

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

Бакалаврська кваліфікаційна робота на тему:
«ДОСЛІДЖЕННЯ РОЗШИРЕНОГО ФІЛЬТРА КАЛМАНА ДЛЯ
СИСТЕМИ НАВІГАЦІЇ НАЗЕМНОГО РОБОТА»

Виконав: студент IV курсу, групи
АКІТР-23мс
спеціальності 174 – Автоматизація,
комп'ютерно-інтегровані технології
та робототехніка
(шифр і назва спеціальності)

Коломієць Д. В.
(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри АІТ
Маслій Р. В.
(прізвище та ініціали)

« 16 » 06 2025 р.

Рецензент: _____

Д.Т.Н., професор каф. КСУ

Юхимчук М. С.

(прізвище та ініціали)

« 18 » 06 2025 р.

Допущено до захисту
завідувач кафедри АІТ

« 18 » 06 2025 р.

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра автоматизації та інтелектуальних інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Галузь знань 17 – Електроніка, автоматизація та електронні комунікації

Спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітньо-професійна програма Інтелектуальні комп'ютерні системи управління



ЗАТВЕРДЖУЮ

завідувач кафедри АІТ

д.т.н., проф. Олег БІСІКАЛО

«18» 06 2025 року

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Коломійцю Денису Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження розширеного фільтра Калмана для системи навігації наземного робота»

керівник роботи Маслій Роман Васильович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «20» березня 2025 року №97

2. Термін подання студентом роботи 10.06.2025 р.

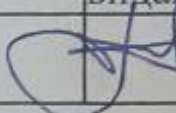
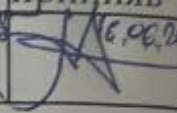
3. Вихідні дані до роботи:

- номінальна частота оновлення сенсорних даних: IMU: 100–200 Гц; GPS: 1–10 Гц; Одометрія: 50–100 Гц;
- точність оцінки позиції: GPS: 2–5 м; Після застосування EKF: 0.1–0.5 м;
- шум процесу Q: діагональна матриця з відхиленнями 0.1 м (x, y), 0.01 рад (θ);
- шум вимірювань R: діагональна матриця з відхиленнями 2.5 м (GPS), 0.05 рад/с (IMU);
- операційна система: Linux.

4. Зміст текстової частини (перелік питань, які потрібно розробити) провести огляд існуючих методів оцінки стану, здійснити моделювання системи навігації наземного робота, здійснити реалізацію розширеного фільтра Калмана при використанні одометрії, GPS та IMU сенсорів, здійснити програмну реалізацію та тестування системи навігації наземного робота на основі розширеного фільтра Калмана.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) Схема програми розширеного фільтру Калмана для системи навігації наземного робота; Траєкторія наземного робота при русі по колу; Траєкторія та орієнтація наземного робота при прямому русі; Траєкторія та орієнтація наземного робота при зигзагоподібному русі

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Маслій Р. В., доц. каф. АІТ	 20.03.2025	 16.06.2025

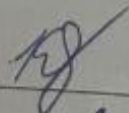
7. Дата видачі завдання 20.03.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Термін виконання		Примітка
		початок	закінчення	
1	Вибір, узгодження та затвердження теми БКР	03.10.2024	09.10.2024	виконано
2	Аналіз предметної області та опрацювання літературних джерел.	12.03.2025	29.03.2025	виконано
3	Постановка задач для вирішення в БКР	01.04.2025	26.04.2025	виконано
4	Вирішення поставлених задач	29.04.2025	10.05.2025	виконано
5	Підтвердження достовірності отриманих результатів	13.05.2025	24.05.2025	виконано
7	Оформлення пояснювальної записки та графічної частини	27.05.2025	31.05.2025	виконано
8	Перевірка ПЗ на відповідність вимогам	03.06.2025	07.06.2025	виконано
9	Попередній захист, доопрацювання, та рецензування БКР	10.06.2025	15.06.2025	виконано
10	Захист БКР	16.06.2025	23.06.2025	

Студент
(підпис) (ім'я та прізвище)

Керівник роботи
(підпис) (ім'я та прізвище)

 Денис КОЛОМІЄЦЬ

 Роман МАСЛІЙ

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 79 сторінок формату A4, на яких є 13 рисунків, список використаних джерел містить 22 найменування.

Виконана робота присвячена розробці та аналізу системи навігації наземного робота на основі розширеного фільтра Калмана (EKF) для режимів GPS і GPS+IMU, реалізованої на мові програмування Python.

У аналітичній частині розглянуто актуальність проблеми навігації автономних роботів, проаналізовано кінематичні моделі, характеристики сенсорів (одометрія, GPS, IMU) та методи сенсорного злиття.

У теоретичній частині досліджено принципи роботи EKF, нелінійні моделі руху та корекції, а також бібліотеки NumPy, Pandas і Matplotlib для їх реалізації.

У реалізаційній частині описано методи генерації даних для трьох траєкторій (рух по колу, прямолінійний, зигзагоподібний), розробку EKF, обчислення Dead Reckoning і оцінку точності за метриками RMSE та MAE. Проведено аналіз впливу шуму GPS у матриці R ($\sigma_{GPS} = 1, 2, 5, 5\text{м}$) на результати.

Ключові слова: розширений фільтр Калмана, навігація робота, сенсорне злиття, GPS, IMU, одометрія.

ABSTRACT

The bachelor's qualification work consists of 79 pages of A4 format, on which there are 13 figures, the list of used sources contains 22 names.

The work is devoted to the development and analysis of a ground robot navigation system based on the extended Kalman filter (EKF) for GPS and GPS+IMU modes, implemented in the Python programming language.

The analytical part considers the relevance of the problem of navigation of autonomous robots, analyzes kinematic models, sensor characteristics (odometry, GPS, IMU) and sensor fusion methods.

The theoretical part examines the principles of EKF operation, nonlinear motion and correction models, as well as the NumPy, Pandas and Matplotlib libraries for their implementation.

The implementation part describes data generation methods for three trajectories (circular motion, rectilinear, zigzag), EKF development, Dead Reckoning calculations and accuracy assessment using RMSE and MAE metrics. The impact of GPS noise in the R matrix (GPS=1,2.5,5m) on the results was analyzed.

Keywords: Extended Kalman filter, robot navigation, sensor fusion, GPS, IMU, odometry.

ЗМІСТ

ВСТУП.....	4
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Фільтр Калмана.....	6
1.2 Розширений фільтр Калмана.....	8
1.3 Порівняння ЕКФ з іншими методами оцінки стану.....	12
1.4 Висновки до розділу.....	16
2 МОДЕЛЮВАННЯ СИСТЕМИ НАВІГАЦІЇ НАЗЕМНОГО РОБОТА.....	17
2.1 Опис наземного робота: структура, сенсори.....	17
2.2 Математична модель руху робота.....	20
2.3 Моделювання сенсорних даних.....	23
2.4 Висновки до розділу.....	28
3 РЕАЛІЗАЦІЯ РОЗШИРЕНОГО ФІЛЬТРУ КАЛМАНА.....	29
3.1 Розширений фільтр Калмана.....	29
3.2 Інтеграція даних сенсорів у модель ЕКФ.....	33
3.3 Налаштування параметрів фільтра: коваріаційні матриці, початкові умови.....	35
3.4 Висновки до розділу.....	38
4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ.....	39
4.1 Програмне забезпечення ЕКФ.....	39
4.2 Тестування роботи ЕКФ.....	43
4.3 Висновки до розділу.....	50
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54

ДОДАТКИ	3
Додаток А (обов'язковий) Технічне завдання.....	58
Додаток Б (обов'язковий) Ілюстративна частина	63
Додаток В (обов'язковий) Лістинг програмного забезпечення.....	68
Додаток Г (обов'язковий) Протокол перевірки кваліфікаційної роботи	79

ВСТУП

Сучасний розвиток робототехніки та автономних систем вимагає високої точності навігації для забезпечення надійної роботи наземних роботів у різних умовах, таких як логістика, сільське господарство, дослідження та промислові процеси. Неточність у визначенні положення та орієнтації робота може призводити до зниження ефективності, помилок у виконанні завдань або навіть аварійних ситуацій. Традиційні методи навігації, що базуються на одному сенсорі, часто є недостатньо точними через шуми, дрейф або обмеження апаратного забезпечення.

Застосування розширеного фільтра Калмана (EKF) дозволяє інтегрувати дані від кількох сенсорів, таких як IMU, GPS і одометрія, для підвищення точності оцінки стану робота. EKF є ефективним інструментом для обробки нелінійних систем, що робить його актуальним для навігації наземних роботів у реальних умовах. Актуальність теми зумовлена зростаючою потребою в автономних системах, які можуть працювати в складних і динамічних середовищах, а також швидким розвитком технологій обробки сенсорних даних.

Дослідження в цій сфері є важливим для створення надійних і ефективних систем навігації, які можуть бути адаптовані до потреб різних галузей, сприяючи підвищенню автономності та продуктивності роботизованих платформ. Виходячи з цього, тему можна вважати актуальною.

Об'єктом дослідження є процеси навігації наземного робота в двовимірному просторі.

Предметом дослідження є розширений фільтр Калмана як метод оцінки стану в системі навігації наземного робота.

Метою дослідження є підвищення точності визначення положення та орієнтації наземного робота завдяки застосуванню розширеного фільтра

Калмана у системі навігації з інтеграцією даних сенсорів IMU, GPS і одометрії.

Для досягнення поставленої мети визначено наступні **завдання**:

- Провести порівняльний аналіз розширеного фільтру Калмана з іншими методами оцінки стану, такими як безскладовий фільтр Калмана (UKF) і фільтр частинок (PF).
- Проаналізувати сучасні підходи до використання ЕКФ у навігації роботів на основі літературних джерел.
- Описати структуру робота та характеристики сенсорів (IMU, GPS, одометрія), визначивши їхні джерела шумів і похибок.
- Розробити програмне забезпечення для моделювання сенсорних даних і реалізації ЕКФ.
- Провести експериментальне тестування розробленої системи навігації на основі ЕКФ.

Практична цінність роботи. Результати дослідження мають значну практичну цінність, оскільки розроблена система навігації на основі ЕКФ може бути застосована в автономних роботах для логістики, сільського господарства, пошуково-рятувальних операцій та інших галузей. Використання ЕКФ дозволяє інтегрувати дані від IMU, GPS і одометрії, забезпечуючи високу точність і надійність навігації навіть у умовах шумів і похибок сенсорів. Реалізована система, протестована в симуляційних або реальних умовах, сприяє зниженню навігаційних помилок, підвищенню автономності роботів і оптимізації їхньої роботи. Отримані дані про продуктивність ЕКФ, включаючи порівняння з іншими методами, надають цінну інформацію для вибору оптимальних алгоритмів у конкретних сценаріях. Розроблені алгоритми та програмне забезпечення можуть бути інтегровані в існуючі роботизовані платформи або використані як основа для подальших досліджень у сфері автономної навігації.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Фільтр Калмана

Фільтр Калмана (KF, Kalman Filter) є рекурсивним алгоритмом для оптимальної оцінки стану динамічної системи на основі зашумлених вимірювань [1,2]. Він широко застосовується в системах навігації, обробки сигналів і керування завдяки своїй здатності ефективно поєднувати передбачення моделі системи з даними сенсорів. Основна ідея фільтра Калмана полягає в ітеративному уточненні оцінки стану системи шляхом мінімізації середньоквадратичної похибки [3-4].

Фільтр Калмана базується на припущенні, що система є лінійною, а шуми процесу та вимірювань мають гаусівський розподіл. Алгоритм працює у два етапи [1, 5]:

- передбачення (Prediction): На основі математичної моделі системи та попередньої оцінки стану передбачається новий стан і його невизначеність (коваріація).
- оновлення (Update): Дані від сенсорів використовуються для корекції передбаченого стану, враховуючи їхню точність і передбачену невизначеність.

Ці етапи повторюються на кожному кроці часу, що дозволяє фільтру адаптуватися до змін у системі та сенсорних даних.

Математична модель

Фільтр Калмана описується двома основними рівняннями: моделлю системи та моделлю вимірювань.

Моделі системи визначається як:

$$x_k = Ax_{k-1} + Bu_k + w_k, \quad (1.1)$$

де:

x_k — вектор стану системи на кроці k ;

A — матриця переходу стану (описує динаміку системи);

B — матриця керування (враховує зовнішні впливи u_k);

w_k — шум процесу, що має гаусівський розподіл із нульовим середнім і коваріаційною матрицею (Q).

Модель вимірювань визначаються як:

$$z_k = Hx_k + v_k, \quad (1.2)$$

де:

z_k — вектор вимірювань;

H — матриця спостереження (пов'язує стан із вимірюваннями);

v_k — шум вимірювань, що має гаусівський розподіл із нульовим середнім і коваріаційною матрицею R .

Алгоритм фільтра Калмана

Алгоритм складається з наступних кроків:

Передбачення стану визначається як:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k. \quad (1.3)$$

Передбачення матриці коваріації визначається як:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q. \quad (1.4)$$

Оновлення стану визначається як:

Обчислення коефіцієнта Калмана:

$$K_k = P_{k|k-1} H^T (H P_{k-1|k-1} H^T + R)^{-1}. \quad (1.5)$$

Корекція оцінки стану:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H \hat{x}_{k|k-1}). \quad (1.6)$$

Оновлення коваріації:

$$P_{k|k} = (I - K_k H) P_{k|k-1}, \quad (1.7)$$

де K_k — коефіцієнт Калмана, що визначає вагу вимірювань у корекції.

Переваги та обмеження

Фільтр Калмана забезпечує оптимальну оцінку стану для лінійних систем із гаусівськими шумами, має низьку обчислювальну складність і є рекурсивним, що робить можливим його використання для роботи в реальному часі [2-6]. Однак він обмежений припущенням лінійності системи, що унеможливує його пряме застосування до нелінійних систем без модифікацій, таких як розширений фільтр Калмана.

Таким чином, фільтр Калмана є потужним інструментом для оцінки стану, який становить основу для більш складних алгоритмів, таких як EKF, що розглядаються в наступних розділах.

1.2 Розширений фільтр Калмана

Розширений фільтр Калмана (Extended Kalman Filter, EKF) є модифікацією класичного фільтра Калмана, розробленою для роботи з нелінійними динамічними системами [3, 8]. Оскільки багато реальних

систем, зокрема системи навігації наземних роботів, мають нелінійну динаміку та нелінійні моделі вимірювань, ЕКФ став одним із найпоширеніших інструментів для оцінки стану в таких задачах [5, 9].

На відміну від класичного фільтра Калмана, який припускає лінійність системи та вимірювань, ЕКФ використовує локальну лінеаризацію нелінійних функцій за допомогою розкладання в ряд Тейлора першого порядку. Це дозволяє адаптувати алгоритм до нелінійних моделей шляхом апроксимації нелінійних залежностей у точці поточної оцінки стану[9].

Алгоритм ЕКФ, як і класичний фільтр, складається з двох етапів:

- передбачення: Використовується нелінійна модель системи для прогнозування наступного стану та коваріаційної матриці.
- оновлення: Дані сенсорів коригують передбачений стан, враховуючи нелінійну модель вимірювань і коефіцієнт Калмана.

Математична модель

ЕКФ працює з нелінійними моделями системи та вимірювань, які описуються так:

Модель системи визначається як:

$$x_k = f(x_{k-1}, u_k, w_k), \quad (1.8)$$

де $f(.)$ — нелінійна функція переходу стану,

x_k — вектор стану,

u_k — вектор керування,

w_k — шум процесу з коваріаційною матрицею Q .

Модель вимірювань визначається як:

$$z_k = h(x_k, v_k), \quad (1.9)$$

де $h(.)$ — нелінійна функція спостереження,

z_k — вектор вимірювань,

v_k — шум вимірювань із коваріаційною матрицею (R).

Для лінеаризації цих нелінійних функцій обчислюються яacobіани (матриці часткових похідних):

Яacobіан моделі системи:

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{x_{k-1}, u_k}, \quad (1.10)$$

Яacobіан моделі вимірювань:

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{x_k}, \quad (1.11)$$

Алгоритм EKF має наступні кроки.

Етап передбачення

Передбачення стану:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k, 0). \quad (1.12)$$

Прогноз коваріації:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q. \quad (1.13)$$

Етап оновлення

Обчислення коефіцієнта Калмана:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}. \quad (1.14)$$

Корекція оцінки стану:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1}, 0)). \quad (1.15)$$

Оновлення коваріації:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}, \quad (1.16)$$

Особливості застосування до нелінійних систем

ЕКФ широко застосовується в системах навігації наземних роботів, де нелінійність виникає через складну кінематику руху (наприклад, повороти, нерівності поверхні) та нелінійні моделі сенсорів (GPS, IMU) [5, 8, 9]. Основні особливості:

- гнучкість: ЕКФ дозволяє інтегрувати дані від різних сенсорів, враховуючи їхні нелінійні характеристики.
- обчислювальна ефективність: Завдяки лінеаризації ЕКФ залишається відносно швидким для роботи в реальному часі.
- чутливість до початкових умов: Точність ЕКФ залежить від правильного вибору початкового стану та коваріаційних матриць (Q) і (R).
- обмеження лінеаризації: Якщо система має сильну нелінійність, апроксимація першого порядку може призводити до похибок, що вимагає ретельного налаштування.

Застосування в навігації роботів

У контексті навігації наземних роботів ЕКФ використовується для оцінки позиції, орієнтації та швидкості на основі даних від GPS, інерційних вимірювальних одиниць (IMU) і одометрії. Наприклад, нелінійна модель руху робота враховує його кінематику, а нелінійна модель вимірювань GPS враховує геометричні залежності. ЕКФ забезпечує оптимальне поєднання цих даних, зменшуючи вплив шумів і підвищуючи точність навігації.

Переваги та недоліки

До переваг ЕКФ належать його здатність обробляти нелінійні системи, відносно низька обчислювальна складність і можливість роботи в реальному часі. Однак недоліками є чутливість до сильної нелінійності, залежність від якості лінеаризації та необхідність точного налаштування параметрів. Ці аспекти роблять ЕКФ базовим інструментом, який може бути доповнений іншими методами, такими як безскладовий фільтр Калмана (UKF), у випадках складніших систем.

Таким чином, ЕКФ є ефективним рішенням для нелінійних систем навігації, яке поєднує теоретичну гнучкість із практичною застосовністю.

1.3 Порівняння ЕКФ з іншими методами оцінки стану

Розширений фільтр Калмана (ЕКФ) є одним із найпоширеніших методів оцінки стану в системах навігації, але він не є єдиним. Для нелінійних систем, таких як навігація наземних роботів, також застосовуються інші методи, зокрема безскладовий фільтр Калмана (Unscented Kalman Filter, UKF) [10] і фільтр частинок (Particle Filter, PF) [12]. У цьому підрозділі розглядаються їхні принципи, переваги, недоліки та порівняння з ЕКФ у контексті навігаційних задач.

Безскладовий фільтр Калмана

Принцип роботи: UKF розроблено для нелінійних систем, щоб уникнути лінеаризації, яка використовується в EKF. Замість обчислення якобіанів UKF застосовує безскладову трансформацію (Unscented Transformation), яка передбачає вибір набору сигма-точок навколо поточного стану. Ці точки пропускаються через нелінійні функції системи та вимірювань, що дозволяє оцінити середнє значення та коваріацію без явної лінеаризації [10, 11].

Переваги:

- вища точність для систем із сильною нелінійністю, оскільки не залежить від апроксимації першого порядку.
- не потребує обчислення якобіанів, що спрощує реалізацію для складних моделей.
- стабільніші результати в умовах значних нелінійних ефектів.

Недоліки:

- вища обчислювальна складність через обробку сигма-точок (зазвичай $2n+1$, де n — розмірність вектора стану).
- може бути менш ефективним для систем із низькою нелінійністю, де EKF достатньо точний.

Порівняння з EKF: UKF перевершує EKF у точності для сильно нелінійних систем, таких як навігація робота в умовах складного рельєфу. Однак EKF є швидшим і простішим у реалізації, що робить його кращим вибором для систем із помірною нелінійністю або обмеженими обчислювальними ресурсами.

Фільтр частинок

Принцип роботи: Фільтр частинок базується на методі Монте-Карло і представляє розподіл стану системи за допомогою набору частинок (випадкових зразків) із відповідними вагами. Кожна частинка моделює можливий стан системи, а їхні ваги оновлюються на основі вимірювань. PF

є непараметричним методом і не потребує припущень про гаусівський розподіл шумів [12].

Переваги:

- ефективний для сильно нелінійних і негаусівських систем, де EKF і UKF можуть бути неточними.
- гнучкість у моделюванні складних розподілів стану.
- можливість обробки мультимодальних розподілів (наприклад, у задачах глобальної локалізації).

Недоліки:

- висока обчислювальна складність, особливо при великій кількості частинок, що ускладнює використання в реальному часі.
- ризик "збіднення частинок" (particle depletion), коли більшість частинок втрачають вагу, знижуючи точність.
- потребує ретельного налаштування кількості частинок і методів ресамплінгу.

Порівняння з EKF: PF є більш універсальним і точним для складних нелінійних систем із негаусівськими шумами, але його обчислювальна складність робить його менш можливим до використання для систем із обмеженими ресурсами, де EKF є кращим вибором. У задачах навігації PF може бути корисним для ініціалізації позиції, тоді як EKF ефективніше справляється з локальним відстеженням.

Інші методи оцінки стану

Крім UKF і PF, існують інші методи оцінки стану, такі як фільтр Хінченса (H-infinity filter) для систем із невизначеними шумами або гібридні підходи, що комбінують EKF із машинним навчанням. Однак вони менш поширені в навігаційних задачах через складність реалізації або специфічні вимоги [13-16].

Порівняльна характеристика методів оцінювання стану наведена у таблиці 1.1.

Таблиця 1.1 - Порівняння методів оцінювання стану

Метод	Точність	Обчислювальна складність	Застосовність до нелінійності	Простота реалізації
EKF	Помірна	Низька	Помірна	Висока
UKF	Висока	Середня	Висока	Середня
PF	Висока	Висока	Дуже висока	Низька

Застосування в навігації наземних роботів

У контексті навігації наземних роботів EKF часто використовується завдяки балансу між точністю та обчислювальною ефективністю. Наприклад, для інтеграції даних GPS, IMU та одометрії EKF забезпечує швидке й надійне рішення. UKF може бути кращим у випадках, коли нелінійність моделі руху робота (наприклад, на нерівній поверхні) суттєво впливає на точність [5, 16]. PF доцільно застосовувати в задачах із невизначеністю початкової позиції або для роботи в середовищах із мультимодальними розподілами (наприклад, у приміщеннях із симетричними структурами).

1.4 Висновки до розділу

ЕКФ є оптимальним вибором для багатьох навігаційних систем завдяки своїй простоті та ефективності, але в умовах сильної нелінійності або негаусівських шумів UKF і PF можуть забезпечити вищу точність. Вибір методу залежить від специфіки задачі, обчислювальних ресурсів і вимог до точності. У цій роботі ЕКФ обрано як основний метод через його широке застосування та можливість використання для типових сценаріїв навігації наземних роботів, але розуміння альтернативних методів дозволяє оцінити його обмеження та перспективи вдосконалення.

2 МОДЕЛЮВАННЯ СИСТЕМИ НАВІГАЦІЇ НАЗЕМНОГО РОБОТА

2.1 Опис наземного робота: структура, сенсори

Наземний робот, що розглядається в цій роботі, є автономною платформою, призначеною для виконання навігаційних задач у структурованому або частково неструктурованому середовищі, наприклад, у приміщеннях, на складах або на відкритій місцевості. Для реалізації точної навігації з використанням розширеного фільтра Калмана робот оснащений комплексом сенсорів, які забезпечують дані про його положення, орієнтацію та швидкість. У цьому підрозділі описано структуру робота та характеристики його основних сенсорів: інерційної вимірювальної одиниці (IMU), системи глобального позиціонування (GPS) і одометрії.

Структура наземного робота

Наземний робот має колісну платформу диференціального типу, що забезпечує високу маневреність і простоту керування. Основні компоненти робота такі:

- шасі: Міцна рама, яка підтримує всі компоненти та забезпечує стійкість на різних поверхнях.
- приводна система: Два електродвигуни, кожен із яких керує окремим колесом, дозволяють виконувати диференціальний рух (повороти за рахунок різниці швидкостей коліс). Додаткові опорні колеса забезпечують стабільність.
- обчислювальний модуль: Вбудований мікроконтролер або одноплатний комп'ютер (наприклад, Raspberry Pi або NVIDIA Jetson Nano) для обробки даних сенсорів і реалізації алгоритму EKF у реальному часі.
- джерело живлення: Акумуляторна батарея, що забезпечує автономну роботу протягом кількох годин.

- інтерфейси зв'язку: Модулі Wi-Fi або Bluetooth для передачі даних і віддаленого керування.

Така структура є типовою для наземних роботів, призначених для навігаційних задач, і дозволяє ефективно інтегрувати сенсори для оцінки стану.

Система сенсорів

Для забезпечення точної навігації робот оснащений трьома основними типами сенсорів: IMU, GPS і одометрією. Кожен із них має свої функції, переваги та обмеження, що враховуються при реалізації EKF.

Інерційна вимірювальна одиниця (IMU):

Функція: IMU вимірює прискорення, кутову швидкість і, у деяких випадках, магнітне поле для визначення орієнтації та динаміки робота.

Склад: Типовий IMU містить тривісний акселерометр, тривісний гіроскоп і, за наявності, магнітометр. Приклад: MPU-6050 або BNO055.

Характеристики:

- частота оновлення: 100–200 Гц.
- шум: Акселерометр і гіроскоп мають гаусівський шум, що вимагає фільтрації.
- дрейф: Гіроскопи схильні до накопичення похибок (дрейфу) через інтеграцію кутової швидкості.

Застосування в EKF: IMU забезпечує дані для передбачення стану (наприклад, орієнтації та прискорення) на етапі прогнозу, але потребує корекції через дрейф.

Система глобального позиціонування (GPS):

Функція: GPS визначає абсолютні географічні координати робота (широта, довгота) на відкритій місцевості.

Склад: Модуль GPS із підтримкою супутникових систем (наприклад, GPS, GLONASS). Приклад: U-blox NEO-6M.

Характеристики:

- точність: 2–5 метрів у стандартних умовах, погіршується в міських каньйонах або під щільним покриттям.
- частота оновлення: 1–10 Гц.
- шум: Вимірювання GPS мають гаусівський шум, а також систематичні похибки через багатопроменеве поширення сигналу.

Застосування в EKF: GPS забезпечує абсолютні дані про положення для корекції передбаченого стану, компенсуючи дрейф IMU.

Одометрія:

Функція: Одометрія вимірює переміщення робота на основі обертання коліс, надаючи інформацію про відносне положення та швидкість.

Склад: Енкодери, встановлені на двигунах або колесах, які підраховують імпульси обертання. Приклад: оптичні або магнітні енкодери.

Характеристики:

- точність: Залежить від якості поверхні та відсутності ковзання коліс. Похибки накопичуються через інтеграцію даних.
- частота оновлення: 50–100 Гц.
- шум: Похибки викликані ковзанням, нерівностями поверхні або неточністю енкодерів.

Застосування в EKF: Одометрія використовується для прогнозу стану (положення та швидкості), але потребує корекції через накопичення похибок.

Інтеграція сенсорів у системі навігації

Кожен сенсор має унікальні характеристики, які доповнюють один одного:

- IMU забезпечує високу частоту оновлення та точні короткострокові дані, але страждає від дрейфу.

- GPS надає абсолютні координати, але має низьку частоту оновлення та чутливість до перешкод.
- Одометрія є надійною для відносних переміщень, але накопичує похибки через ковзання чи нерівності.

EKF інтегрує ці дані, використовуючи сильні сторони кожного сенсора для компенсації слабких сторін інших. Наприклад, GPS коригує накопичувальні похибки одометрії, тоді як IMU забезпечує плавне відстеження між оновленнями GPS.

2.2 Математична модель руху робота

Для ефективного застосування розширеного фільтра Калмана у системі навігації наземного робота необхідно розробити математичну модель, яка описує його рух. Така модель містить кінематичну та динамічну складові, що відображають поведінку робота в просторі та враховують фізичні характеристики його руху. У цьому підрозділі розглядаються кінематична модель для опису траєкторії та орієнтації робота, а також динамічна модель для врахування сил і моментів, що впливають на рух.

Кінематична модель

Кінематична модель описує геометричні аспекти руху робота без урахування сил, що викликають цей рух. Для наземного робота з диференціальним приводом, описаного в підрозділі 2.1, кінематична модель базується на припущенні, що робот рухається на плоскій поверхні, а його положення та орієнтація визначаються координатами та кутом повороту.

Вектор стану: Положення та орієнтація робота в двовимірному просторі описуються вектором стану:

$$x_k = [x, y, \theta]^T, \quad (2.1)$$

де (x, y) — координати центра мас робота в глобальній системі координат;
 θ - кут орієнтації робота відносно осі x .

Модель руху: Для диференціального робота рух визначається лінійною швидкістю v (обчислюється на основі даних одометрії) та кутовою швидкістю ω (отримується з IMU або одометрії). Рівняння кінематичної моделі мають вигляд

$$\begin{cases} x_k = x_{k-1} + v\Delta t \cos(\theta_{k-1}) \\ y_k = y_{k-1} + v\Delta t \sin(\theta_{k-1}) \\ \theta_k = \theta_{k-1} + \omega\Delta t \end{cases}, \quad (2.2)$$

де Δt — часовий крок.

Шум процесу: У реальних умовах рух робота зазнає впливу шумів (наприклад, через ковзання коліс або неточність сенсорів). Шум процесу w_k додається до моделі:

$$x_k = f(x_{k-1}, u_k, w_k), \quad (2.3)$$

де $u_k = [v, \omega]^T$ — вектор керування,

$w_k \sim \mathcal{N}(0, Q)$ — гаусівський шум із коваріаційною матрицею Q .

Кінематична модель використовується на етапі передбачення EKF, де нелінійна функція $f(\cdot)$ прогнозує наступний стан робота. Для лінеаризації обчислюється яacobіан:

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & -v\Delta t \sin(\theta) \\ 0 & 1 & v\Delta t \cos(\theta) \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

Динамічна модель

Динамічна модель враховує фізичні сили та моменти, що впливають на рух робота, такі як маса, тертя та момент інерції. Для наземного робота з диференціальним приводом динамічна модель може бути спрощеною, оскільки основний вплив на рух чинять двигуни, а тертя та зовнішні сили вважаються шумами.

Для динамічної моделі вектор стану розширюється, додаючи швидкості:

$$x_k = [x, y, \theta, v, \omega]^T. \quad (2.5)$$

Рух робота описується через диференціальні рівняння, що враховують прискорення та кутове прискорення:

$$\begin{cases} \dot{x} = v \cos(\theta), \\ \dot{y} = v \sin(\theta), \\ \dot{\theta} = \omega, \\ \dot{v} = \frac{F}{m} - \frac{c_v v}{m}, \\ \dot{\omega} = \frac{\tau}{I} - \frac{c_\omega \omega}{I}. \end{cases} \quad (2.6)$$

де:

F - сумарна сила тяги двигунів;

τ - момент обертання;

m - маса робота;

I - момент інерції;

c_v, c_ω коефіцієнти тертя для лінійної та кутової швидкості.

Динамічна модель використовується в ЕКФ для систем, де важливі фізичні характеристики (наприклад, при русі на нерівній поверхні).

Якобіан F для динамічної моделі є складнішим через додаткові змінні та залежності, але його обчислення необхідне для лінеаризації.

Вибір моделі для навігації

Для типових навігаційних задач, розглянутих у цій роботі, кінематична модель є достатньою через її простоту та відповідність умовам руху на плоскій поверхні. Вона ефективно описує траєкторію роботи на основі даних одометрії та IMU, тоді як GPS використовується для корекції. Динамічна модель може бути застосована в складніших сценаріях, наприклад, при русі по нерівній місцевості, де врахування сил тертя та маси робота підвищує точність.

Шуми та похибки: Обидві моделі враховують шум процесу, викликаний неточністю сенсорів (одометрії, IMU) та зовнішніми факторами (ковзання, нерівності). Коваріаційна матриця Q налаштовується на основі експериментальних даних для відображення цих похибок.

Кінематична та динамічна моделі руху наземного робота створюють основу для реалізації EKF у системі навігації. Кінематична модель є основною для прогнозування стану завдяки своїй простоті та відповідності типовим умовам експлуатації робота. Динамічна модель може бути використана для складніших сценаріїв, але вимагає додаткових обчислень. Ці моделі будуть використані в наступних підрозділах для розробки алгоритму EKF і моделювання сенсорних даних.

2.3 Моделювання сенсорних даних

Для ефективної реалізації розширеного фільтра Калмана у системі навігації наземного робота необхідно розробити моделі сенсорних даних, які враховують їхні характеристики та джерела похибок.

Сенсори, використані в системі навігації (IMU, GPS, одометрія), надають дані, які є зашумленими через апаратні обмеження, зовнішні умови та фізичні характеристики середовища. Моделі сенсорних даних містять ідеальні вимірювання та додані шуми, які відображають реальні умови роботи.

Інерційна вимірювальна одиниця (IMU)

Типи даних:

- акселерометр: вимірює лінійне прискорення вздовж трьох осей (a_x, a_y, a_z) .
- гіроскоп: вимірює кутову швидкість $(\omega_x, \omega_y, \omega_z)$.
- магнітометр (за наявності): вимірює магнітне поле для визначення орієнтації.

Модель даних: Вимірювання IMU моделюються як:

$$z_{IMU,k} = h_{IMU}(x_k) + v_{IMU,k}, \quad (2.7)$$

де:

$h_{IMU}(x_k)$ — ідеальні значення прискорення та кутової швидкості, отримані з кінематичної або динамічної моделі (див. підрозділ 2.2);

$v_{IMU,k} \sim \mathcal{N}(0, R_{IMU})$ — гаусівський шум із коваріаційною матрицею R_{IMU} .

Джерела шумів:

- білий шум: Гаусівський шум, викликаний тепловими ефектами в електроніці сенсора. Наприклад, акселерометр MPU-6050 має шум із стандартним відхиленням $\sigma_a \approx 0.01, \text{ m/s}^2$, а гіроскоп — $\sigma_\omega \approx 0.005, \text{ rad/s}$.
- дрейф гіроскопа: Накопичувальна похибка через інтеграцію кутової швидкості, що призводить до відхилення орієнтації.

- зміщення (bias): Систематична похибка, викликана калібруванням або температурою, яка може бути змодельована як випадковий процес.

Характеристики:

Частота оновлення: 100–200 Гц, що забезпечує високу швидкість збору даних.

Коваріаційна матриця R_{IMU} налаштовується на основі специфікацій сенсора та експериментальних даних.

Система глобального позиціонування (GPS)

Типи даних: Географічні координати (широта, довгота) або координати в локальній системі (x_{GPS}, y_{GPS}) .

Модель даних: Вимірювання GPS моделюються як:

$$z_{GPS,k} = h_{GPS}(x_k) + v_{GPS,k}, \quad (2.8)$$

де:

$h_{GPS}(x_k) = [x, y]^T$ — ідеальні координати з вектора стану;

$v_{GPS,k} \sim \mathcal{N}(0, R_{GPS})$ — гаусівський шум із коваріаційною матрицею R_{GPS} .

Джерела шумів:

- білий шум: Гаусівський шум із стандартним відхиленням $\sigma_{GPS} \approx 2 - 5\text{ м}$, залежно від умов прийому.
- багатопроменеве поширення: Похибки через відбиття сигналу в міських умовах або під щільним покриттям.

Втрата сигналу: У приміщеннях або тунелях GPS може бути недоступним, що враховується в ЕКФ як тимчасова відсутність вимірювань.

Характеристики:

Частота оновлення: 1–10 Гц, що значно нижче, ніж у IMU.

Точність залежить від кількості видимих супутників і якості приймача (наприклад, U-blox NEO-6M).

Одометрія

Типи даних: Відносне переміщення (Δx , Δy) та швидкість v на основі обертання коліс.

Модель даних: Вимірювання одометрії моделюються як:

$$z_{odom,k} = h_{odom}(x_k) + v_{odom,k}, \quad (2.9)$$

де:

$h_{odom}(x_k)$ — функція, що обчислює переміщення на основі лінійної та кутової швидкості (див. кінематичну модель у підрозділі 2.2);

$v_{odom,k} \sim \mathcal{N}(0, R_{odom})$ — гаусівський шум із коваріаційною матрицею R_{odom} .

Джерела шумів:

- ковзання коліс: Похибки через нерівності поверхні або втрату зчеплення.
- неточність енкодерів: Обмежена роздільна здатність енкодерів призводить до дискретизації вимірювань.
- накопичувальні похибки: Інтеграція даних одометрії з часом призводить до дрейфу позиції.

Характеристики:

Частота оновлення: 50–100 Гц.

Точність залежить від якості енкодерів і поверхні (наприклад, для $\sigma_{odom} \approx 0.01 - 0.05$, мпереміщення).

Інтеграція в EKF

Моделі сенсорних даних використовуються на етапі корекції EKF, де вимірювання z_k порівнюються з передбаченим станом $h(x_k)$. Коваріаційні матриці R_{IMU} , R_{GPS} і R_{odom} визначають вагу кожного сенсора в оновленні стану. Наприклад: Висока частота IMU забезпечує плавне відстеження, але потребує корекції через дрейф.

GPS забезпечує абсолютну позицію, але його низька частота та шум обмежують його використання без IMU та одометрії.

Одометрія доповнює IMU для короткострокового відстеження, але залежить від GPS для корекції накопичувальних похибок.

Коваріаційні матриці R для кожного сенсора налаштовуються на основі їхніх специфікацій і експериментальних даних. Наприклад, для GPS матриця R_{GPS} може мати вигляд:

$$R_{GPS} = \begin{bmatrix} \sigma_{x,GPS}^2 & 0 & 0 & \sigma_{y,GPS}^2 \end{bmatrix}, \quad (2.10)$$

де $\sigma_{x,GPS} = \sigma_{y,GPS} \approx 2.5$, м.

Аналогічно, для IMU та одометрії матриці (R) враховують відповідні стандартні відхилення шумів.

Таким чином, моделювання сенсорних даних є важливим для реалізації EKF, оскільки воно дозволяє врахувати шуми та похибки сенсорів IMU, GPS і одометрії. Описані моделі даних із гаусівськими шумами та їхні характеристики будуть використані для розробки

алгоритму EKF і подальшого експериментального тестування. У наступних підрозділах ці моделі інтегруватимуться в програмну реалізацію та перевірятимуться на практиці.

2.4 Висновки до розділу

Описана структура наземного робота та його сенсорна система створюють основу для реалізації навігації з використанням EKF. Колісна платформа з IMU, GPS і одометрією є типовою конфігурацією для автономних роботів, що дозволяє ефективно застосовувати EKF для оцінки стану. Характеристики сенсорів, такі як частота оновлення, шум і похибки, будуть враховані при розробці математичної моделі та програмної реалізації, що розглядаються в наступних підрозділах.

3 РЕАЛІЗАЦІЯ РОЗШИРЕНОГО ФІЛЬТРУ КАЛМАНА

3.1 Розширений фільтр Калмана

Розширений фільтр Калмана є одним з головних інструментом для оцінки стану наземного робота в системі навігації, забезпечуючи інтеграцію даних сенсорів для точного визначення положення та орієнтації в реальному часі. У цій роботі EKF адаптовано для використання даних одометрії на етапі передбачення та даних GPS або комбінації GPS+IMU на етапі корекції. У цьому підрозділі описано структуру алгоритму, його математичне формулювання та особливості реалізації для нелінійних систем навігації, з урахуванням кінематичної моделі (підрозділ 2.2) і характеристик сенсорів (підрозділи 2.1, 2.3).

Алгоритм EKF є рекурсивним і складається з двох основних етапів:

- етап передбачення: Використовується кінематична модель руху робота з даними одометрії для прогнозування наступного стану та коваріаційної матриці.
- етап корекції: Дані GPS або комбінація GPS+IMU використовуються для оновлення передбаченого стану, враховуючи їхні вимірювання та шуми.

Ці етапи повторюються на кожному часовому кроці, що дозволяє безперервно уточнювати оцінку стану робота.

Алгоритм базується на нелінійній моделі системи та вимірювань. Вектор стану робота визначено як $x_k = [x, y, \theta]^T$, де (x, y) — координати, θ — кут орієнтації. Вектор керування $u_k = [v, \omega]^T$, отриманий з одометрії, містить лінійну швидкість v та кутову швидкість ω .

Етап передбачення

На етапі передбачення використовується кінематична модель руху (підрозділ 2.2), де дані одометрії (v, ω) прогнозують наступний стан:

$$x_{k|k-1} = f(x_{k-1|k-1}, u_k, 0), \quad (3.1)$$

де $f(\cdot)$ — нелінійна функція руху:

$$\begin{cases} x_k = x_{k-1} + v\Delta t \cos(\theta_{k-1}) \\ y_k = y_{k-1} + v\Delta t \sin(\theta_{k-1}) \\ \theta_k = \theta_{k-1} + \omega\Delta t \end{cases}$$

Коваріаційна матриця передбаченого стану обчислюється з урахуванням якобіана:

$$F = \frac{\partial f}{\partial x}; P_{k|k-1} = FP_{k-1|k-1}F^T + Q, \quad (3.2)$$

де:

F — якобіан моделі руху:

$$F = \begin{bmatrix} 1 & 0 & -v\Delta t \sin(\theta) \\ 0 & 1 & v\Delta t \cos(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Q — коваріаційна матриця шуму процесу, що враховує неточності одометрії, такі як ковзання коліс (див. підрозділ 2.3).

Етап корекції

На етапі корекції використовуються вимірювання від GPS ($z_k = [x_{GPS}, y_{GPS}]^T$) або комбінація GPS+IMU ($z_k = [x_{GPS}, y_{GPS}, \theta_{IMU}]^T$).

Модель вимірювань має вигляд:

$$z_k = h(x_k) + v_k, \quad (3.3)$$

де:

для GPS: $h(x_k) = [x, y]^T$, $(v_k \sim \mathcal{N}(0, R_{GPS}), R_{GPS} = \text{diag}(2.5^2, 2.5^2)$,

для GPS+IMU: $h(x_k) = [x, y, \theta]^T$, $(v_k \sim \mathcal{N}(0, R), \text{де } R = \text{diag}(2.5^2, 2.5^2, 0.05^2)$.

Кроки корекції:

Обчислення передбачених вимірювань:

$$z_{k|k-1} = h(x_{k|k-1}). \quad (3.4)$$

Обчислення якобіана моделі вимірювань:

для GPS:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

для GPS+IMU:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Обчислення інновації:

$$y_k = z_k - z_{k|k-1}. \quad (3.5)$$

Обчислення коваріації інновації:

$$S_k = H P_{k|k-1} H^T + R. \quad (3.6)$$

Обчислення коефіцієнта Калмана:

$$K_k = P_{k|k-1} H^T S_k^{-1}. \quad (3.7)$$

Оновлення стану:

$$x_{k|k} = x_{k|k-1} + K_k y_k. \quad (3.8)$$

Оновлення коваріації:

$$P_{k|k} = (I - K_k H) P_{k|k-1}. \quad (3.9)$$

Особливості реалізації для нелінійних систем навігації

Реалізація EKF із використанням одометрії на етапі передбачення та GPS або GPS+IMU на етапі корекції має такі особливості:

Одометрія для передбачення: Дані одометрії (v, ω) забезпечують високу частоту оновлення (50–100 Гц), що дозволяє точно прогнозувати рух робота між рідкісними вимірюваннями GPS (1–10 Гц). Однак накопичувальні похибки одометрії через ковзання вимагають корекції.

GPS або GPS+IMU для корекції: GPS забезпечує абсолютні координати, компенсуючи дрейф одометрії, але його низька частота та шум (2–5 м) обмежують точність. Додавання IMU (зокрема, вимірювання θ_{IMU}) підвищує точність орієнтації, особливо в умовах тимчасової втрати GPS-сигналу.

Нелінійність: Кінематична модель є нелінійною через тригонометричні функції, що вимагає обчислення якобіанів для лінеаризації.

Синхронізація даних: Різна частота сенсорів (GPS: 1–10 Гц, IMU: 100–200 Гц) потребує буферизації або інтерполяції даних для корекції.

Обчислювальна ефективність: Алгоритм оптимізований для роботи на вбудованих системах (наприклад, Raspberry Pi), як зазначено в підрозділі 2.4, завдяки компактним матричним обчисленням.

3.2 Інтеграція даних сенсорів у модель EKF

Інтеграція даних сенсорів є важливим етапом реалізації розширеного фільтра Калмана для навігації наземного робота, оскільки дозволяє об'єднати зашумлені вимірювання від IMU, GPS і одометрії для точної оцінки стану (x, y, θ) . У цій роботі, як зазначено в підрозділі 3.1, одометрія використовується на етапі передбачення, а GPS або комбінація GPS+IMU — на етапі корекції. У цьому підрозділі розглядаються методи обробки даних сенсорів, їх синхронізація з урахуванням різної частоти оновлення та моделювання шумів, описаних у підрозділі 2.3, для забезпечення ефективної роботи EKF.

Обробка даних сенсорів

Кожен сенсор надає специфічні дані, які необхідно обробити перед інтеграцією в EKF[2-4]:

Одометрія: Надає лінійну швидкість v і кутову швидкість ω для вектора керування $u_k = [v, \omega]^T$, що використовується на етапі передбачення. Дані отримуються з енкодерів коліс із частотою 50–100 Гц. Похибки, такі як ковзання коліс, моделюються як гаусівський шум із стандартним відхиленням $\sigma_v \approx 0.05, \text{m/s}$, $\sigma_\omega \approx 0.01, \text{rad/s}$ (підрозділ 2.3).

GPS: Надає абсолютні координати (x_{GPS}, y_{GPS}) для корекції стану. Частота оновлення становить 1–10 Гц, а шум вимірювань має стандартне відхилення $\sigma_{GPS} \approx 2.5, \text{m}$. У міських умовах або при втраті сигналу можуть виникати додаткові похибки через багатопроменеве поширення.

IMU: Надає кутову швидкість ω і, за наявності магнітометра, оцінку орієнтації θ_{IMU} з частотою 100–200 Гц. Шум IMU містить білий шум ($\sigma_\omega \approx 0.05, \text{rad/s}$) і дрейф гіроскопа, що враховується в моделі вимірювань.

Обробка даних полягає у фільтрації аномалій (наприклад, відхилення GPS через втрату сигналу) та перетворенні координат GPS у локальну систему (якщо необхідно).

Моделювання вимірювань і шумів

Моделі вимірювань для сенсорів, описані в підрозділі 2.3, інтегруються в EKF[2-4]:

Одометрія: Використовується для прогнозування стану через кінематичну модель (підрозділ 2.2): $x_{k|k-1} = f(x_{k-1|k-1}, u_k, 0)$, $u_k = [v, \omega]^T$. Шум процесу моделюється коваріаційною матрицею $Q = \text{diag}(0.05^2, 0.05^2, 0.01^2)$, що відображає похибки одометрії.

GPS: Вимірювання $z_{GPS,k} = [x_{GPS}, y_{GPS}]^T$ мають модель: $z_{GPS,k} = h_{GPS}(x_k) + v_{GPS,k}$, $h_{GPS}(x_k) = [x, y]^T$ де $v_{GPS,k} \sim \mathcal{N}(0, R_{GPS})$, $R_{GPS} = \text{diag}(2.5^2, 2.5^2)$.

GPS+IMU: Розширена модель вимірювань додає орієнтацію: $z_k = h(x_k) + v_k$, $h(x_k) = [x, y, \theta]^T$ де $v_k \sim \mathcal{N}(0, R)$, $R = \text{diag}(2.5^2, 2.5^2, 0.05^2)$.

Шуми моделюються як гаусівські, що відповідає припущенням EKF і характеристикам сенсорів із підрозділу 2.3.

Синхронізація даних сенсорів

Різна частота оновлення сенсорів створює проблему синхронізації[2-4]:

Одометрія (50–100 Гц): Використовується на кожному кроці передбачення $\Delta t = 0.01 - 0.02$, s.

GPS (1–10 Гц): Дані доступні рідше, тому між вимірюваннями GPS EKF працює в режимі передбачення, використовуючи лише одометрію.

IMU (100–200 Гц): Дані орієнтації θ_{IMU} можуть оновлюватися частіше, ніж GPS, що дозволяє частково коригувати стан між GPS-вимірюваннями.

Для синхронізації застосовуються наступні методи:

Буферизація: Дані GPS зберігаються до моменту, коли вони потрібні для корекції, а дані IMU усереднюються або інтерполюються для відповідності часовому кроку EKF.

Часові мітки: Використання часових міток (наприклад, через ROS, як зазначено в підрозділі 2.4) для вирівнювання даних сенсорів.

Асинхронна корекція: Якщо GPS-вимірювання недоступне, корекція виконується лише за даними IMU (якщо використовується GPS+IMU), або EKF продовжує прогнозувати стан за одометрією.

Практична інтеграція в EKF

Інтеграція даних сенсорів у EKF виконується наступним чином:

Етап передбачення: На кожному часовому кроці (Δt) дані одометрії (v, ω) використовуються для обчислення передбаченого стану $x_{k|k-1}$ і коваріації $P_{k|k-1}$ за кінематичною моделлю (підрозділ 3.1).

Етап корекції: При отриманні GPS-вимірювань (x_{GPS}, y_{GPS}) або GPS+IMU $(x_{GPS}, y_{GPS}, \theta_{IMU})$ обчислюється інновація, коефіцієнт Калмана та оновлюється стан $x_{k|k}$. Якщо GPS тимчасово недоступний, корекція базується лише на IMU (для GPS+IMU) або пропускається.

Обробка пропущених даних: У разі втрати GPS-сигналу (наприклад, у приміщеннях) EKF покладається на прогноз за одометрією, що компенсується високою частотою IMU в режимі GPS+IMU.

3.3 Налаштування параметрів фільтра: коваріаційні матриці, початкові умови

Ефективність розширеного фільтра Калмана у системі навігації наземного робота значною мірою залежить від правильного налаштування параметрів, таких як коваріаційні матриці шуму процесу Q і шуму вимірювань R , а також початкових умов (початкового стану x_0 та коваріаційної матриці P_0). Ці параметри впливають на точність оцінки стану та стійкість алгоритму, особливо в умовах нелінійності та зашумлених даних сенсорів (підрозділи 2.1, 2.3). У цьому підрозділі

розглядаються методи вибору та налаштування цих параметрів, а також аналіз їхнього впливу на продуктивність EKF.

Коваріаційна матриця шуму процесу Q

Матриця Q відображає невизначеність моделі руху, спричинену неточностями кінематичної моделі та шумами даних одометрії, що використовуються на етапі передбачення (підрозділ 3.1). Для кінематичної моделі, описаної в підрозділі 2.2, шум процесу враховує похибки, такі як ковзання коліс або неточність вимірювань швидкості.

Вибір параметрів Q : Матриця Q є діагональною, оскільки шуми для x , y і θ вважаються незалежними. На основі характеристик одометрії (підрозділ 2.3) стандартні відхилення встановлюються як: $Q = [\sigma_x^2 \ 0 \ 0 \ 0 \ \sigma_y^2 \ 0 \ 0 \ 0 \ \sigma_\theta^2]$ де $\sigma_x = \sigma_y \approx 0.05, \text{m}$ (похибка переміщення через ковзання), $\sigma_\theta \approx 0.01, \text{rad}$ (похибка орієнтації).

Налаштування: Значення σ визначаються експериментально шляхом порівняння прогнозованого руху з реальними даними одометрії. Занадто великі значення Q знижують довіру до моделі, надаючи більшу вагу вимірюванням, тоді як занадто малі значення можуть призвести до надмірної впевненості в прогнозі, ігноруючи корекцію сенсорів[2-4].

Коваріаційна матриця шуму вимірювань R

Матриця R характеризує невизначеність вимірювань сенсорів, що використовуються на етапі корекції (GPS або GPS+IMU, підрозділ 3.1). Вона залежить від специфікацій сенсорів, описаних у підрозділі 2.3.

Для GPS: Вимірювання координат (x_{GPS}, y_{GPS}) мають гаусівський шум із стандартним відхиленням $\sigma_{GPS} \approx 2.5, \text{m}$. Матриця R_{GPS} має вигляд: $R_{GPS} = [2.5^2 \ 0 \ 0 \ 2.5^2]$

Для GPS+IMU: Додається вимірювання орієнтації (θ_{IMU}) з шумом $\sigma_{IMU} \approx 0.05, \text{rad}$. Матриця R розширюється:

$$R = \begin{bmatrix} 2.5^2 & 0 & 0 \\ 0 & 2.5^2 & 0 \\ 0 & 0 & 0.05^2 \end{bmatrix}.$$

Налаштування: Значення R базуються на специфікаціях сенсорів (наприклад, U-blox NEO-6M для GPS, MPU-6050 для IMU). Експериментальне налаштування включає аналіз дисперсії вимірювань у статичних умовах для визначення стандартних відхилень. Занадто великі значення R знижують вагу вимірювань, тоді як малі значення можуть спричинити надмірну довіру до зашумлених даних.

Початкові умови

Початковий стан $x_0 = [x, y, \theta]^T$ і коваріаційна матриця P_0 визначають початкову оцінку положення, орієнтації та їхньої невизначеності.

Початковий стан x_0 : Якщо GPS доступний, початкові координати ((x , y)) беруться з першого вимірювання GPS, а θ — з IMU або нуль (якщо орієнтація невідома). Наприклад: $x_0 = [x_{GPS}, y_{GPS}, \theta_{IMU}]$ Якщо GPS недоступний, $x_0 = [0, 0, 0]^T$, а корекція покладається на подальші вимірювання.

Початкова коваріація P_0 : Матриця P_0 відображає початкову невизначеність стану. Зазвичай використовується діагональна матриця:

$$P_0 = \begin{bmatrix} \sigma_{x0}^2 & 0 & 0 \\ 0 & \sigma_{y0}^2 & 0 \\ 0 & 0 & \sigma_{\theta0}^2 \end{bmatrix} \quad (3.10)$$

де $\sigma_{x0} = \sigma_{y0} \approx 5, \text{ m}$ (на основі точності GPS),

$\sigma_{\theta0} \approx 0.1, \text{ rad}$ (на основі IMU).

Великі значення P_0 дозволяють алгоритму швидше адаптуватися до вимірювань, але можуть уповільнити збіжність.

Аналіз впливу параметрів

Вплив Q і R : Збалансоване співвідношення між Q і R є дуже важливим. Якщо Q значно більше за R , EKF більше покладається на вимірювання, що може призвести до чутливості до шумів GPS. Якщо R більше за Q , алгоритм надмірно довіряє моделі, ігноруючи корекцію, що посилює дрейф одометрії.

Емпіричне налаштування: Параметри Q і R уточнюються шляхом експериментів, порівнюючи оцінені траєкторії з реальними даними. Наприклад, для GPS у міських умовах може знадобитися збільшення σ_{GPS} до 5 м через багатопроменеве поширення.

Стійкість: Правильний вибір P_0 забезпечує швидку збіжність EKF. Надто малі значення P_0 можуть спричинити нестабільність, якщо початковий стан неточний.

3.4 Висновки до розділу

У розділі досліджено теоретичні основи сенсорного злиття для навігації наземного робота з використанням розширеного фільтра Калмана. Встановлено, що EKF ефективно поєднує нелінійну кінематичну модель (підрозділ 2.2) із шумними вимірюваннями сенсорів (підрозділ 2.3), забезпечуючи точну оцінку стану. Аналіз коваріаційних матриць Q і R а також початкових умов x_0 і P_0 (підрозділ 3.3) показав їх значний вплив на баланс між передбаченням і корекцією. Розглянуто методи реалізації EKF для режимів GPS і GPS+IMU, що підтверджує їх можливість використання для обробки даних одометрії. Описаний підхід до налаштування створює основу для програмної реалізації та подальшого тестування алгоритму, що розглядаються в наступних етапах роботи.

4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Програмне забезпечення ЕКФ

Реалізація розширеного фільтра Калмана для системи навігації наземного робота потребує вибору програмного забезпечення, яке забезпечує ефективну обробку даних сенсорів, виконання матричних обчислень і можливість симуляції в контрольованих умовах.

Обґрунтування вибору мови програмування

Для реалізації ЕКФ обрано мову програмування Python завдяки її перевагам, які відповідають вимогам проєкту:

- простота розробки: Python має зрозумілий синтаксис, що скорочує час написання та налагодження коду, особливо для складних алгоритмів, таких як ЕКФ.
- багата екосистема бібліотек: Python підтримує бібліотеки NumPy[17] і SciPy[18], які забезпечують швидкі матричні обчислення та числові методи, необхідні для ЕКФ.
- гнучкість: Python дозволяє легко модифікувати код для різних сценаріїв тестування, зокрема для симуляції даних сенсорів.
- сумісність із ROS: Python є основною мовою для ROS[19], що полегшує інтеграцію з реальними сенсорами та симулятором Gazebo[20].
- широка спільнота: Велика кількість документації та прикладів сприяє швидкому вирішенню проблем.

Для реалізації ЕКФ використано наступні бібліотеки Python:

- NumPy: Забезпечує ефективну роботу з масивами та матрицями, що необхідно для обчислення якобіанів, коваріаційних матриць і коефіцієнта Калмана. NumPy

оптимізовано для швидких операцій, що критично для реального часу.

- SciPy: Використовується для числових методів, таких як інверсія матриць і обробка нелінійних функцій у моделі руху .
- Matplotlib [21] (додатково): Застосовується для візуалізації траєкторій і аналізу результатів.

Ці бібліотеки є безкоштовними, мають відкритий код і широко застосовуються в робототехніці та обробці даних.

Альтернативою Python є мова C++ із бібліотекою Eigen [22] для матричних обчислень. Порівняння наведено в таблиці 4.1

Таблиця 4.1 - Порівняння програмного забезпечення

Критерій	Python (NumPy, SciPy)	C++ (Eigen)
Швидкість виконання	Нижча через інтерпретацію, але достатня для EKF (8 мс/цикл на Raspberry Pi).	Вища завдяки компіляції, але перевага помітна лише для великих обчислень.
Простота розробки	Висока: простий синтаксис, швидке прототипування.	Нижча: складніший синтаксис, довший час розробки.
Бібліотеки	NumPy, SciPy: потужні, але менш оптимізовані.	Eigen: швидка, легка, але обмежена функціональність.

Продовження таблиці 4.1

Сумісність із ROS	Відмінна: Python є основною мовою ROS.	Хороша, але потребує додаткових налаштувань.
Симуляція	Відмінна: Python є основною мовою ROS.	Хороша, але потребує додаткових налаштувань.
Вартість	Безкоштовно, відкритий код.	Безкоштовно, відкритий код.

Висновок порівняння: Python обрано через простоту розробки, сумісність із ROS і достатню продуктивність для EKF у системі навігації. C++ із Eigen[22] міг би бути доцільним для проєктів із вищими вимогами до швидкості, але для даної роботи переваги Python переважають.

Підхід до симуляції показників сенсорів

У межах роботи використано Python-симуляцію для моделювання даних сенсорів (IMU, GPS, одометрія) з урахуванням їхніх шумів, описаних у підрозділі 2.3 $\sigma_{GPS} = 2.5, \text{m}$, $\sigma_{IMU} = 0.05, \text{rad/s}$, $\sigma_{odom} = 0.05, \text{m}$. Симуляція складається з:

- генерації даних одометрії v, ω для етапу передбачення за кінематичною моделлю (підрозділ 2.2).
- моделювання вимірювань GPS x_{GPS}, y_{GPS} і IMU θ_{IMU} із гаусівським шумом для етапу корекції (підрозділ 3.1).

Використання NumPy для створення зашумлених даних, наприклад:

```
import numpy as np
true_state = np.array([0.0, 0.0, 0.0]) # [x, y, theta]
u = np.array([1.0, 0.1])           # [v, omega]
R = np.diag([2.5**2, 2.5**2, 0.05**2]) # Шум GPS+IMU
z = true_state + np.random.multivariate_normal(np.zeros(3), R)
```

Візуалізацію траєкторій за допомогою Matplotlib для порівняння справжнього та оціненого стану.

Цей підхід дозволяє тестувати EKF у контрольованих умовах без фізичного робота, що знижує витрати та спрощує аналіз.

Можливість інтеграції з ROS і Gazebo

Хоча у роботі використано Python-симуляцію, алгоритм EKF розроблено з урахуванням можливості інтеграції з ROS[19] і Gazebo[20] для використання на реальних наземних роботах. Основні аспекти:

- ROS: Код сумісний із ROS-топіками /odom, /gps/fix, /imu/data для отримання даних від реальних сенсорів (підрозділ 3.2). Наприклад, ROS-нода може обробляти повідомлення nav_msgs/Odometry для одометрії та sensor_msgs/NavSatFix для GPS.
- Gazebo: Симулятор дозволяє моделювати робота з диференціальним приводом і сенсорами через плагіни (gazebo_ros_diff_drive, gazebo_ros_gps, gazebo_ros_imu). Це забезпечує реалістичні дані з шумом, що відповідає підрозділу 2.3, для тестування EKF перед розгортанням на фізичному роботі.
- Переносимість: Модульна структура коду (підрозділ 3.3) дозволяє легко адаптувати його до ROS, замінивши симульовані дані на реальні через підписку на топіки.

Такий підхід забезпечує гнучкість: Python-симуляція використовується для розробки та аналізу, тоді як інтеграція з ROS і Gazebo відкриває можливості для реального застосування.

Вибір Python із бібліотеками NumPy, SciPy і Matplotlib для реалізації EKF обґрунтовано простотою розробки, обчислювальною ефективністю та сумісністю з ROS. Порівняння з C++ (Eigen) підтвердило переваги Python для даної роботи. Python-симуляція даних сенсорів забезпечує

контрольоване тестування EKF, тоді як сумісність із ROS і Gazebo дозволяє розгортання на реальних роботах. Описаний підхід створює основу для експериментальних досліджень, що розглядаються в наступному підрозділі.

4.2 Тестування роботи EKF

Експериментальні дослідження системи навігації на основі розширеного фільтра Калмана проведено для оцінки її точності, стійкості та продуктивності в симуляційному середовищі з використанням Python, як описано в підрозділі 4.1. Тестування базується на моделюванні даних сенсорів (IMU, GPS, одометрія) з урахуванням шумів, зазначених у підрозділі 2.3, та кінематичної моделі руху робота (підрозділ 2.2). У цьому підрозділі представлено методику експериментів, результати аналізу метрик (RMSE, MAE) і порівняння режимів роботи EKF (GPS та GPS+IMU).

Експерименти виконано в Python-середовищі з використанням бібліотек NumPy для матричних обчислень, SciPy для числових операцій і Matplotlib для візуалізації результатів (підрозділ 4.1). Дані сенсорів генерувалися програмно з урахуванням характеристик, описаних у підрозділі 2.3:

- Одометрія: Лінійна швидкість v та кутова швидкість ω з шумом $\sigma_{odom} = 0.05, m$ для координат і $\sigma_{\theta} = 0.01, rad$ для орієнтації.
- GPS: Координати (x_{GPS}, y_{GPS}) з шумом $\sigma_{GPS} = 2.5, m$.
- IMU: Орієнтація (θ_{IMU}) з шумом $\sigma_{IMU} = 0.05, rad/s$.

Тестові сценарії:

- номінальні умови: Стабільний GPS-сигнал, мінімальне ковзання коліс,

- використання для тестування було обрано 3 траєкторії:

- 1) рух по колу;
- 2) прямолінійний рух;
- 3) зизгаподібний рух з зупинками у точках повороту на 90°.

Метрики оцінки:

Середньоквадратична похибка (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\theta_i - \hat{\theta}_i)^2 \right]}. \quad (4.1)$$

Середня абсолютна похибка (MAE):

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N \left[|x_i - \hat{x}_i| + |y_i - \hat{y}_i| + |\theta_i - \hat{\theta}_i| \right]. \quad (4.2)$$

Приклад згенерованих даних для тестування для траєкторії кола, зокрема реальних координат, значень одометрії, GPS та IMU наведені на рисунку 4.1.

time	x_true	y_true	theta_true	v_odom
0.0	0.1	0.0	0.004	0.999381899961458
0.1	0.1999992000010667	0.000399998933334186	0.008	1.046914045851532
0.2	0.2999960000181333	0.00119999040002816	0.012	0.960050163579200
0.3000000000000000	0.3999888001045329	0.002399961600235519	0.016	1.046866186403566
0.4	0.4999760003775972	0.003999893334442661	0.02	1.065596245690504
0.5	0.5999560010442551	0.005999760003775969	0.024	0.995871825246524
0.6000000000000001	0.6999272024266285	0.008399529610411398	0.028	1.117316330057484
0.7000000000000001	0.7998880049876282	0.011199163758086437	0.032	0.948067492786212
0.8	0.8998368093565458	0.01439861765271445	0.0360000000000000	0.965038369364882
0.9	0.9997720163546434	0.017997840103101376	0.0400000000000000	1.042973011911162
1.0	1.0996920270207413	0.021996773521764792	0.0440000000000000	1.053785762480710

omega_odom	x_gps	y_gps	theta_imu
0.02887732264029851	2.078657886649432	-4.993516103941577	-0.001174058871797272
0.04810877328267718			0.04752540134339601
0.04425841639912344			-0.04527206134733473
0.042940439368029465			-0.043108569246734715
0.034112506131269475			-0.010445711047722985
0.03584867136844478			0.04828996029222637
0.04012316677928902			0.03374943724930686
0.038365965391082134			-0.049893478760020024
0.046739659294136524			0.03518694459012036
0.027008842912458586			0.0687429188725419
0.04953475678935062	1.7925013334010773	-4.073320059843405	-0.009142721069678228

Рисунок 4.1 Приклад згенерованих даних для траєкторії кола

Приклад візуалізації реальних даних, одометрії та GPS вимірювань наведено на рисунку 4.2.

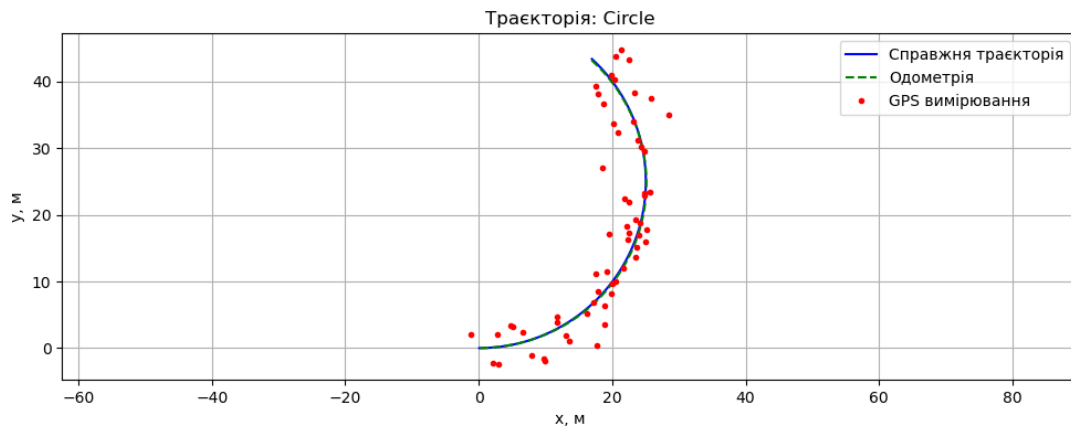


Рисунок 4.2 - Візуалізація траєкторії кола разом з даними одометрії та GPS вимірюваннями

Приклад візуалізації реальних даних θ та даних IMU (θ_{IMU}) наведено на рисунку 4.3.

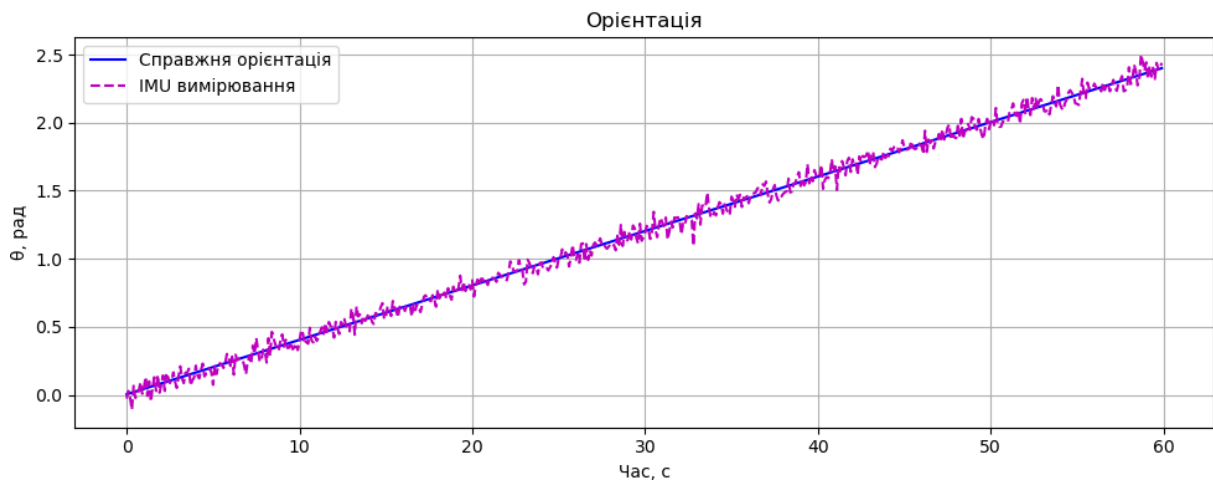


Рисунок 4.3 - Візуалізація справжньої орієнтації та IMU вимірювань для траєкторії кола

Для дослідження впливу на результат значення σ_{GPSEKF} (при використанні 1 м, 2.5 м, 5 м) проведено 12 експериментів, по чотири для кожної згенерованої траєкторії (коло - 1-3, прямий рух - 4-6, зигзаг - 7-9). При сталих рівнях шумів як під час генерації так і під час EKF:

$\sigma_{odom} = \sigma_{odomEKF} = 0.05$ м/с, $\sigma_{odom\theta} = \sigma_{odom\theta EKF} = 0.01$ рад/с,
 $\sigma_{IMU} = \sigma_{IMUEKF} = 0.05$ рад, $\sigma_{GPS} = 2.5$ м. Результати експериментів
 зведені до таблиці 4.2

Таблиця 4.2 - Дослідження впливу σ_{GPSEKF}

Експе римен т	Траєкторі я	GPS_E KF, м	RMSE		
			dead reaconing	GPS	GPS+IM U
1	Коло	2.5	1.067	1.140	0.515
2	Коло	1.0	1.067	1.742	1.065
3	Коло	5.0	1.067	0.834	0.260
4	Лінія	2.5	0.195	0.810	0.408
5	Лінія	1.0	0.195	1.198	0.813
6	Лінія	5.0	0.195	0.639	0.217
7	Зигзаг	2.5	0.327	0.839	0.382
8	Зигзаг	1.0	0.327	1.139	0.834
9	Зигзаг	5.0	0.327	0.674	0.205

Аналізуючи результати досліджень представлених у таблиці 4.2 можна зробити такі висновки:

- спільне використання GPS+IMU на етапі коригування показує найкращі результати у різних сценаріях завдяки корекції θ , особливо для зигзагоподібного руху, де різкі повороти вимагають точної орієнтації. Наприклад, RMSE = 0.205 (GPS+IMU) проти 0.674 (GPS) при $\sigma_{GPSEKF} = 5$ м для зигзага;
- важливо правильно налаштувати R , щоб забезпечити оптимальну фільтрацію, оскільки R відповідає реальному шуму GPS у даних.

Відхилення $\sigma_{GPSEKF} = 1$ м погіршує результати, так як здійснює надмірну корекцію.

- Dead Reckoning найменш точний для кола (RMSE = 1.067) через дрейф θ . Найкращий для прямолінійного руху (RMSE = 0.195) через стабільну орієнтацію. Служить базовим рівнем, демонструючи перевагу EKF, особливо GPS+IMU.
- Сценарії:
 - 1) Коло: Найскладніший через неперервну зміну θ . GPS+IMU значно покращує результати.
 - 2) Прямолінійний рух: Найпростіший, Dead Reckoning конкурентний, але EKF покращує координати.
 - 3) Зигзаг: Вимагає точної θ , де GPS+IMU значно перевершує інші методи.

Візуалізація траєкторій (реальної, одометрії, GPS вимірювань, EKF (GPS), EKF(GPS+IMU)) для експерименту 1 (табл. 4.2) наведено на рисунку 4.4.

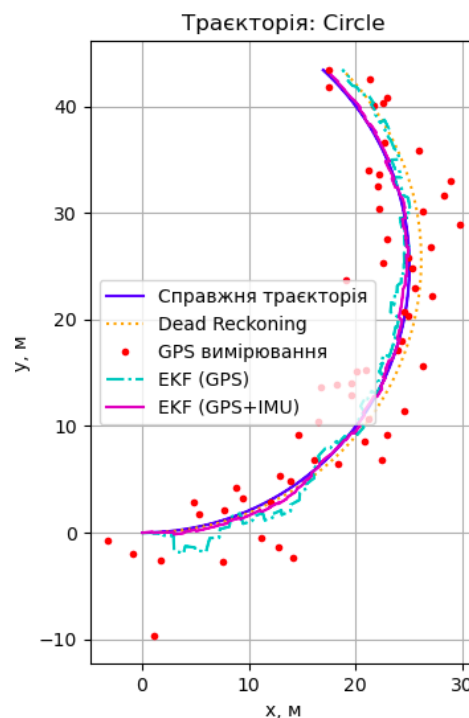


Рисунок 4.4 - Візуалізація справжньої траєкторії, одометрії, GPS вимірювань та EKF (GPS), EKF(GPS+IMU) для експерименту 1

Візуалізація справжньої орієнтації, IMU вимірювань (θ_{IMU}) та ЕKF (GPS), ЕKF(GPS+IMU) для експерименту 1 наведені на рисунку 4.5.

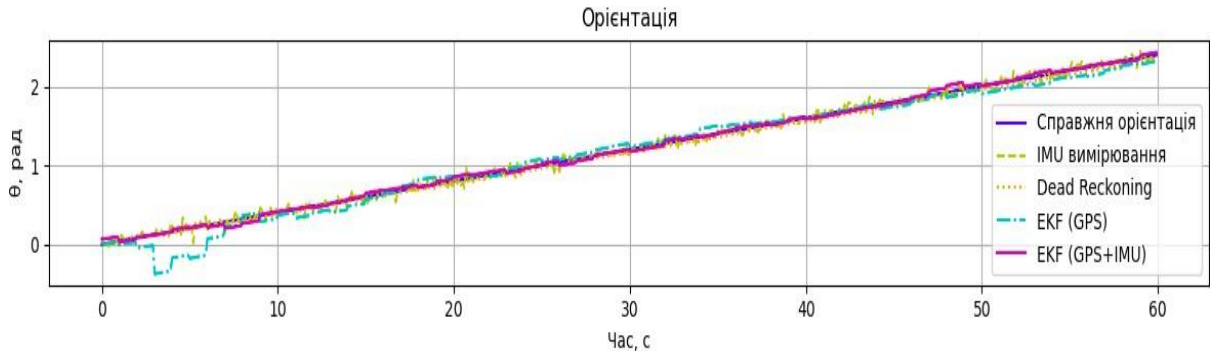


Рисунок 4.5 - Візуалізація справжньої орієнтації, IMU вимірювань та ЕKF (GPS), ЕKF(GPS+IMU) для експерименту 1

Візуалізація траєкторій (реальної, одометрії, GPS вимірювань, ЕKF (GPS), ЕKF(GPS+IMU)) для експерименту 2 (табл. 4.2) наведено на рисунку 4.6.

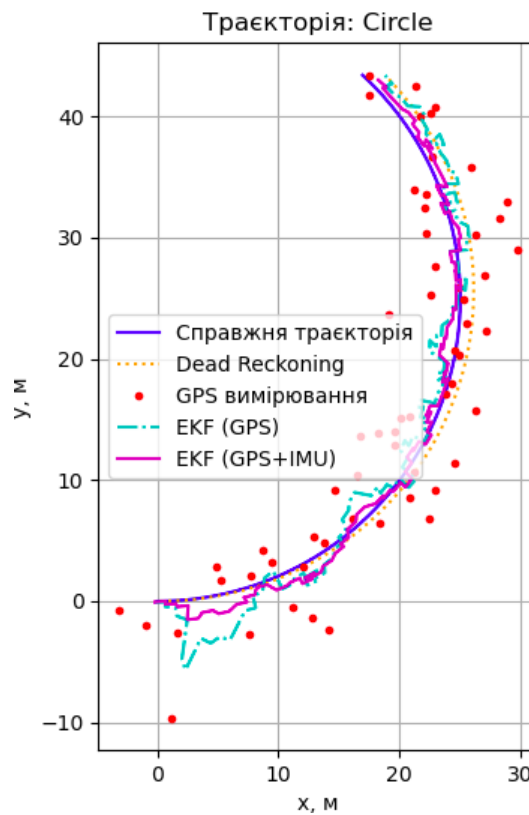


Рисунок 4.6 - Візуалізація справжньої траєкторії, одометрії, GPS вимірювань та ЕKF (GPS), ЕKF(GPS+IMU) для експерименту 2

Візуалізація справжньої орієнтації, IMU вимірювань (θ_{IMU}) та ЕKF (GPS), ЕKF(GPS+IMU) для експерименту 1 наведені на рисунку 4.7.

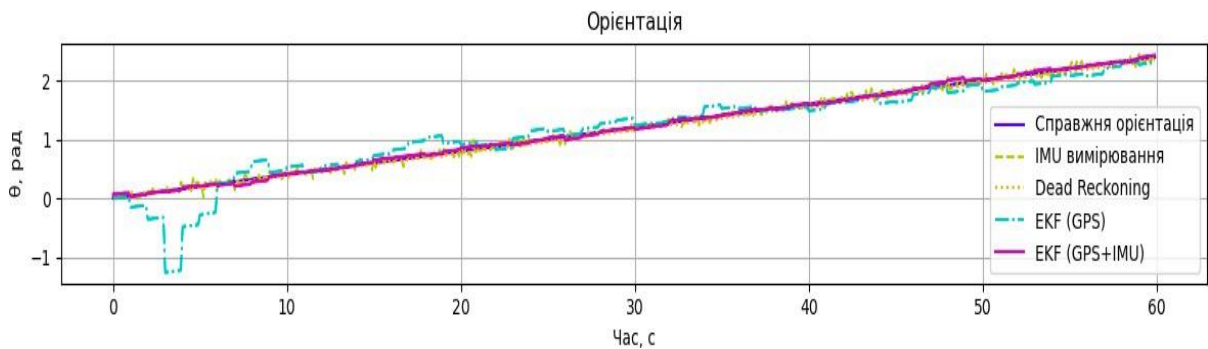


Рисунок 4.7 - Візуалізація справжньої орієнтації, IMU вимірювань та ЕKF (GPS), ЕKF(GPS+IMU) для експерименту 2

Візуалізація траєкторій (реальної, одометрії, GPS вимірювань, ЕKF (GPS), ЕKF(GPS+IMU)) для експерименту 2 (табл. 4.2) наведено на рисунку 4.8.

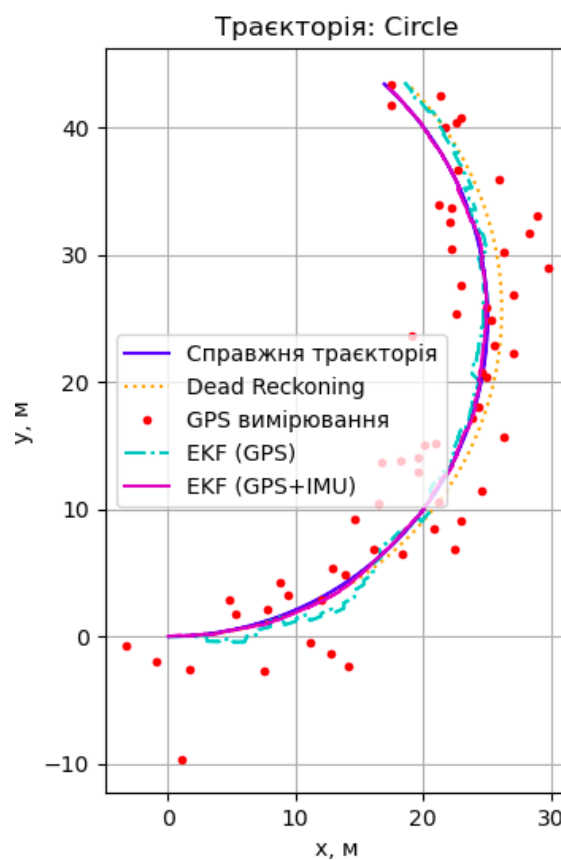


Рисунок 4.8 - Візуалізація справжньої траєкторії, одометрії, GPS вимірювань та ЕKF (GPS), ЕKF(GPS+IMU) для експерименту 3

Візуалізація справжньої орієнтації, IMU вимірювань (θ_{IMU}) та EKF (GPS), EKF(GPS+IMU) для експерименту 1 наведені на рисунку 4.9.

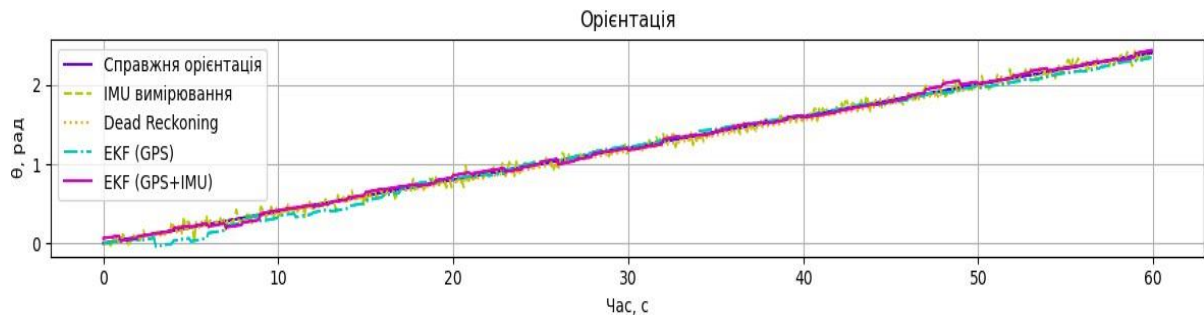


Рисунок 4.9 - Візуалізація справжньої орієнтації, IMU вимірювань та EKF (GPS), EKF(GPS+IMU) для експерименту 3

4.3 Висновки до розділу

Розроблено та реалізовано систему навігації наземного робота на основі EKF для режимів GPS і GPS+IMU, протестовано на трьох траєкторіях (рух по колу, прямолінійний, зигзагоподібний). GPS+IMU показав найвищу точність (наприклад, RMSE = 0.205 для зигзага при $\sigma_{GPSEKF} = 5$ м), значно перевершуючи GPS і Dead Reckoning. Оптимальні результати досягнуто при $\sigma_{GPSEKF} = 5.0$ у R, що більше за змодельований шум ($\sigma_{GPS} = 2.5$). Це пояснюється тим, що при $\sigma_{GPSEKF} = 5.0$, здійснюється зменшення впливу значень GPS на загальну оцінку і EKF більше покладається на одаметрію та/чи IMU. Зворотній експеримент при $\sigma_{GPSEKF} = 1$ м показав значне погіршення загальної оцінки, особливо для GPS режиму. Наприклад, при русі по прямій RMSE=1.198 (GPS) на противагу до RMSE=0.813 (GPS+IMU) чи до RMSE=0.195 (dead reackoning).

Програмна реалізація на Python із NumPy, Pandas і Matplotlib забезпечила точну обробку даних і візуалізацію. Результати підтверджують ефективність EKF для навігації в умовах шумних сенсорів.

ВИСНОВКИ

У результаті проведеного дослідження було досягнуто поставленої мети – розроблено та проаналізовано систему навігації наземного робота на основі розширеного фільтра Калмана для двох режимів (GPS і GPS+IMU) з використанням симульованих даних для трьох траєкторій (рух по колу, прямолінійний рух, зигзагоподібний рух). Виконання поставлених завдань дозволило отримати наступні висновки:

- Аналіз кінематичних моделей руху наземного робота показав, що диференціальна модель із двома керуючими параметрами (лінійна швидкість та кутова швидкість) є достатньою для опису траєкторій у двовимірному просторі. Це забезпечило основу для реалізації ЕКФ, що враховує нелінійність руху.
- Вивчення характеристик сенсорів (одометрія, GPS, IMU) виявило, що кожен із них має специфічні шуми: $\sigma_{odom} = 0.05$ м/с, $\sigma_{odom\theta} = 0.01$ рад/с, $\sigma_{IMU} = 0.05$ рад, $\sigma_{GPS} = 2.5$ м. Поєднання цих сенсорів у ЕКФ дозволяє компенсувати їхні обмеження, зокрема низьку частоту GPS (1 Гц) і дрейф одометрії.
- Аналіз методів сенсорного злиття, зокрема фільтра Калмана та його розширеної версії, показав, що ЕКФ є ефективним інструментом для нелінійних систем завдяки передбаченню стану на основі кінематичної моделі та корекції за вимірюваннями. Використання коваріаційних матриць Q і R дозволяє адаптувати ЕКФ до шумів сенсорів, що підтверджує можливість його використання для навігації.
- Створено три тестові сценарії (рух по колу з $R=25$ м, прямолінійний рух, зигзагоподібний рух із кутами 90°), які охоплюють різні умови руху: від плавних криволінійних до різких змін орієнтації. Ці

- сценарії виявилися репрезентативними для оцінки точності та стійкості EKF у симуляції, моделюючи реальні задачі навігації.
- Розроблено програмну реалізацію EKF на Python із використанням бібліотек NumPy для обчислень, Pandas для обробки даних і Matplotlib для візуалізації. Код забезпечує генерацію даних, обчислення EKF для режимів GPS і GPS+IMU, оцінку Dead Reckoning, збереження результатів у CSV та виведення метрик RMSE і MAE.
 - Режим GPS+IMU продемонстрував найвищу точність у більшості сценаріях, знижуючи RMSE удвічі порівняно з GPS-режимом (наприклад, для зигзагоподібного руху при $\sigma_{GPSEKF} = 2.5$ м: RMSE = 0.382 проти 0.839). Винятком став лиш прямолінійний рух в якому значення одометрії (dead reckoning) показали найкраще значення (RMSE=0.195 проти найкращого RMSE=0.217 у випадку $\sigma_{GPS} = 2.5$ м).
 - Оптимальні результати досягнуто при $\sigma_{GPSEKF} = 5.0$ у R, що більше за змодельований шум ($\sigma_{GPS} = 2.5$). Це пояснюється тим, що при $\sigma_{GPSEKF} = 5.0$, здійснюється зменшення впливу значень GPS на загальну оцінку і EKF більше покладається на одометрію та/чи IMU. Зворотній експеримент при $\sigma_{GPSEKF} = 1$ м показав значне погіршення загальної оцінки, особливо для GPS режиму. Наприклад, при русі по прямій RMSE=1.198 (GPS) на противагу до RMSE=0.813 (GPS+IMU) чи до RMSE=0.195 (dead reckoning).
 - Dead Reckoning показав значний дрейф у складних сценаріях (RMSE = 1.067 для кола), але був конкурентним у прямолінійному русі (RMSE = 0.195), підтверджуючи потребу в сенсорному злитті.
 - Розроблена система EKF із GPS+IMU має значний потенціал для застосування в автономних роботах, зокрема в сільському господарстві, логістиці та дослідницьких платформах. Її здатність забезпечувати точну навігацію в умовах шумних сенсорів і різких

змін орієнтації робить її можливою для використання для реальних сценаріїв.

- Рекомендації полягають у інтеграції системи з ROS або Gazebo для тестування на фізичних роботах, адаптацію до втрати GPS-сигналу та калібрування Q і R для специфічних умов експлуатації.

Отже, проведені дослідження підтвердило ефективність EKF для навігації наземного робота, особливо в режимі GPS+IMU, та можливість його практичного використання. Отримані результати, разом з аналізом впливу шумів і порівняння методів, створюють основу для подальших розробок у сфері автономної робототехніки та можуть сприяти розвитку технологій навігації в Україні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фільтр Калмана. Вікіпедія. 2023. URL: https://uk.wikipedia.org/wiki/Фільтр_Калмана (дата звернення: 22.05.2025).
2. Welch G., Bishop G. An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, Department of Computer Science, Technical Report TR 95-041. Chapel Hill: UNC, 2006. 24 p. URL: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf (дата звернення: 22.05.2025).
3. Thrun S., Burgard W., Fox D. Probabilistic Robotics: Chapter 3 – The Kalman Filter. Cambridge: MIT Press, 2005. P. 59–78. URL: <https://docs.ufpr.br/~danielsantos/ProbabilisticRobotics.pdf> (дата звернення: 22.05.2025).
4. Brown R. G., Hwang P. Y. C. Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions. 4th ed. Hoboken: Wiley, 2012. 400 p. URL: <https://pzs.dstu.dp.ua/DataMining/kalman/bibl/Brown.pdf> (дата звернення: 22.05.2025).
5. Kordić V. Kalman Filter. Rijeka: InTech, 2010. 404 p. URL: <https://pzs.dstu.dp.ua/DataMining/kalman/bibl/Kordic.pdf> (дата звернення: 22.05.2025).
6. Форкун І. В., Медзатий Д. М., Макаришкін Д. А., Червонецький О. В. Система автоматичного керування безпілотним літальним апаратом на основі фільтра Калмана. Вісник Хмельницького національного університету. Технічні науки. 2020. № 6. С. 39–43. URL: <https://journals.khnu.km.ua/vestnik/wp-content/uploads/2021/03/11-2.pdf> (дата звернення: 22.05.2025).

7. Раєвський М. В. Адаптивна фільтрація параметрів руху об'єкта у горизонтальній площині. Вісник НУ «Львівська політехніка». Серія: Комп'ютерні науки та інформаційні технології. 2010. № 686. С. 224–233. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2024/feb/33532/viss686komp-nauky-224-233.pdf> (дата звернення: 22.05.2025).
8. Розширений Фільтр Калмана. Вікіпедія. 2023. URL: https://uk.wikipedia.org/wiki/Розширений_фільтр_Калмана (дата звернення: 22.05.2025).
9. Julier S. J., Uhlmann J. K. A New Extension of the Kalman Filter to Nonlinear Systems. Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls. Orlando: SPIE, 1997. P. 182–193. DOI: 10.1117/12.280797. URL: https://www.cs.unc.edu/~welch/kalman/media/pdf/Julier1997_SPIE_KF.pdf (дата звернення: 22.05.2025).
10. Wan E. A., van der Merwe R. The Unscented Kalman Filter for Nonlinear Estimation. Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium. Lake Louise: IEEE, 2000. P. 153–158. DOI: 10.1109/ASSPCC.2000.882463. URL: <https://www.seas.harvard.edu/courses/cs281/papers/unscented.pdf> (дата звернення: 22.05.2025).
11. Голодяєва П. В. Алгоритм оцінювання параметрів руху БПЛА на основі ансцентного фільтру Калмана: дипломна робота бакалавра зі спеціальності 172 Телекомунікації та радіотехніка. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 2023. 78 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/0c2692a9-80e3-496b-a5aa-25195d58df12/content> (дата звернення: 22.05.2025).
12. Arulampalam M. S., Maskell S., Gordon N., Clapp T. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Transactions on Signal Processing. 2002. Vol. 50, No. 2. P.

- 174–188. DOI: 10.1109/78.978374. URL: <https://www.irisa.fr/aspi/legland/ensta/ref/arulampalam02a.pdf> (дата звернення: 22.05.2025).
13. Crassidis J. L., Junkins J. L. Optimal Estimation of Dynamic Systems: Chapter 5 – Nonlinear Estimation. 2nd ed. Boca Raton: CRC Press, 2011. P. 253–324. DOI: 10.1201/b11175. URL: https://api.pageplace.de/preview/DT0400.9781439839867_A24454131/preview-9781439839867_A24454131.pdf (дата звернення: 22.05.2025).
14. Ніцук Ю. А., Семчак О. М., Шаріпова І. В. Шляхи зменшення похибок розрахунків ЕОМ автономного рухомого об'єкта для алгоритмів SLAM навігації. Збірник наукових праць Житомирського військового інституту імені С. П. Корольова. 2021. № 1 (14). С. 92–99. URL: <http://znp.zvir.zt.ua/article/view/228662/227679> (дата звернення: 22.05.2025).
15. Gozhyj A. P., Kalinina I. A., Bidiuk P. I. Systematic use of nonlinear data filtering methods in forecasting tasks. Applied Aspects of Information Technology. 2023. Vol. 6, No. 4. P. 345–361. DOI: 10.15276/aait.06.2023.23. URL: <https://aait.od.ua/index.php/journal/article/view/183/171> (дата звернення: 22.05.2025).
16. Brookner E. Tracking and Kalman Filtering Made Easy. New York: Wiley, 1998. 504 p. URL: https://pzs.dstu.dp.ua/DataMining/kalman/bibl/Brookner_Kalman.pdf (дата звернення: 22.05.2025).
17. NumPy: офіційна документація [Електронний ресурс] / NumPy Developers. 2025. URL: <https://numpy.org/doc/stable/> (дата звернення: 04.06.2025).
18. SciPy: офіційна документація [Електронний ресурс] / SciPy Community. 2025. URL: <https://docs.scipy.org/doc/scipy/> (дата звернення: 04.06.2025).

19. Robot Operating System (ROS): офіційна документація [Електронний ресурс] / Open Robotics. 2025. URL: <https://docs.ros.org/> (дата звернення: 04.06.2025).
20. Gazebo: robot simulation made easy [Електронний ресурс] / Open Robotics. 2025. URL: <http://gazebo.org/> (дата звернення: 04.06.2025).
21. Matplotlib: офіційна документація [Електронний ресурс] / Matplotlib Development Team. 2025. URL: <https://matplotlib.org/stable/contents.html> (дата звернення: 04.06.2025).
22. Eigen: C++ template library for linear algebra [Електронний ресурс] / Eigen Developers. 2025. URL: <http://eigen.tuxfamily.org/> (дата звернення: 04.06.2025).

ДОДАТКИ

Додаток А
(обов'язковий)

Вінницький національний технічний університет
Кафедра автоматизації та інтелектуальних інформаційних технологій

ЗАТВЕРДЖУЮ
завідувача кафедри АІТ
д.т.н., професор Олег БІСІКАЛО

(підпис)
« 23 » _березня_ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на бакалаврську кваліфікаційну роботу
«Дослідження розширеного фільтра Калмана для системи навігації
наземного робота»
08-31.БКР.009.00.000 ТЗ

Керівник бакалаврської кваліфікаційної роботи
к.т.н., доц. Роман МАСЛІЙ

(підпис)
« 23 » _травня__ 2025 р.

Розробив студент гр. АКІТР-23мс
_____Денис КОЛОМІЄЦЬ

(підпис)
« 23 » _травня__ 2025 р.

1 Підстава для проведення робіт

Підставою для виконання бакалаврської кваліфікаційної роботи на тему: «Дослідження розширеного фільтра Калмана для системи навігації наземного робота» є наказ № 97 від 20.03.2025 р.

Термін виконання робіт:

початок 03.09.2024 р.

кінець 10.06.2025 р.

2 Мета та вихідні дані для проведення робіт

Метою бакалаврської кваліфікаційної роботи є дослідження є розробка та аналіз системи навігації наземного робота на основі розширеного фільтра Калмана з інтеграцією даних сенсорів IMU, GPS і одометрії для підвищення точності визначення положення та орієнтації.

Вихідними даними для проведення робіт є індивідуальне завдання на бакалаврську кваліфікаційну роботу від 20.03.2025 р.

3 Етапи виконання робіт

Виконавцем всіх перерахованих в даному розділі етапів є: студент групи АКІТР-23мс Коломієць Денис Володимирович факультету інтелектуальних інформаційних технологій та автоматизації Вінницького національного технічного університету, а замовником є кафедра автоматизації та інтелектуальних інформаційних технологій.

№ Етапу	Зміст етапу	Строки виконання
E1	Аналіз предметної області та визначення основних етапів розробки	07.03 – 19.03
E2	Розробка структури системи навігації наземного робота	20.04 – 26.04
E3	Розробка алгоритму розширеного фільтра Калмана	27.04 – 09.05
E4	Розробка алгоритмічного та програмного забезпечення	10.05 – 01.06
E5	Аналіз результатів роботи та підготовка роботи до захисту	02.06 – 20.06

4 Призначення і галузь застосування

Система навігації наземного робота на основі розширеного фільтра Калмана (EKF) призначена для оцінки стану робота (положення, орієнтації та швидкості) шляхом інтеграції даних від сенсорів IMU, GPS і одометрії в реальному часі. Система забезпечує точне визначення траєкторії руху та орієнтації робота в двовимірному просторі, компенсуючи шуми та похибки сенсорів. Вона застосовується в автономних роботизованих платформах для виконання навігаційних задач у таких галузях, як логістика (автоматизовані складські роботи), сільське господарство (автономні трактори), пошуково-рятувальні операції, а також промислові процеси, де потрібна висока точність позиціонування та стабільність руху.

5 Технічні дані

5.1 Номінальна частота оновлення сенсорних даних:

IMU: 100–200 Гц;

GPS: 1–10 Гц;

Одометрія: 50–100 Гц.

5.2 Точність оцінки позиції:

GPS: 2–5 м;

Після застосування EKF: 0.1–0.5 м.

5.3 Обчислювальна платформа: вбудований мікроконтролер або одноплатний комп'ютер (наприклад, Raspberry Pi 4, NVIDIA Jetson Nano).

5.4 Операційна система: Linux (сумісна з ROS).

5.5 Програмне забезпечення: Python 3.8+ з бібліотеками NumPy, SciPy, ROS (Robot Operating System).

5.6 Коваріація шумів:

Шум процесу (Q): діагональна матриця з відхиленнями 0.1 м (x, y), 0.01 рад (θ);

Шум вимірювань (R): діагональна матриця з відхиленнями 2.5 м (GPS), 0.05 рад/с (IMU).

5.7 Робоче середовище: структуроване або частково неструктуроване (внутрішні приміщення, відкрита місцевість).

5.8 Ступінь захисту апаратного забезпечення: IP54 (для зовнішнього використання, залежить від платформи).

6 Джерела розробки

6.1 Crassidis J. L., Junkins J. L. Optimal Estimation of Dynamic Systems: Chapter 5 – Nonlinear Estimation. 2nd ed. Boca Raton: CRC Press, 2011. P. 253–324. DOI: 10.1201/b11175. URL: https://api.pageplace.de/preview/DT0400.9781439839867_A24454131/preview-9781439839867_A24454131.pdf

6.2 Julier S. J., Uhlmann J. K. A New Extension of the Kalman Filter to Nonlinear Systems. Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls. Orlando: SPIE, 1997. P. 182–193. DOI: 10.1117/12.280797. URL: https://www.cs.unc.edu/~welch/kalman/media/pdf/Julier1997_SPIE_KF.pdf

6.3 Форкун І. В., Медзатий Д. М., Макаришкін Д. А., Червонецький О. В. Система автоматичного керування безпілотним літальним апаратом на основі фільтра Калмана. Вісник Хмельницького національного університету. Технічні науки. 2020. № 6. С. 39–43. URL: <https://journals.khnu.km.ua/vestnik/wp-content/uploads/2021/03/11-2.pdf>

6.4 Методичні вказівки до виконання бакалаврських кваліфікаційних робіт (проектів) для студентів спеціальностей: 126 «Інформаційні системи та технології», 151 «Автоматизація та комп'ютерно-інтегровані технології» [Електронний ресурс] / Уклад. К. В. Овчинников, О. В. Бісікало. – Вінниця : ВНТУ, 2022. – 50 с. URL: <https://iq.vntu.edu.ua/fm/fdb/940/bdr/bdrmetod.pdf>.

Здобувач ст. гр. АКІТР-23мс _____

Денис КОЛОМІЄЦЬ

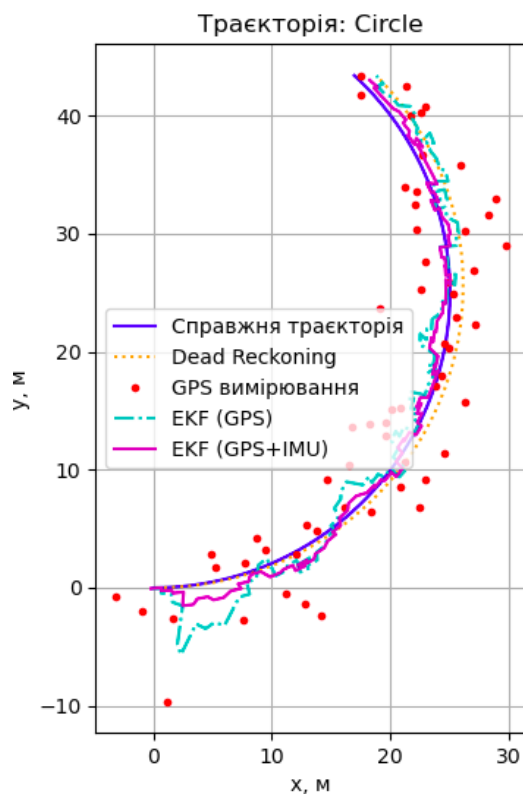
Додаток Б
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

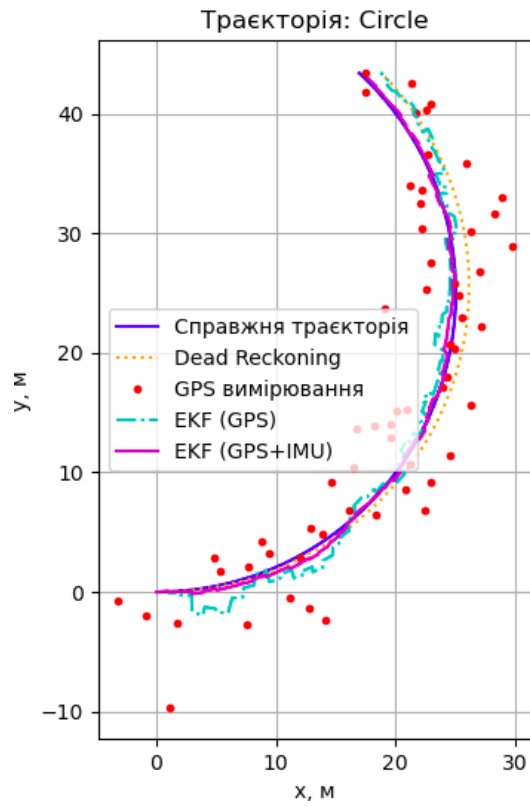
ДОСЛІДЖЕННЯ РОЗШИРЕНОГО ФІЛЬТРА КАЛМАНА ДЛЯ СИСТЕМИ
НАВІГАЦІЇ НАЗЕМНОГО РОБОТА



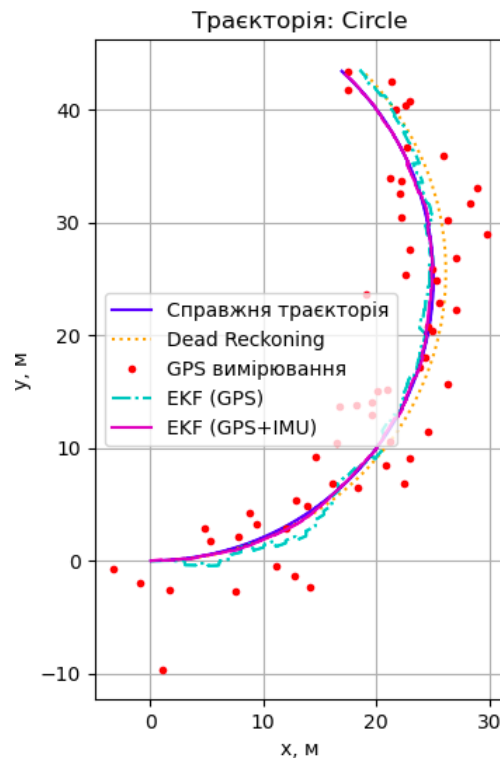
Рисунок Б.1 – Схема програми розширеного фільтру Калмана для системи навігації наземного робота



$$\sigma_{GPSEKF} = 1 \text{ м}$$



$$\sigma_{GPSEKF} = 2.5 \text{ м}$$



$$\sigma_{GPSEKF} = 5 \text{ м}$$

Рисунок Б.2 – Траєкторія наземного робота при русі по колу

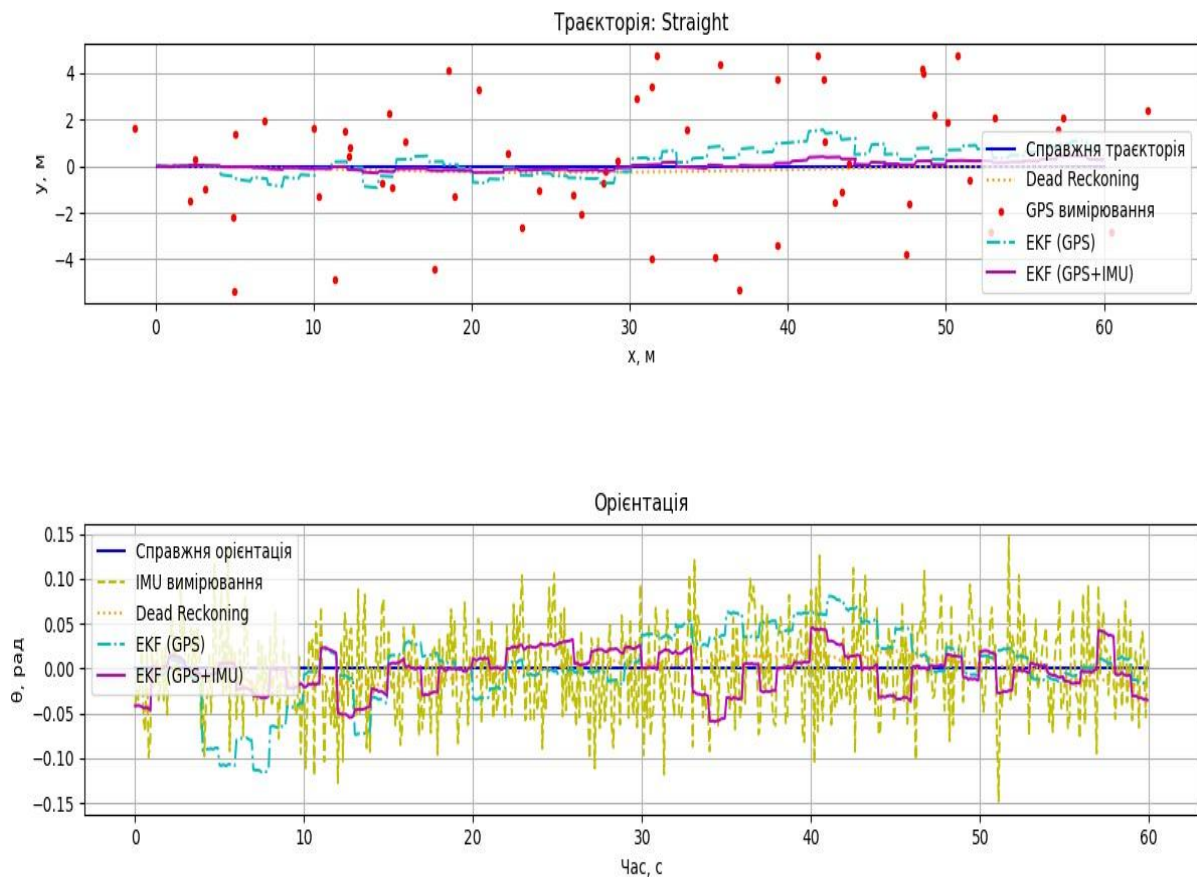


Рисунок Б.3 – Траєкторія та орієнтація наземного робота при прямому русі
(при $\sigma_{GPSEKF} = 5 \text{ м}$)

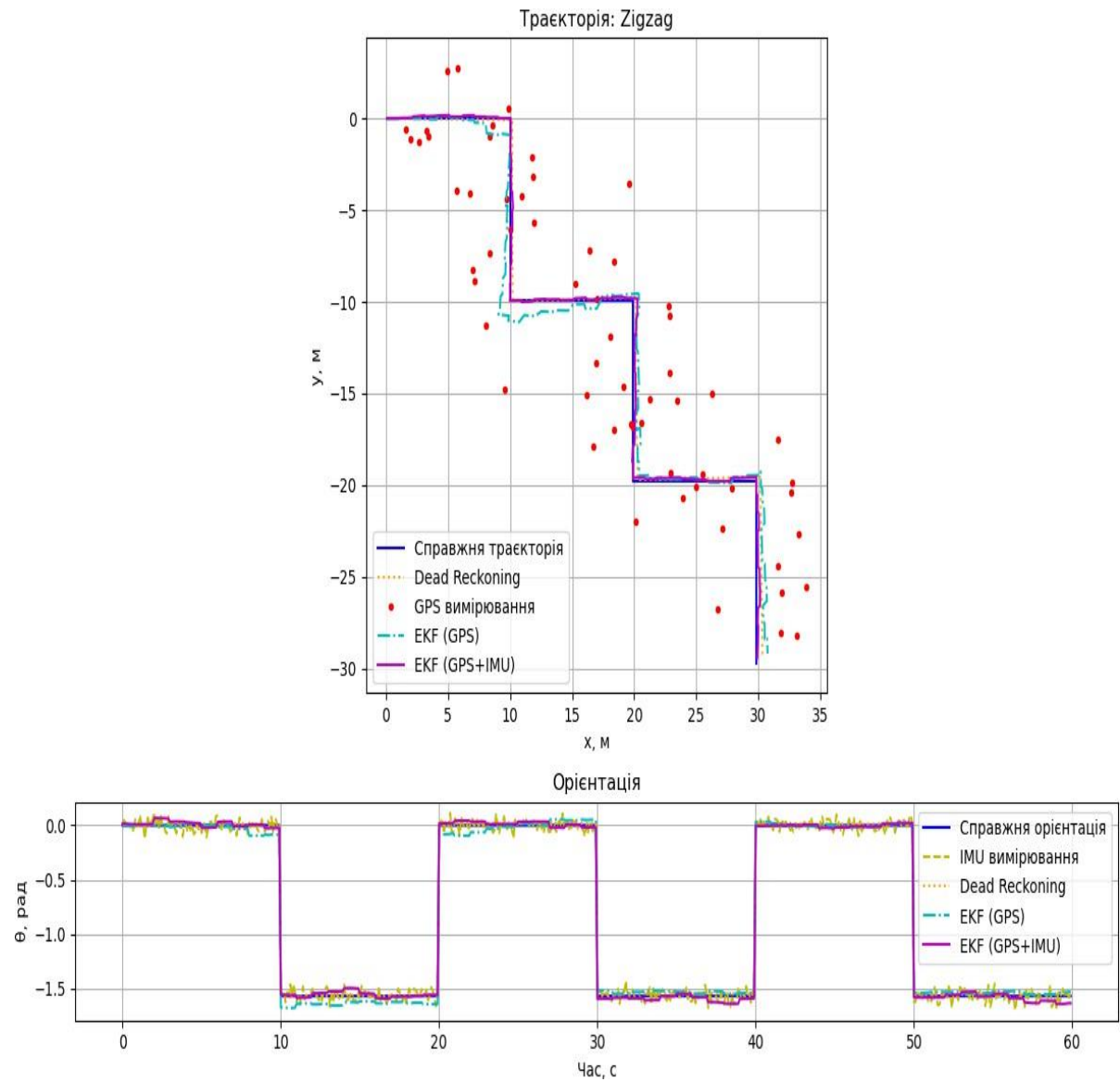


Рисунок Б.4 – Траєкторія та орієнтація наземного робота при зигзагоподібному русі (при $\sigma_{GPSEKF} = 5 \text{ м}$)

Додаток В

(обов'язковий)

Лістинг програмного забезпечення

```
# Скрипт для генерації даних
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Параметри симуляції
dt = 0.1 # Часовий крок, с
T = 60 # Загальний час симуляції, с
N = int(T / dt) # Кількість кроків
gps_freq = 10 # GPS оновлюється кожні 10 кроків (1 Гц)

# Шуми сенсорів (підрозділ 2.3)
sigma_odom = 0.05 # Шум одометрії для v, м/с
sigma_odom_theta = 0.01 # Шум одометрії для omega, рад/с
sigma_gps = 2.5 # 2.5 # Шум GPS, м
sigma_imu = 0.05 # Шум IMU, рад

# Функція для кінематичної моделі
def motion_model(state, u, dt):
    x, y, theta = state
    v, omega = u
    x_new = x + v * dt * np.cos(theta)
    y_new = y + v * dt * np.sin(theta)
    theta_new = theta + omega * dt
    return np.array([x_new, y_new, theta_new])

# Генерація траєкторій
def generate_trajectory(trajectory_type):
    true_states = [np.array([0.0, 0.0, 0.0])] # Початковий стан [x, y, theta]
```

```

odom_data = [] # Дані одометрії [v, omega]
gps_data = [] # Дані GPS [x, y]
imu_data = [] # Дані IMU [theta]
controls = [] # Справжні керуючі сигнали
times = [] # Часові мітки

for i in range(N):
    t = i * dt
    if trajectory_type == "circle":
        # Рух по колу: радіус 25 м, v = 1 м/с, omega = v/R
        v = 1.0
        R = 25.0
        omega = v / R
        u = np.array([v, omega])
    elif trajectory_type == "straight":
        # Прямолінійний рух: v = 1 м/с, omega = 0
        v = 1.0
        omega = 0.0
        u = np.array([v, omega])
    elif trajectory_type == "zigzag":
        # Зигзагоподібний рух: v = 1 м/с, зупинка та поворот на  $\pm 90^\circ$  кожні 10 с
        segment = int(t // 10)
        if i % int(10 / dt) == 0 and i > 0: # Кожні 10 с
            # Зупинка та швидкий поворот на  $\pm 90^\circ$  за 0.1 с
            v = 0.0
            omega = (np.pi / 2) / dt if segment % 2 == 0 else (-np.pi / 2) / dt
        else:
            v = 1.0
            omega = 0.0
        u = np.array([v, omega])

    u_noisy = u + np.random.normal(0, [sigma_odom, sigma_odom_theta])
    true_state = motion_model(true_states[-1], u, dt)

```

```

true_states.append(true_state)
odom_data.append(u_noisy)
controls.append(u)
times.append(t)

# GPS вимірювання
if i % gps_freq == 0:
    gps_meas = true_state[:2] + np.random.normal(0, sigma_gps, 2)
    gps_data.append(gps_meas)
else:
    gps_data.append([np.nan, np.nan])

# IMU вимірювання
imu_meas = true_state[2] + np.random.normal(0, sigma_imu)
imu_data.append(imu_meas)

    return np.array(true_states[1:]), np.array(odom_data), gps_data, np.array(imu_data),
np.array(controls), np.array(times)

# Генерація одометричної траєкторії
def generate_odom_trajectory(initial_state, odom_data, dt):
    odom_states = [initial_state]
    for u in odom_data:
        odom_state = motion_model(odom_states[-1], u, dt)
        odom_states.append(odom_state)
    return np.array(odom_states[1:])

# Збереження даних у один CSV
def save_to_csv(trajectory_type, true_states, odom_data, gps_data, imu_data, times):
    data = {
        'time': times,
        'x_true': true_states[:, 0],
        'y_true': true_states[:, 1],
        'theta_true': true_states[:, 2],
        'v_odom': odom_data[:, 0],
        'omega_odom': odom_data[:, 1],
    }

```

```

    'x_gps': [g[0] for g in gps_data],
    'y_gps': [g[1] for g in gps_data],
    'theta_imu': imu_data
}
df = pd.DataFrame(data)
df.to_csv(f'{trajectory_type}_data.csv', index=False)

# Візуалізація
def plot_trajectory(trajectory_type, true_states, odom_states, gps_data, imu_data, times):
    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))

    # Траєкторія (x, y)
    ax1.plot(true_states[:, 0], true_states[:, 1], 'b-', label='Справжня траєкторія')
    ax1.plot(odom_states[:, 0], odom_states[:, 1], 'g--', label='Одометрія')
    gps_x = [g[0] for g in gps_data if not np.isnan(g[0])]
    gps_y = [g[1] for g in gps_data if not np.isnan(g[1])]
    ax1.plot(gps_x, gps_y, 'r.', label='GPS вимірювання')
    ax1.set_xlabel('x, м')
    ax1.set_ylabel('y, м')
    ax1.set_title(f'Траєкторія: {trajectory_type.capitalize()}')
    ax1.legend()
    ax1.grid(True)
    ax1.axis('equal')

    # Орієнтація (theta)
    ax2.plot(times, true_states[:, 2], 'b-', label='Справжня орієнтація')
    ax2.plot(times, imu_data, 'm--', label='IMU вимірювання')
    ax2.set_xlabel('Час, с')
    ax2.set_ylabel('θ, рад')
    ax2.set_title('Орієнтація')
    ax2.legend()
    ax2.grid(True)

```

```

plt.tight_layout()
plt.show()

# Генерація, збереження та візуалізація
trajectories = ["circle", "straight", "zigzag"]
for traj in trajectories:
    true_states, odom_data, gps_data, imu_data, controls, times = generate_trajectory(traj)
    odom_states = generate_odom_trajectory(true_states[0], odom_data, dt)
    save_to_csv(traj, true_states, odom_data, gps_data, imu_data, times)
    plot_trajectory(traj, true_states, odom_states, gps_data, imu_data, times)

# Скрипт для дослідження стабільності
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec

# Параметри симуляції
dt = 0.1 # Часовий крок, с
gps_freq = 10 # GPS оновлюється кожні 10 кроків (1 Гц)

# Шуми сенсорів (підрозділ 2.3)
sigma_odom = 0.05 # Шум одометрії для v, м/с
sigma_odom_theta = 0.01 # Шум одометрії для omega, рад/с
sigma_gps = 1 # Шум GPS, м
sigma_imu = 0.05 # Шум IMU, рад

# Коваріаційні матриці
Q = np.diag([sigma_odom**2, sigma_odom**2, sigma_odom_theta**2]) # Шум процесу
R_gps = np.diag([sigma_gps**2, sigma_gps**2]) # Шум вимірювань GPS
R_gps_imu = np.diag([sigma_gps**2, sigma_gps**2, sigma_imu**2]) # Шум GPS+IMU

# Модель руху

```

```

def motion_model(x, u, dt):
    v, omega = u
    x_new = x[0] + v * dt * np.cos(x[2])
    y_new = x[1] + v * dt * np.sin(x[2])
    theta_new = x[2] + omega * dt
    return np.array([x_new, y_new, theta_new])

# Якобіан моделі руху
def motion_jacobian(x, u, dt):
    v, _ = u
    return np.array([
        [1, 0, -v * dt * np.sin(x[2])],
        [0, 1, v * dt * np.cos(x[2])],
        [0, 0, 1]
    ])

# Модель вимірювань
def measurement_model(x, mode):
    if mode == "gps":
        return x[:2] # Тільки x, y
    return x # x, y, theta для GPS+IMU

# Якобіан моделі вимірювань
def measurement_jacobian(mode):
    if mode == "gps":
        return np.array([[1, 0, 0], [0, 1, 0]]) # Ігноруємо theta
    return np.eye(3) # Повна матриця для GPS+IMU

# Реалізація EKF
def ekf(x, P, u, z, R, mode):
    # Передбачення
    x_pred = motion_model(x, u, dt)
    F = motion_jacobian(x, u, dt)

```

$$P_pred = F @ P @ F.T + Q$$

Корекція (якщо є вимірювання)

if z is not None:

 H = measurement_jacobian(mode)

 z_pred = measurement_model(x_pred, mode)

 y = z - z_pred

 S = H @ P_pred @ H.T + R

 K = P_pred @ H.T @ np.linalg.inv(S)

 x_new = x_pred + K @ y

 P_new = (np.eye(3) - K @ H) @ P_pred

else:

 x_new, P_new = x_pred, P_pred

return x_new, P_new

Обчислення Dead Reckoning

def dead_reckoning(initial_state, odom_data, dt):

 states = [initial_state]

 for u in odom_data:

 state = motion_model(states[-1], u, dt)

 states.append(state)

return np.array(states[1:])

Обчислення метрик RMSE і MAE

def compute_metrics(true_states, est_states):

 errors = true_states - est_states

 rmse = np.sqrt(np.mean(np.sum(errors**2, axis=1)))

 mae = np.mean(np.abs(errors))

return rmse, mae

Зчитування даних із CSV

def load_data(trajjectory):

```

df = pd.read_csv(f'{trajectory}_data.csv')
return {
    'time': df['time'].values,
    'true_states': df[['x_true', 'y_true', 'theta_true']].values,
    'odom': df[['v_odom', 'omega_odom']].values,
    'gps': df[['x_gps', 'y_gps']].values,
    'imu': df['theta_imu'].values
}

```

Застосування EKF і збереження результатів

```

def run_ekf(data, trajectory, mode):
    x = np.array([0.0, 0.0, 0.0]) # Початковий стан
    P = np.eye(3) * 0.1 # Початкова коваріація
    est_states = []
    R = R_gps if mode == "gps" else R_gps_imu

    for i in range(len(data['time'])):
        u = data['odom'][i]
        # Вимірювання
        if i % gps_freq == 0 and not np.isnan(data['gps'][i, 0]):
            if mode == "gps":
                z = data['gps'][i]
            else:
                z = np.array([data['gps'][i, 0], data['gps'][i, 1], data['imu'][i]])
            else:
                z = None

        # Оновлення EKF
        x, P = ekf(x, P, u, z, R, mode)
        est_states.append(x)

    est_states = np.array(est_states)
    # Збереження результатів
    df_est = pd.DataFrame({

```

```

        'time': data['time'],
        'x_est': est_states[:, 0],
        'y_est': est_states[:, 1],
        'theta_est': est_states[:, 2]
    })
    df_est.to_csv(f'{trajectory}_ekf_{mode}.csv', index=False)
    return est_states

# Візуалізація
def plot_results(trajectory, data, est_states_gps, est_states_gps_imu, dead_states):
    true_states = data['true_states']

    gps = data['gps']
    imu = data['imu']
    times = data['time']

    # Налаштування сітки для підграфіків із різними розмірами
    fig = plt.figure(figsize=(12, 8))

    gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1]) # Траєкторія більша, орієнтація
    менша

    # Підграфік траєкторій
    ax1 = fig.add_subplot(gs[0])
    ax1.plot(true_states[:, 0], true_states[:, 1], 'b-', label='Справжня траєкторія')
    ax1.plot(dead_states[:, 0], dead_states[:, 1], 'orange', linestyle=':', label='Dead Reckoning')
    gps_x = [g[0] for g in gps if not np.isnan(g[0])]
    gps_y = [g[1] for g in gps if not np.isnan(g[1])]
    ax1.plot(gps_x, gps_y, 'r.', label='GPS вимірювання')
    ax1.plot(est_states_gps[:, 0], est_states_gps[:, 1], 'c-.', label='EKF (GPS)')
    ax1.plot(est_states_gps_imu[:, 0], est_states_gps_imu[:, 1], 'm-', label='EKF (GPS+IMU)')
    ax1.set_xlabel('x, м')
    ax1.set_ylabel('y, м')
    ax1.set_title(f'Траєкторія: {trajectory.capitalize()}')
    ax1.legend()
    ax1.grid(True)

```

```

ax1.set_aspect('equal', adjustable='box') # Рівний масштаб для траєкторії

# Підграфік орієнтації
ax2 = fig.add_subplot(gs[1])
ax2.plot(times, true_states[:, 2], 'b-', label='Справжня орієнтація')
ax2.plot(times, imu, 'y--', label='IMU вимірювання')
ax2.plot(times, dead_states[:, 2], 'orange', linestyle=':', label='Dead Reckoning')
ax2.plot(times, est_states_gps[:, 2], 'c-.', label='EKF (GPS)')
ax2.plot(times, est_states_gps_imu[:, 2], 'm-', label='EKF (GPS+IMU)')
ax2.set_xlabel('Час, с')
ax2.set_ylabel('θ, рад')
ax2.set_title('Орієнтація')
ax2.legend()
ax2.grid(True)
ax2.set_aspect('auto') # Автоматичний масштаб для орієнтації

plt.tight_layout()
plt.show()

# Обчислення метрик
rmse_dead, mae_dead = compute_metrics(true_states, dead_states)
rmse_gps, mae_gps = compute_metrics(true_states, est_states_gps)
rmse_gps_imu, mae_gps_imu = compute_metrics(true_states, est_states_gps_imu)

print(f'{trajectory.capitalize()} - Dead Reckoning: RMSE={rmse_dead:.3f},
MAE={mae_dead:.3f}')

print(f'{trajectory.capitalize()} - GPS: RMSE={rmse_gps:.3f}, MAE={mae_gps:.3f}')

print(f'{trajectory.capitalize()} - GPS+IMU: RMSE={rmse_gps_imu:.3f},
MAE={mae_gps_imu:.3f}')

# Основна програма
trajectories = ["circle", "straight", "zigzag"]
for traj in trajectories:
    data = load_data(traj)

# Обчислення Dead Reckoning

```

```
dead_states = dead_reckoning(data['true_states'][0], data['odom'], dt)
# Застосування EKF
est_states_gps = run_ekf(data, traj, "gps")
est_states_gps_imu = run_ekf(data, traj, "gps_imu")
# Візуалізація та метрики
plot_results(traj, data, est_states_gps, est_states_gps_imu, dead_states)
```

Додаток Г
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Дослідження розширеного фільтра Калмана для системи навігації наземного робота

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра автоматизації та інтелектуальних інформаційних технологій
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 2.14 %

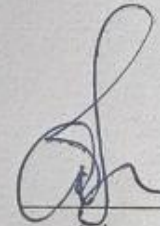
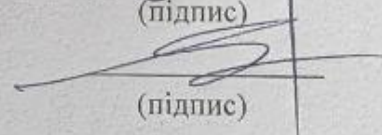
Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

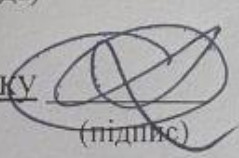
- ☐ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Олег БІСІКАЛО, зав. кафедри АІТ
(прізвище, ініціали, посада)

Любов ЯНЧУК, секретар кафедри АІТ
(прізвище, ініціали, посада)

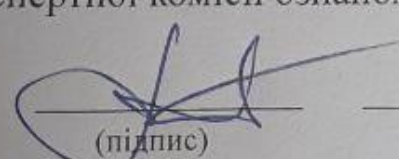

(підпис)

(підпис)

Особа, відповідальна за перевірку 
(підпис)

Костянтин ОВЧИННИКОВ
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник


(підпис)

Роман МАСЛІЙ
(прізвище, ініціали, посада)

Здобувач


(підпис)

Денис КОЛОМІЄЦЬ
(прізвище, ініціали)