ИНЖЕНЕРНЫХ РАСЧЕТОВ

Аналіз залежностей у складних системах є однією з ключових задач при математичному моделюванні. У даному дослідженні для визначення коефіцієнтів k_t , X та Y обрано метод поліноміальної регресії. Цей метод дозволяє описувати нелінійні залежності між змінними та має широку сферу застосування, зокрема у фізичних, хімічних та технічних процесах.

Мета даного дослідження полягає у побудові математичної моделі, оцінці якості її апроксимації та аналізі отриманих результатів. Особлива увага приділяється якості апроксимації даних, яка оцінюється за допомогою залишкової суми квадратів (SSE), коефіцієнта детермінації (R^2) та інших статистичних критеріїв.

3.1 Поліном: визначення та основні властивості

Поліномом називається алгебраїчний вираз, який складається з кінцевої суми добутків змінних, піднесених до цілих невід'ємних степенів, та відповідних коефіцієнтів. Загальний вигляд полінома однієї змінної x має форму:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (3.1)$$

де n — степінь полінома;

$a_n, a_{n-1}, \dots, a_0$ — коефіцієнти полінома;

Степенем полінома називається найбільший показник степеня змінної при ненульовому коефіцієнті, що визначає "ступінь складності" полінома.

Коефіцієнти полінома є числами, які множать змінні або їх степені, і можуть бути дійсними або комплексними, залежно від контексту задачі.

Поліноми можна додавати, віднімати, множити та ділити. При додаванні або відніманні поліномів додаються або віднімаються відповідні коефіцієнти при однакових степенях змінної. При множенні поліномів коефіцієнти перемножуються, а степені змінної додаються. Поліном можна поділити на інший поліном нижчого або того ж степеня з отриманням частки і остачі.

Значення змінної x , при яких $P(x)=0$, називаються коренями або нулями полінома. Згідно з основною теоремою алгебри, поліном степеня n має рівно n коренів у комплексній площині (з урахуванням кратності). Поліном можна розкласти на добуток поліномів нижчого степеня. Якщо відомі корені полінома, його можна подати у вигляді:

$$P(x) = a_n(x - x_1)^{k_1}(x - x_2)^{k_2} \dots (x - x_m)^{k_m} \quad (3.2)$$

де $x_1, x_2, \dots, x_m$ — корені полінома;

$k_1, k_2, \dots, k_m$ — їх кратності.

Поліноми широко застосовуються в різних галузях математики та фізики.

Вони використовуються при моделюванні процесів, в теорії наближень, при розв'язанні диференціальних рівнянь та в багатьох інших областях.

Наприклад, розглянемо поліном другого степеня (квадратний поліном):

$$P(x) = 2x^2 - 3x + 1 \quad (3.3)$$

У цьому випадку степінь полінома $n=2$, коефіцієнти $a_2=2$, $a_1=-3$, $a_0=1$. Можна знайти корені за формулою квадратного рівняння.

Одним із основних застосувань поліномів у моделюванні є апроксимація функцій. За допомогою поліномів можна наближено представити складні функції, для яких аналітичний вираз може бути невідомим або занадто складним для використання. Поліноміальна

апроксимація дозволяє спростити обчислення та аналіз функцій, зберігаючи при цьому необхідну точність.

3.2 Сучасний стан справ в області математичного моделювання за допомогою поліномів

Математичне моделювання є невід'ємною частиною сучасних наукових досліджень та інженерних застосувань, дозволяючи описувати, аналізувати та прогнозувати поведінку складних систем. Одним із основних інструментів математичного моделювання є поліноміальні моделі, які використовуються для апроксимації залежностей між змінними. У цьому огляді розглянемо сучасний стан справ у використанні поліноміальних моделей, аналіз їхньої точності для різних порядків, а також порівняння з нелінійними та експоненційними моделями при апроксимації змінних X та Y .

Поліноміальні моделі є одними з найпоширеніших методів апроксимації завдяки їх простоті та гнучкості. Вони дозволяють описати нелінійні залежності між змінними шляхом розширення лінійної регресії до більш високих порядків. Поліноміальні моделі використовуються в різних галузях, таких як економіка, фізика, біологія, інженерія та інші, для вирішення задач прогнозування, оптимізації та аналізу даних.

Точність поліноміальних моделей значною мірою залежить від їхнього порядку. Модель першого порядку (лінійна) часто є основою для початкового аналізу, але може бути недостатньою для опису складніших залежностей. Збільшення порядку полінома дозволяє моделі краще відповідати даним, знижуючи похибку апроксимації. Однак підвищення порядку моделі також збільшує ризик перенавчання (overfitting), коли модель занадто добре підходить під навчальні дані, але погано узагальнюється на нові.

Для змінної kt , аналіз точності поліноміальних моделей різного порядку показує, що моделі середніх порядків (наприклад, другий або третій)

часто забезпечують оптимальний баланс між точністю та складністю. Вони можуть адекватно відобразити основні тенденції даних без надмірного ускладнення. Використання методів регуляризації, таких як Ridge або Lasso регресії, може допомогти зменшити ризик перенавчання при використанні високопорядкових поліномів.

При апроксимації змінних X та Y часто розглядаються не лише поліноміальні, але й інші типи моделей, зокрема нелінійні та експоненційні. Нелінійні моделі можуть включати різні функціональні форми, такі як логарифмічні, тригонометричні або раціональні функції, що дозволяє більш гнучко адаптуватися до складних залежностей між змінними.

Експоненційні моделі особливо корисні для опису процесів, що характеризуються швидким зростанням або спадом, таких як розповсюдження хвороб, інвестиційні процеси або радіоактивний розпад. Вони дозволяють моделювати експоненційні зміни, що часто спостерігаються в природі та техніці.

Порівняння поліноміальних моделей з нелінійними та експоненційними показує, що вибір моделі залежить від характеру даних та конкретної задачі. Поліноміальні моделі можуть бути більш універсальними, але іноді менш інтуїтивно зрозумілими для специфічних типів даних, де нелінійні або експоненційні моделі можуть забезпечити кращу відповідність. Наприклад, для даних, що демонструють експоненціальне зростання, експоненційна модель може забезпечити точнішу апроксимацію з меншою кількістю параметрів у порівнянні з високопорядковим поліномом.

Сучасний стан математичного моделювання з використанням поліноміальних моделей демонструє їхню високу ефективність та широке застосування в різних галузях науки та техніки. Аналіз точності поліноміальних моделей різного порядку показує, що вибір оптимального порядку є ключовим для досягнення балансу між точністю та узагальнюваністю. Порівняння з нелінійними та експоненційними моделями підкреслює важливість вибору відповідної моделі залежно від характеру даних.

та специфіки задачі. В майбутньому, подальші дослідження можуть зосередитись на комбінуванні різних типів моделей та використанні машинного навчання для автоматичного вибору найкращої моделі для конкретних даних, що ще більше підвищить точність та ефективність математичних моделей.

3.3 Аналіз точності поліноміальних моделей різного порядку для kT

У межах даного дослідження було проведено аналіз залежності між температурою та коефіцієнтом kT , який використовується для розрахунку характеристик підшипників. Основною метою є побудова поліноміальних моделей різного порядку для апроксимації цієї залежності та вибір найоптимальнішої моделі за точністю. Для кожної моделі оцінено основні метрики, зокрема коефіцієнт детермінації R^2 , скоригований R^2 , суму квадратів залишків SSE і середньоквадратичну похибку RMSE.

Залежність kT від температури зазвичай є нелінійною, що обумовлює доцільність використання поліноміальних моделей для апроксимації цієї залежності. Додатково враховувалося, що шум даних може виникати через похибки вимірювального обладнання або нестабільність умов експерименту.

У межах даного дослідження розглянуто побудову та оцінку точності поліноміальних моделей другого, третього та четвертого порядків для апроксимації залежності між змінними x_{data} та y_{data} . Для кожної моделі проведено аналіз основних метричних показників, таких як коефіцієнт детермінації R^2 , скоригований коефіцієнт детермінації $Adj R^2$, сума квадратів залишків SSE, а також середньоквадратична похибка RMSE. Нижче наведено детальний опис отриманих результатів та зроблено висновки щодо адекватності моделей.

Поліноміальна модель другого порядку (poly2):

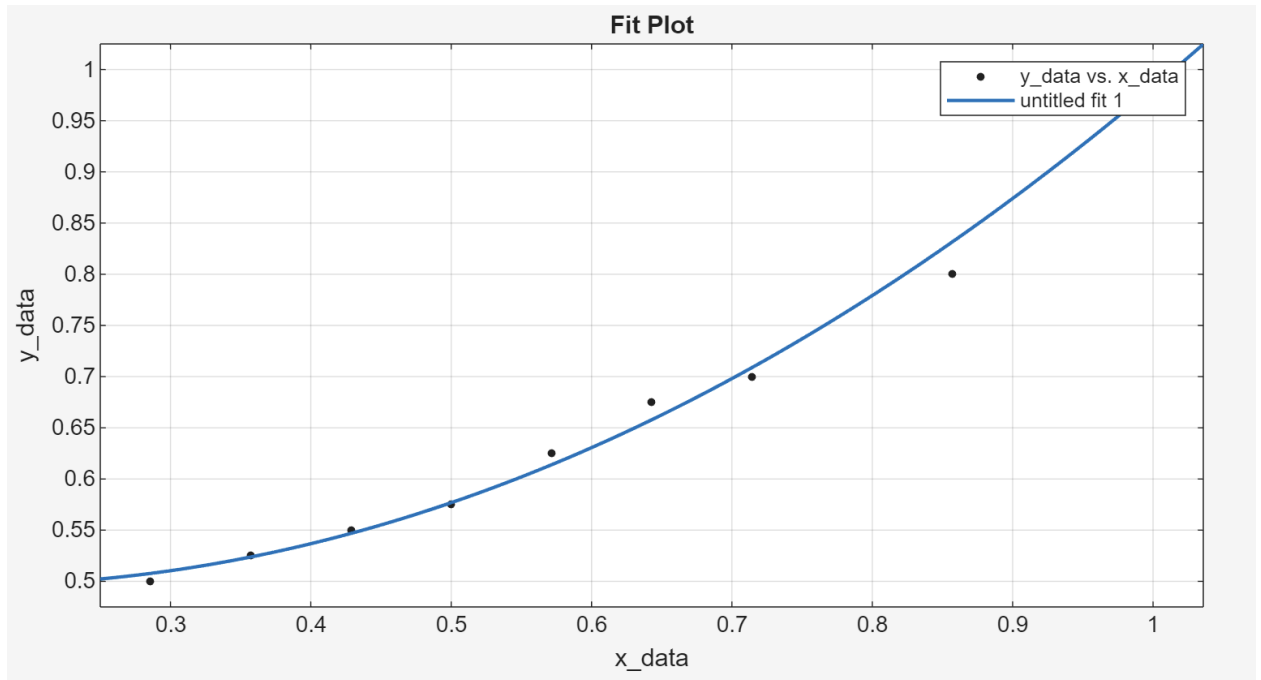


Рисунок 3.1 – модель другого порядку (poly2)

Аналізуючи першу модель, було побудовано квадратичний поліном. На графіку видно, що крива досить добре відповідає тренду емпіричних даних. Основні характеристики моделі свідчать про її високу якість: значення коефіцієнта детермінації R^2 становить 0.99066, що говорить про те, що близько 99% варіації залежної змінної пояснюється моделлю. При цьому скоригований R^2 також наближається до цього значення — 0.98754. Однак сума квадратів залишків SSE дорівнює 0.0018815, а RMSE — 0.017708, що свідчить про наявність залишкових похибок. Хоча ці похибки незначні, можливе поліпшення моделі через збільшення порядку полінома.

Поліноміальна модель третього порядку (poly3):

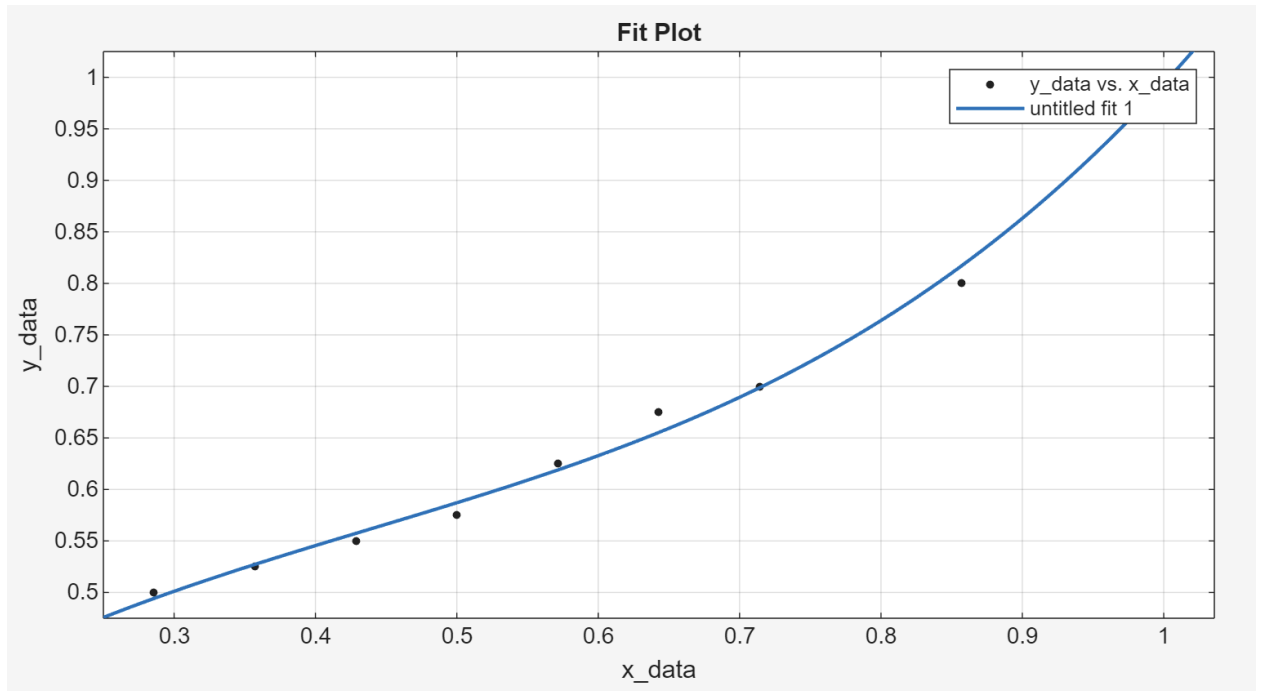


Рисунок 3.2 – модель третього порядку (poly3)

Під час використання полінома третього порядку точність моделі зросла, про що свідчить підвищення R^2 до 0.99505. Крім того, скоригований коефіцієнт детермінації досягнув значення 0.99208, що підтверджує доцільність використання моделі в цьому випадку. Сума квадратів залишків SSE зменшилася до 0.0009965, а RMSE — до 0.014117, що є свідченням зменшення похибок. На графіку видно, що модель краще узгоджується з емпіричними даними, порівняно з попередньою. Отже, використання полінома третього порядку забезпечує більш точне наближення.

Поліноміальна модель четвертого порядку (poly4):

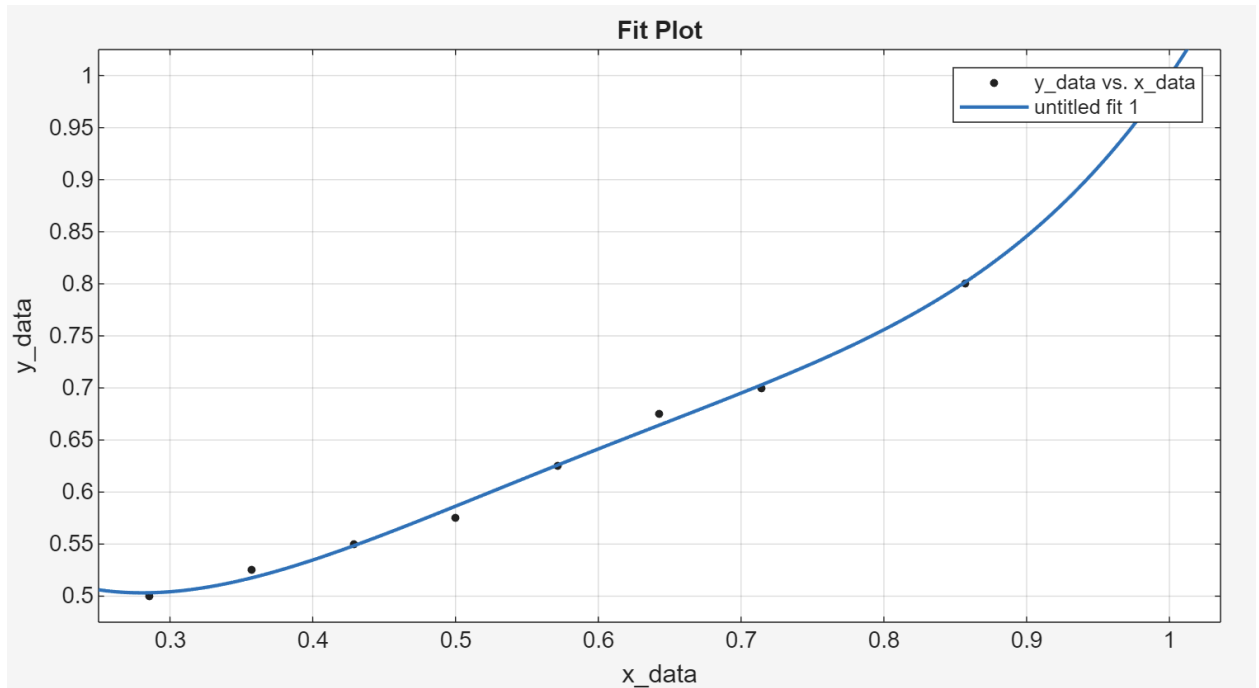


Рисунок 3.3 – модель четвертого порядку (poly4)

Найвищу точність показала модель четвертого порядку, яка продемонструвала R^2 на рівні 0.99839. Скориговане значення R^2 — 0.99678, що підтверджує, що модель не тільки добре описує залежність, але й не схильна до переобчислення. Значення SSE склало 0.00032468, а RMSE зменшилося до 0.0090094, що вказує на мінімальні похибки. Графік підтверджує, що ця модель найточніше відображає характер змін між x_{data} та y_{data} .

Результати дослідження свідчать про те, що з підвищенням порядку полінома покращується якість апроксимації. Зокрема, поліном четвертого порядку демонструє найкращу відповідність емпіричним даним, забезпечуючи мінімальні залишкові похибки. Проте слід враховувати, що моделі з вищим порядком можуть бути більш чутливими до шуму в даних, а також потребують більшого обсягу обчислювальних ресурсів. Тому вибір оптимального порядку полінома повинен враховувати не лише точність, але й складність моделі та обсяг доступних даних.

Таким чином, для розв'язання задач, які вимагають високої точності, доцільно використовувати моделі третього або четвертого порядку, а для простіших задач із менш критичними вимогами до точності цілком достатньо моделі другого порядку.

3.4 Метод найменших квадратів

Метод найменших квадратів (МНК) є одним із найпоширеніших і ефективних способів оцінювання параметрів у математичному моделюванні, статистиці та економетриці. Його основна мета полягає у знаходженні таких параметрів моделі, які забезпечують найкраще узгодження між експериментальними даними та теоретичною моделлю. Цей метод мінімізує суму квадратів відхилень між спостережуваними значеннями залежної змінної та значеннями, які прогнозуються моделлю.

Суть методу полягає в наступному: якщо маємо набір експериментальних даних, що складається із значень незалежної змінної x_i та відповідних значень залежної змінної y_i , потрібно знайти таку функцію $f(x, \beta)$, де β — параметри моделі, яка б найкраще описувала ці дані. Критерієм "найкращості" є мінімізація суми квадратів залишків:

$$S(\beta) = \sum_{i=1}^n (y_i - f(x_i, \beta))^2 \quad (3.4)$$

де $y_i - f(x_i, \beta)$ — залишок (або похибка моделі) для кожного спостереження. Основні кроки реалізації МНК включають визначення виду функції $f(x, \beta)$, вибір початкових значень параметрів β , а також застосування математичних методів для мінімізації цільової функції $S(\beta)$. Для лінійних моделей $f(x, \beta)$ є лінійною комбінацією параметрів, наприклад, $f(x, \beta) = \beta_0 + \beta_1 x$, що значно

спрощує обчислення, оскільки для таких моделей мінімум $S(\beta)$ можна знайти аналітичним шляхом.

У випадку нелінійних моделей мінімізація може вимагати чисельних методів, таких як градієнтний спуск, метод Ньютона або інші алгоритми оптимізації. Однак основний принцип залишається незмінним — зменшення суми квадратів залишків.

Метод найменших квадратів має широку область застосування, зокрема в регресійному аналізі, інтерполяції, розв'язанні диференціальних рівнянь і багатьох інших завданнях. Його переваги включають простоту реалізації, математичну строгість і надійність у роботі з великими обсягами даних. Водночас, МНК має обмеження, зокрема чутливість до викидів у даних, що можуть суттєво вплинути на точність моделі. Для подолання цього недоліку використовуються модифіковані варіанти МНК, такі як зважений метод найменших квадратів або робастні методи.

3.5 Аналіз апроксимації X та Y

Для визначення точності апроксимації часто використовують залишки, які є різницею між виміряними значеннями y та розрахованими моделлю значеннями $\hat{y}$. Чим меншими є залишки, тим точніше модель відображає вихідні дані.

Залишки аналізуються за допомогою таких підходів:

Сума квадратів залишків SSE:

$$SSE = \sum (y_i - \hat{y}_i)^2 \quad (3.5)$$

Цей показник характеризує сумарну похибку апроксимації та мінімізується в методі найменших квадратів.

Середня квадратична похибка MSE: Розраховується як середнє значення SSE, поділене на кількість ступенів свободи:

$$MSE = \frac{SSE}{n - p} \quad (3.6)$$

де n — кількість спостережень;
 p — кількість параметрів моделі.

Для оцінки якості моделі також використовують дисперсійний аналіз, який дозволяє оцінити відношення поясненої та загальної дисперсії:

1. Загальна сума квадратів TSS:

$$TSS = \sum (y_i - \bar{y})^2 \quad (3.7)$$

2. Пояснена сума квадратів SSR:

$$SSR = \sum (\hat{y}_i - \bar{y})^2 \quad (3.8)$$

3. Коефіцієнт детермінації R^2 :

$$R^2 = 1 - \frac{SSE}{TSS} \quad (3.9)$$

Він показує частку дисперсії, пояснену моделлю.

Для оцінки надійності параметрів моделі використовуються довірчі інтервали, які обчислюються на основі стандартних помилок. Стандартні помилки параметрів визначаються через асимптотичну матрицю варіаційно-коваріаційних значень. Довірчі інтервали побудовані за допомогою розподілу Стюдента, що дозволяє визначити діапазон значень параметра з заданим рівнем довіри (наприклад, 95%).

Використовувалася функція моделі виду:

$$fit(x, a, b, c, d) = \frac{a * x + b}{x^2 + c * x + d} \quad (3.10)$$

З використанням експериментальних даних, параметри a,b,c,d були знайдені за допомогою методу найменших квадратів.

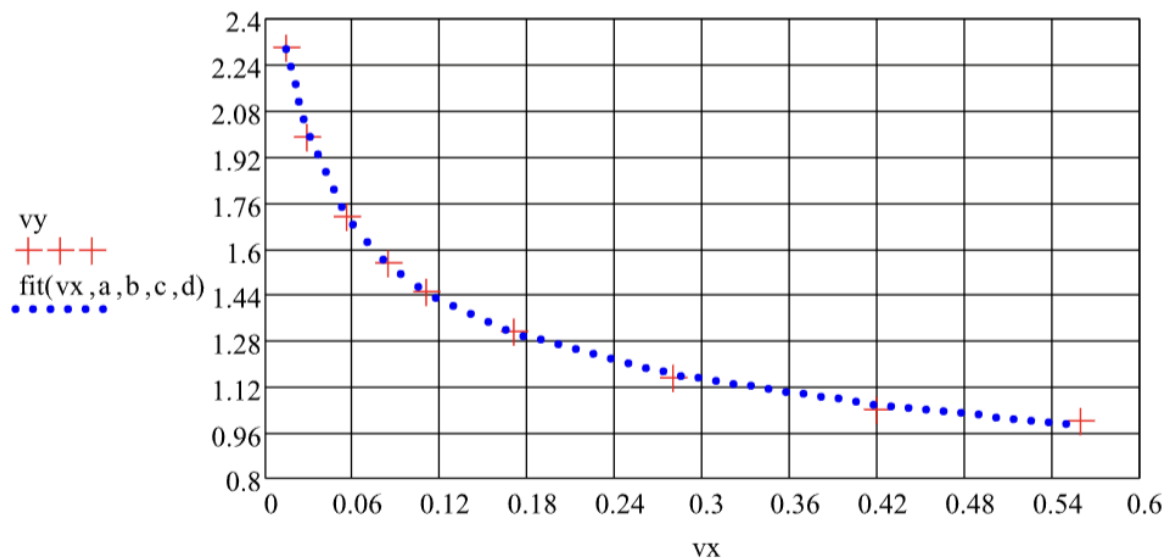


Рисунок 3.4 – Порівняння експериментальних даних і апроксимації функції

Параметри моделі:

Згідно з отриманими результатами, значення параметрів: $a=2.576$, $b=0.2059$, $c=2.305$, $d=0.07309$.

Критерії оцінки:

1. Сума квадратів залишків (SSE): $x^2 = 1.35 * 10^{-3}$.
2. Коефіцієнт детермінації (Fit): 0.999.
3. Середньоквадратична похибка (MSE): $2.25 * 10^{-4}$.

Використовується експоненційна функція:

$$fit(x, a, b, c) = a * e^{b*x} + c \quad (3.11)$$

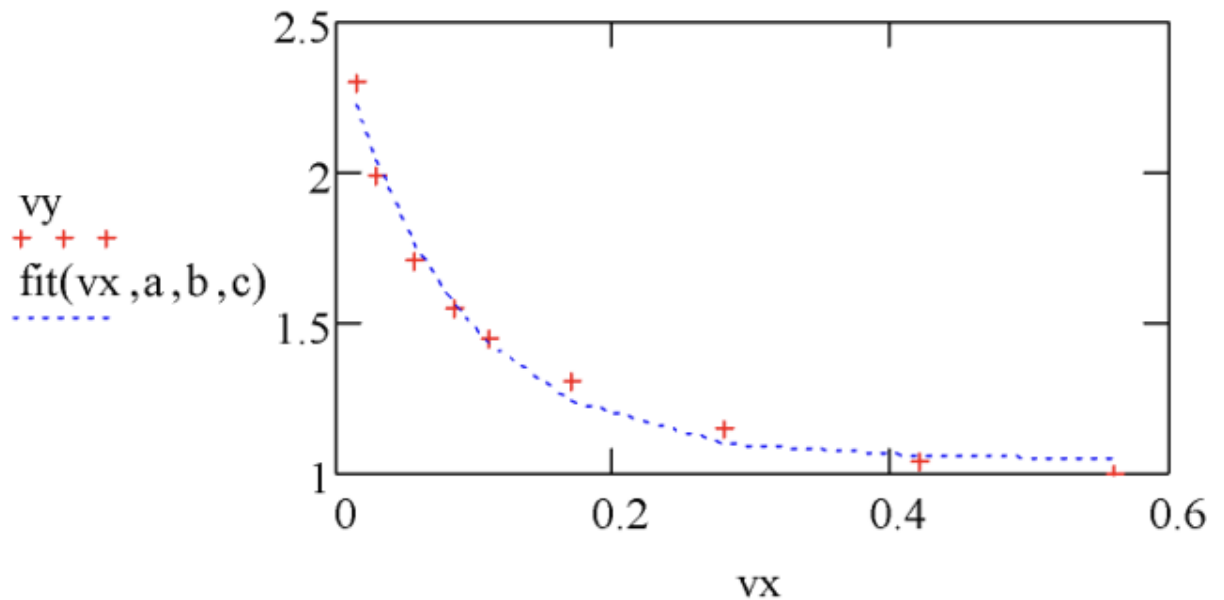


Рисунок 3.5 – Порівняння експериментальних даних і апроксимації функції (модель з параметрами a, b, c)

Згідно з даними, отриманими в результаті підгонки: $a=1.384$,
 $b=-11.737$, $c=1.051$

Значення vx і vy були отримані експериментально й описують залежність, яку слід апроксимувати за допомогою обраної функції.

Критерії оцінки:

1. Сума квадратів залишків (SSE): $x^2 = 0.023$.
2. Коефіцієнт детермінації (Fit): $3.9 * 10^3$.
3. Середньоквадратична похибка (MSE): 0.958.

Показники точності:

Критерій	Нелінійна модель	Експоненційна модель
Сума квадратів залишків (SSE)	$1.35 \cdot 10^{-3}$	0.023
Середньоквадратична помилка (MSE)	$2.25 \cdot 10^{-4}$	$3.9 \cdot 10^{-3}$
Коефіцієнт детермінації (Fit)	0.999	0.985

SSE та MSE: Нелінійна модель демонструє менші значення залишків, що вказує на більш точну апроксимацію.

Коефіцієнт детермінації: Для нелінійної моделі $R^2 = 0.999$, що означає, що 99.9% варіацій даних пояснюються моделлю. Для експоненційної моделі $R^2 = 0.985$, що також вказує на високу точність, але трохи меншу.

Аналіз залишків:

1. Нелінійна модель: Залишки рівномірно розподілені навколо нуля, що свідчить про високу адекватність моделі.
2. Експоненційна модель: Залишки також рівномірно розподілені, однак їх діапазон більший, що свідчить про трохи гіршу відповідність моделі реальним даним.

Складність моделі:

1. Нелінійна модель використовує 4 параметри (a,b,c,d), що підвищує її складність і може призвести до перенавчання на інших даних.
2. Експоненційна модель використовує 3 параметри (a,b,c), що робить її простішою і більш узагальненою для практичного використання.

Для аналізу даних було обрано нелінійну модель, оскільки вона продемонструвала значно вищу точність порівняно з експоненційною. Нелінійна модель краще описує залежність між змінними, має менші

залишкові похибки та вище значення точності. Ця модель забезпечує найбільш точний опис експериментальних даних, що є ключовим для досягнення мети дослідження.

3.6 Довірчі інтервали

Для моделі $fit(x, a, b, c) = a * e^{b * x} + c$ розрахунки довірчих інтервалів виконуються за допомогою асимптотичних оцінок параметрів. Ці оцінки є лише наближенням до справжніх параметрів, тому довірчі інтервали формуються на основі матриці дисперсії-коваріації параметрів. Ця матриця визначається шляхом множення матриці похідних, відомої як матриця Якобіана, на середньоквадратичну похибку.

Довірчі інтервали були визначені на основі матриці дисперсії-коваріації, яка враховує вплив варіації кожного параметра на поведінку функції. Для цього було використано асимптотичний підхід із залученням матриці Якобіана, яка визначає часткові похідні моделі за кожним параметром. Всі розрахунки проведені для рівня значущості $\alpha=0.01$, що відповідає 99% рівню довіри.

Отримані результати довірчих інтервалів для параметрів показують наступні межі:

1. Параметр a : від 1.103 до 1.665,
2. Параметр b : від -17.212 до -6.263
3. Параметр c : від 0.959 до 1.203

Графічний аналіз рисунок [3.6] показує, що модель добре відповідає вимірним даним. Довірчі інтервали для прогнозованих значень демонструють, що розрахункові точки не виходять за межі довіри, що свідчить про стабільність моделі та коректність її параметрів.

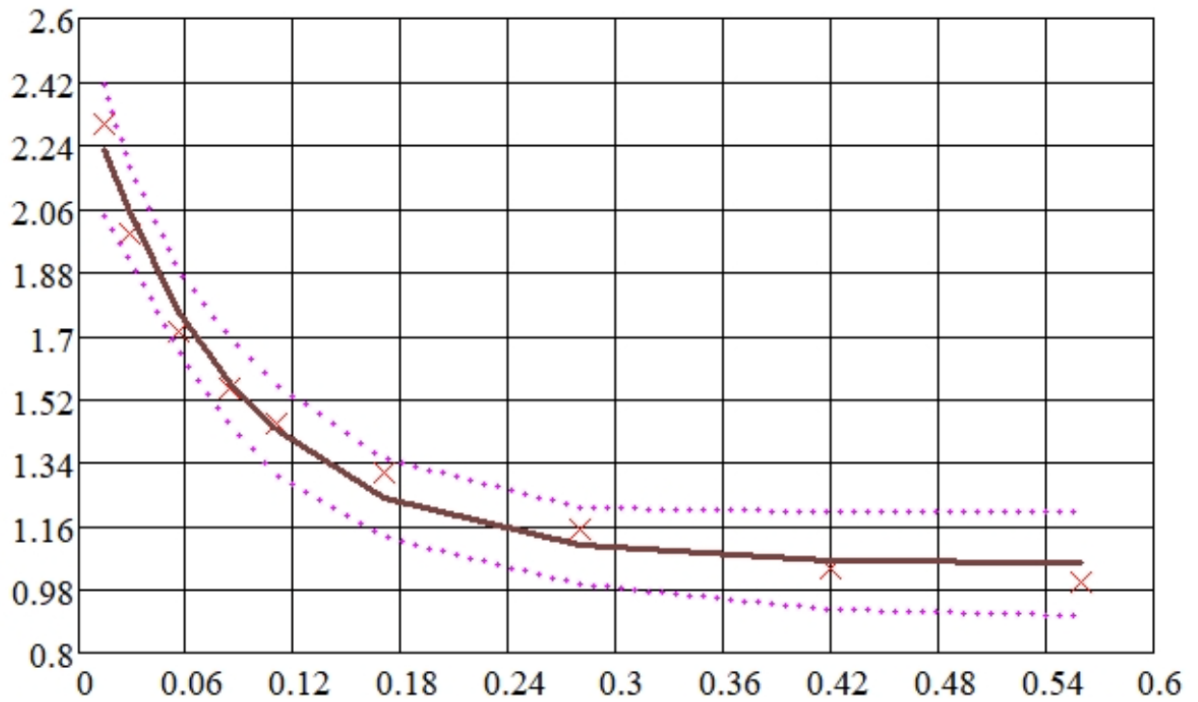


Рисунок 3.6 – Порівняння експериментальних даних і апроксимації функції
(модель з параметрами a , b , c)

Ці значення свідчать про те, що кожен із параметрів моделі є статистично значущим, а інтервали довіри досить вузькі, що вказує на низький рівень невизначеності в оцінках.

Прогнозовані значення функції також були забезпечені довірчими інтервалами. Ці інтервали враховують як дисперсію похибки моделі, так і варіацію, що зумовлена невизначеністю параметрів.

На основі проведеного аналізу можна стверджувати, що дана модель є ефективною для опису залежності, а її параметри мають високий рівень надійності. Це дозволяє використовувати її для прогнозування значень у межах досліджуваного діапазону зі значною впевненістю.

ВИСНОВКИ

У процесі виконання магістерської роботи було розроблено та реалізовано веб-сервіс для розрахунку радіальних кулькових підшипників, який поєднує в собі сучасні технології та математичні моделі. Створена математична модель на основі поліноміальних методів дозволяє з високою точністю прогнозувати поведінку підшипників під різними навантаженнями та умовами експлуатації. Це значно підвищує надійність та ефективність інженерних розрахунків у галузі машинобудування.

Використання фреймворку NestJS для серверної частини забезпечило структурованість та масштабованість бекенду, дозволяючи легко інтегрувати додаткові функціональні можливості у майбутньому. База даних MongoDB надала гнучкість у збереженні та обробці великих обсягів даних, що є критично важливим для інженерних застосувань з великою кількістю параметрів та результатів розрахунків. Клієнтська частина, реалізована на Vue.js, забезпечила інтуїтивно зрозумілий інтерфейс користувача, сприяючи підвищенню зручності та швидкості роботи з сервісом.

Розроблений веб-сервіс не лише спрощує процес розрахунку радіальних кулькових підшипників, але й відкриває можливості для подальшого розвитку та вдосконалення. Він може бути інтегрований у виробничі процеси підприємств, сприяючи автоматизації та цифровізації галузі. Крім того, сервіс може служити освітнім інструментом для студентів та фахівців, надаючи доступ до сучасних методів розрахунку та аналізу.

Загалом, виконана робота демонструє успішне поєднання теоретичних знань та практичних навичок у сфері програмування, математичного моделювання та інженерії. Розроблений веб-сервіс має потенціал стати важливим інструментом для фахівців у галузі машинобудування, сприяючи підвищенню ефективності та якості інженерних рішень.

Таким чином, виконана магістерська робота має як теоретичну, так і практичну цінність. Застосування сучасних математичних моделей, побудованих на основі поліноміальних методів, забезпечує високу точність розрахунків, що є важливим для забезпечення тривалого терміну експлуатації радіальних кулькових підшипників. Інтеграція передових технологій, таких як NestJS, Vue.js та MongoDB, демонструє успішний підхід до створення масштабованих і функціональних веб-застосунків у сфері інженерії.

Крім того, даний веб-сервіс може стати основою для створення комплексних інженерних платформ, що об'єднують декілька напрямків аналізу та розрахунків. Завдяки своїй адаптивності й простоті у використанні, він може сприяти популяризації сучасних технологій у машинобудівній галузі, а також покращити освітній процес у підготовці фахівців.

Таким чином, результати роботи свідчать про доцільність впровадження розроблених рішень у практику машинобудівних підприємств та навчальних закладів, сприяючи подальшій автоматизації, підвищенню якості та інноваційності інженерних процесів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ГОСТ 520-2011. Кулькові та роликові підшипники. Основні технічні вимоги.
2. ISO 281:2007. Підшипники кочення. Динамічне навантаження та розрахунок тривалості експлуатації.
3. Тимошенко С.П., Гудзь І.П. "Розрахунок та проектування підшипників ковзання та кочення". Київ: Техніка, 2018.
4. Vue.js Documentation. URL: <https://vuejs.org/> (дата звернення: 05.10.2024).
5. NestJS Documentation. URL: <https://nestjs.com/> (дата звернення: 12.08.2024).
6. MongoDB Documentation. URL: <https://www.mongodb.com/docs/> (дата звернення: 08.10.2024).
7. TypeScript Handbook. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 22.11.2024).
8. ml-regression Library Documentation. URL: <https://www.npmjs.com/package/ml-regression> (дата звернення: 11.11.2024).
9. Креймер Н.А. "Метод найменших квадратів та його застосування у моделюванні". Москва: Наука, 2016.
10. Поляков В.С. "Математичні методи та моделі для технічного аналізу". Харків: Основа, 2020.
11. WebStorm IDE Documentation. URL: <https://www.jetbrains.com/webstorm/documentation/> (дата звернення: 01.08.2024).
12. Smith A., Johnson B. "Database Design Principles". New York: Wiley, 2021.
13. Анур'єв В.І. Довідник конструктора-машинобудівника: У 3 т. Т. 1. — 9-те вид., перероб. і доп. — М.: Машинобудування, 2006. — 928 с.
14. Решетов Д.М. Деталі машин: Підручник для вузів. — 4-те вид., перероб. і доп. — М.: Машинобудування, 1989. — 640 с.