Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Математичні моделі та методи визначення
траєкторії руху об'єкта за відеозображенням
(тема)

Виконав:
студент _2_ курсу, групи _ПМм-19-1
Толкачова М.Г.
(прізвище, ініціали)
Спеціальність
113 Прикладна математика
(код і повна назва спеціальності)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма
Прикладна математика
(повна назва освітньої програми)
Керівник доц. Єсілевський В.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ _____ Тевяшев А.Д.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _інформаційно-аналітичних технологій та менеджменту
Кафедра _прикладної математики
Рівень вищої освіти _другий (магістерський)
Спеціальність _113 Прикладна математика
(код і повна назва)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _Прикладна математика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____
(підпис)
"____" _____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _Толкачовій Марині Геннадіївні
(прізвище, ім'я, по батькові)

1. Тема роботи _Математичні моделі та методи визначення траєкторії руху об'єкта за відеозображенням
затверджена наказом по університету від 23 жовтня 2020 р. № 1422 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2020 р.

3. Вихідні дані до роботи _змодельовані дані початкової траєкторії об'єкта

4. Перелік питань, що потрібно опрацювати в роботі
1. Системний аналіз проблеми визначення траєкторії руху об'єкта по відеозображенню
2. Вибір і обґрунтування методу розв'язання
3. Програмна реалізація
4. Результати обчислювального експерименту
5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Актуальність теми роботи
2. Постановка задачі
3. Системний аналіз проблеми
4. Метод чисельного аналізу
5. Результати обчислювального експерименту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	вересень 2020 р.	виконано
2	Вибір та обґрунтування методу	жовтень – листопад 2020 р.	виконано
3	Розробка алгоритму і програми	листопад – грудень 2020 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	листопад – грудень 2020 р.	виконано
5	Робота над текстом пояснювальної записки	грудень 2020 р.	виконано
6	Представлення роботи на рецензію в ЕК	грудень 2020 р.	виконано

Дата видачі завдання _1_ вересня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Єсілевський В.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 76 с., 5 табл., 25 рис., 1 дод., 10 джерел.

СИГМА-ТОЧКОВИЙ ФІЛЬТР КАЛМАНА, ПРОГНОЗУВАННЯ ШЛЯХУ, ТРАЄКТОРІЯ, ВІДЕОЗОБРАЖЕННЯ, НЕЛІНІЙНА СИСТЕМА.

Об’єкт дослідження – процес руху об’єкта у відеоряді.

Мета роботи – розробити програмний продукт для прогнозування траєкторії руху об’єкта по відеозображенню.

Методи дослідження – сигма-точковий фільтр Калмана.

В рамках даної атестаційній роботі було досліджено методи прогнозування шляху та методи вилучення функцій з відео. Розроблено математичну модель для визначення траєкторії руху об’єкта, який рухається нелінійно, за допомогою сигма-точкового фільтра Калмана. Для цього було розглянути та проаналізовано існуючі засоби та рішення в області відстеження об’єктів. Та створено власний програмний продукт.

ABSTRACT

Introductory note: 76 pages, 5 tables, 25 figures, 1 appendixes, 10 sources.

UNSCENTED KALMAN FILTER, PATH FORECASTING, TRAJECTORY, VIDEO IMAGE, NONLINEAR SYSTEM.

Object of research – the process of moving the object in the video sequence.

Purpose of work – develop a software product for predicting the trajectory of the object on the video image.

Methods of research – Unscented Kalman Filter.

Within the framework of this attestation work the methods of path prediction and methods of extracting functions from video were investigated. A mathematical model has been developed to determine the trajectory of an object moving nonlinearly using a Kalman sigma point filter. To do this, the existing tools and solutions in the field of object tracking were reviewed and analyzed. And created its own software product.

ЗМІСТ

С.

Вступ 8

1 Системний аналіз проблеми визначення траєкторії руху об’єкта по відеозображенню та постановка задач дослідження 9

1.1 Системний аналіз проблеми визначення траєкторії руху об’єкта по відеозображенню 9

1.1.1 Вербальна модель системи 9

1.1.2 Морфологічний опис системи 9

1.1.3 Функціональна модель системи 10

1.1.4 Інформаційна модель системи 14

1.2 Аналіз сценаріїв вирішення проблеми визначення траєкторії руху об’єкта по відеозображенню 15

1.2.1 Модель аналізу проблеми 15

1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій 17

1.2.3 Модель вирішення проблеми 23

1.3 Змістовна та формальна постановка задачі 24

1.3.1 Змістовна постановка задачі 24

1.3.2 Формальна постановка задачі 26

1.4 Постановка задач дослідження 27

2 Вибір та обґрунтування методу розв’язання 28

2.1 Дослідження прогнозування шляху 28

2.2 Методи прогнозування шляху 29

2.2.1 Байєсовські моделі 29

2.2.2 Мінімізація енергії 31

2.2.3 Глибоке навчання 32

2.2.4 Зворотнє навчання з підкріпленням 33

2.2.5 Інші підходи 35

2.3 Сигма-точковий Фільтр Калмана 37
2.3.1 Математичні очікування та коваріації нелінійних перетворень 37
2.3.1.1 Математичне очікування нелінійного перетворення 38
2.3.1.2 Коваріація нелінійного перетворення 43
2.3.2 Сигма-точкове перетворення 47
2.3.2.1 Апроксимація математичного очікування 49
2.3.2.2 Апроксимація коваріації 53
2.3.3 Сигма-точковий Фільтр Калмана 56
3 Програмна реалізація 62
3.1 Вибір мови програмування та необхідних бібліотек 62
3.2 Опис програми 63
4 Результати обчислювального експерименту 64
5 Аналіз можливих застосувань 69
Висновки 70
Перелік джерел посилання 71
Додаток А Код програми 72

ВСТУП

У сучасному світі велику роль в зборі інформації відіграють різні системи відеоспостереження. Системами відеоспостереження оснащуються багато підприємств, організації та важливі об'єкти інфраструктури. Такі системи використовуються для охорони або моніторингу певних ситуацій.

Серед усього розмаїття завдань, для вирішення яких створюються системи відеоспостереження, виділяється завдання відстеження рухомих об'єктів у відеопотоці. Система відстеження рухомих об'єктів необхідна сьогодні в багатьох областях. Інформація, отримана від такої системи відстеження, може бути використана для аналізу характеристик об'єкта, таких як швидкість, розмір і шлях об'єкта, або для прогнозування подальшої траєкторії руху об'єкта. Ці дані, наприклад, можуть бути використані для моніторингу дорожньої ситуації, виявлення аварій або інших подій.

В процесі обробки даних виникають проблеми, пов'язані з точністю вимірювального інструмента. Для зменшення цієї похибки використовуються алгоритми фільтрації. Фільтрація даних являє собою об'єднання деякої інформації про об'єкт, отриманої відповідно до моделі руху і вимірюваних даних.

У даній роботі використовується сигма-точковий фільтр Калмана. Фільтр Калмана – один з найпопулярніших інструментів фільтрації даних, який використовується в багатьох галузях науки (наприклад, для вирішення різних завдань в економічній сфері та інженерних дослідженнях). Він використовується для ефективної оцінки стану системи на основі серії зашумлених вимірювань або вимірювань з декількох датчиків. Перевага сигма-точкового фільтра Калмана в тому, що він може значно покращити результати оцінки (у порівнянні з розширеним фільтром Калмана) для нелінійних систем. А саме таку систему ми і будемо розглядати в атестаційній роботі.

1 СИСТЕМНИЙ АНАЛІЗ ПРОБЛЕМИ ВИЗНАЧЕННЯ ТРАЄКТОРІЇ РУХУ ОБ'ЄКТА ПО ВІДЕОЗОБРАЖЕННЮ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз проблеми визначення траєкторії об'єкта по відеозображенню

1.1.1 Вербальна модель системи

Об'єкт аналізу – «Визначення траєкторії руху об'єкта по відеозображенню».

Предмет аналізу – «Сигма-точковий фільтр Калмана».

Точка зору: дослідник.

Ціль: передбачити подальшу ймовірну траєкторію руху об'єкта.

На виході системи ми повинні отримати графік траєкторії руху конкретного об'єкта. На вхід системи ми подаємо данні у вигляді кадрів із відео, на яких присутній даний об'єкт. Управління системи – алгоритми машинного навчання та мережа Інтернет. Механізми системи – дослідник та програмне забезпечення.

1.1.2 Морфологічний опис системи

Морфологічний опис задачі інтелектуального аналізу почнемо з розгляду поняття зовнішнього середовища.

Зовнішнє середовище – сукупність всіх об'єктів за межами границі системи, зміна властивостей яких мають вплив на систему, а також тих об'єктів, чиї властивості змінюються в результаті поведінки системи.

Для опису функціонування моделі з зовнішнім середовищем на рисунку 1.1 представлено модель типу «чорний ящик».

Модель, що представлена як «чорний ящик», включає моделі границі, модель зовнішнього середовища, входи і виходи.

Виходи системи – це цільовий продукт системи. Входи системи – це частіше всього ресурсний вплив зовнішнього середовища. «Чорний ящик» здійснює перетворення входів в виходи системи.

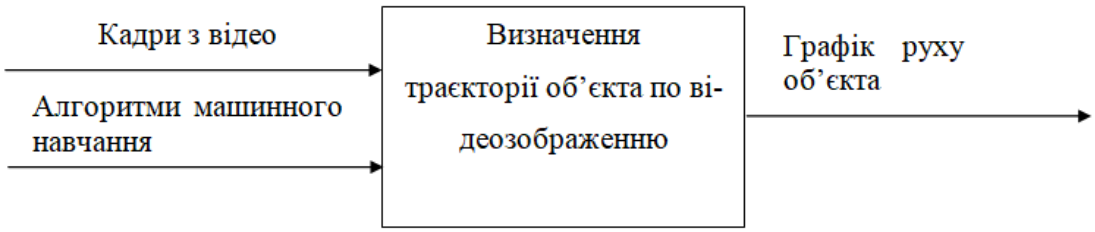

Рисунок 1.1 – Модель системи типу «чорний ящик»

1.1.3 Функціональна модель системи

Для чіткого формулювання логіки та взаємодії процесів використовується мова моделювання бізнес-процесів IDEF0. Ми будуємо процес визначення траєкторії руху об'єкта по відеозображенню.

Процес моделювання будь-якої системи в IDEF0 починається з визначення контексту, тобто найбільш абстрактного рівня опису системи в цілому.
В контекст входить визначення суб'єкта моделювання, цілі і точка зору на модель [1]. Для даної задачі:

– точка зору – дослідник;

– суб'єкт – зображення, вилучені з відео;

– ціль – аналіз кадрів із відео з метою прогнозування траєкторії об'єкта.

Контекстна діаграма IDEF0 зображує функціонування системи в цілому (рисунок 1.2). Основною задачею є визначення траєкторії руху об'єкта, яка виконується дослідником за допомогою ЕОМ. Інформаційним ресурсом є мережа Інтернет, яка застосовуються до вхідних даних. В результаті виконання роботи

очікується отримання графіку ймовірного маршруту об’єкта.

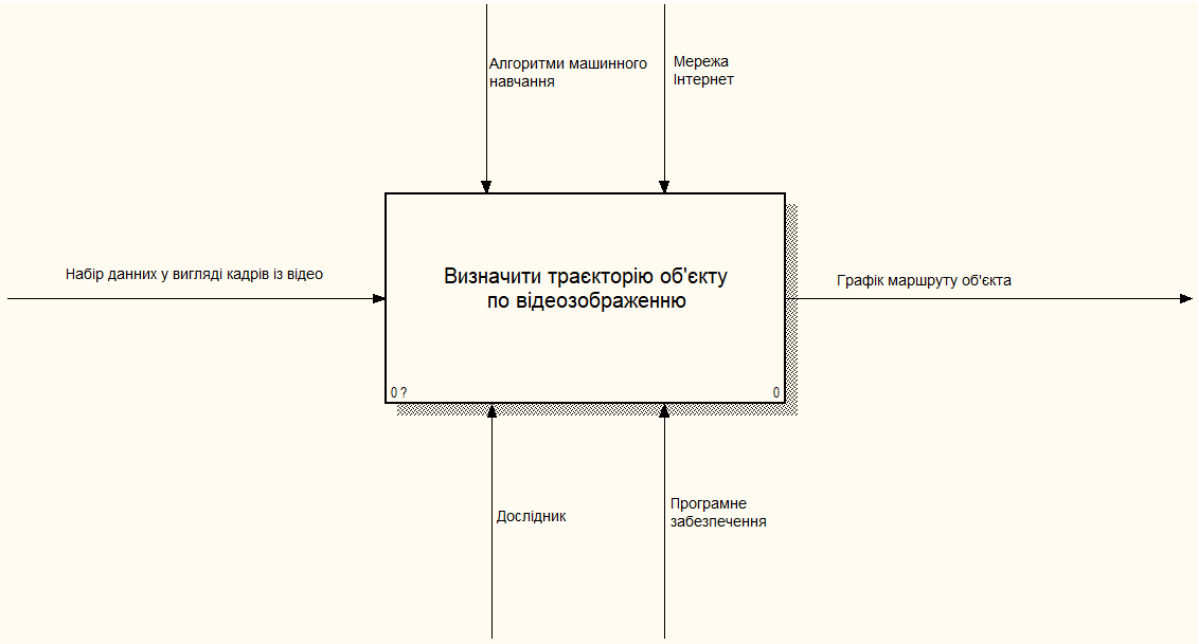

Рисунок 1.2 – Контекстна діаграма IDEF0 (рівень А – 0)

Щоб розглянути функціональну частину більш детально, виконується декомпозиція системи. Декомпозиція роботи «Визначити траєкторію об’єкта по відеозображенню» зображена на рисунку 1.3. Даний процес розділено на три задачі:

– зібрати дані;

– розробити код;

– отримати результат.

Після декомпозиції контекстної діаграми проводиться декомпозиція кожного великого фрагмента системи на більш дрібні і так далі, до досягнення потрібного рівня подробиці опису.

IDEF3 є стандартом документування технологічних процесів, що відбуваються на підприємстві, і надає інструментарій для наочного дослідження і моделювання їх сценаріїв. IDEF3 широко застосовується при розробці інформаційних систем [2]. При цьому використовується інструмент візуального моделювання бізнес-процесів. Система описується як упорядкована послідовність подій з одночасним описом об'єктів, що мають відношення до процесу, що мо-

делюється.

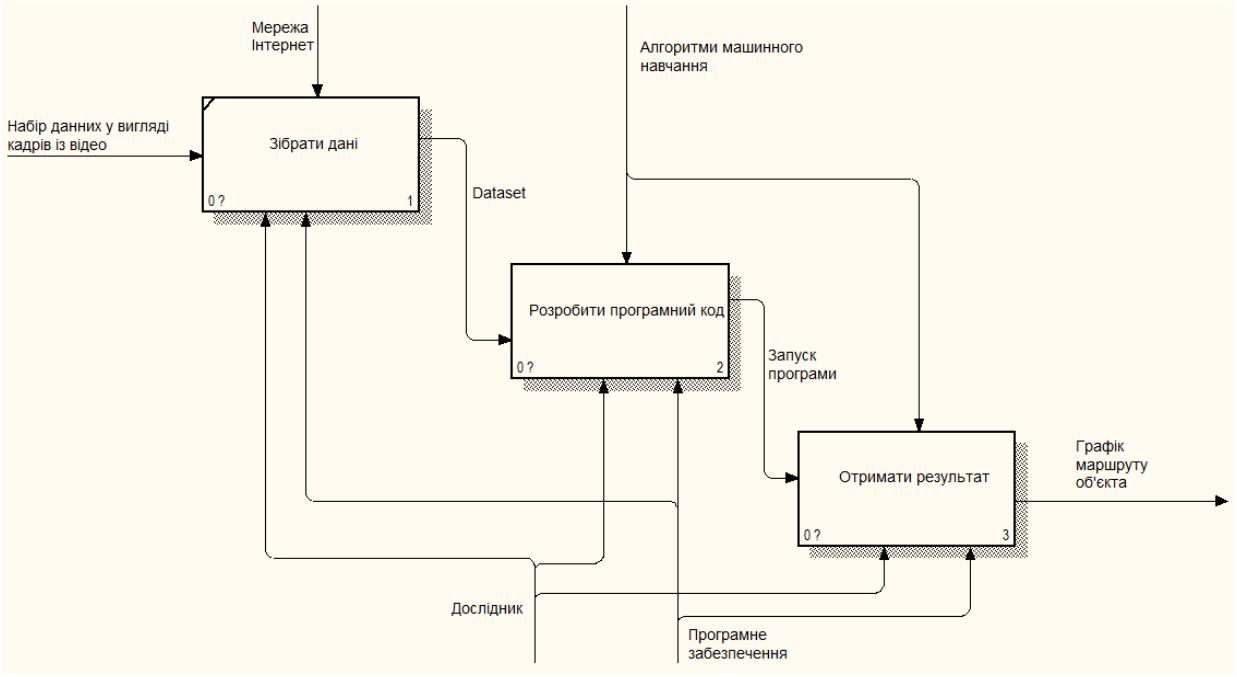

Рисунок 1.3 – Декомпозиція контекстної діаграми (рівень A0)

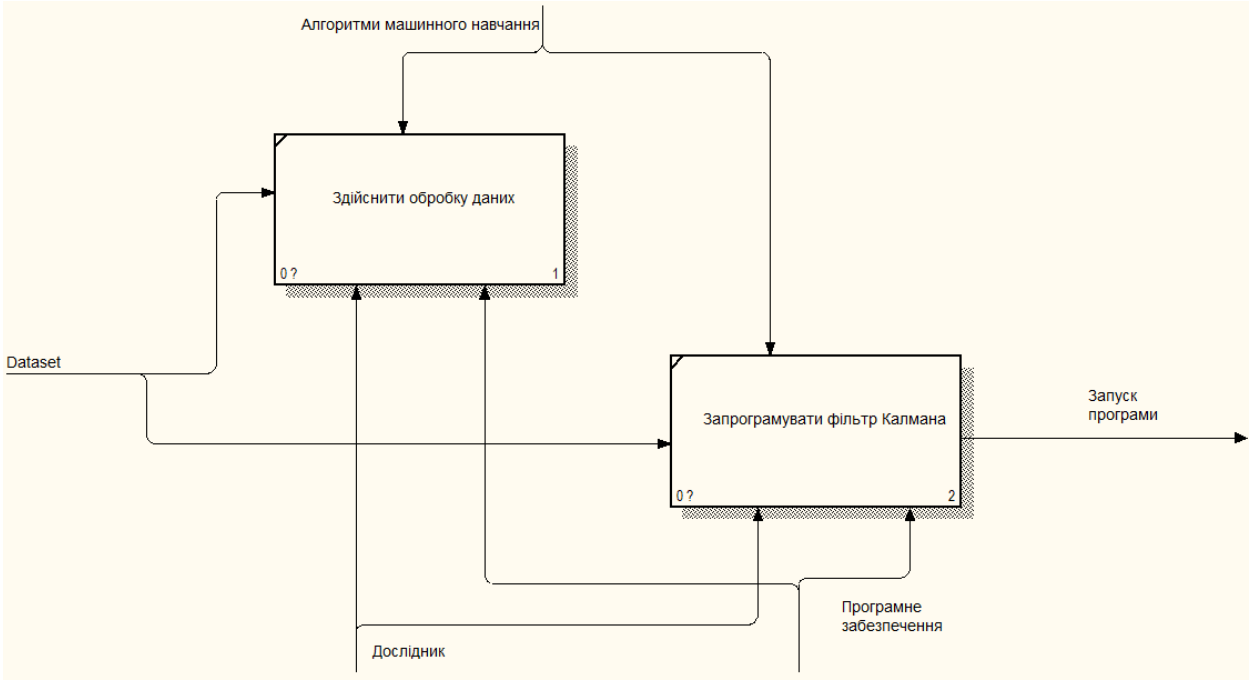

Рисунок 1.4 – Декомпозиція роботи «Розробити програмний код» (рівень A2)

Розглянемо опис роботи у стандарті IDEF3 на рисунках 1.6 – 1.8.

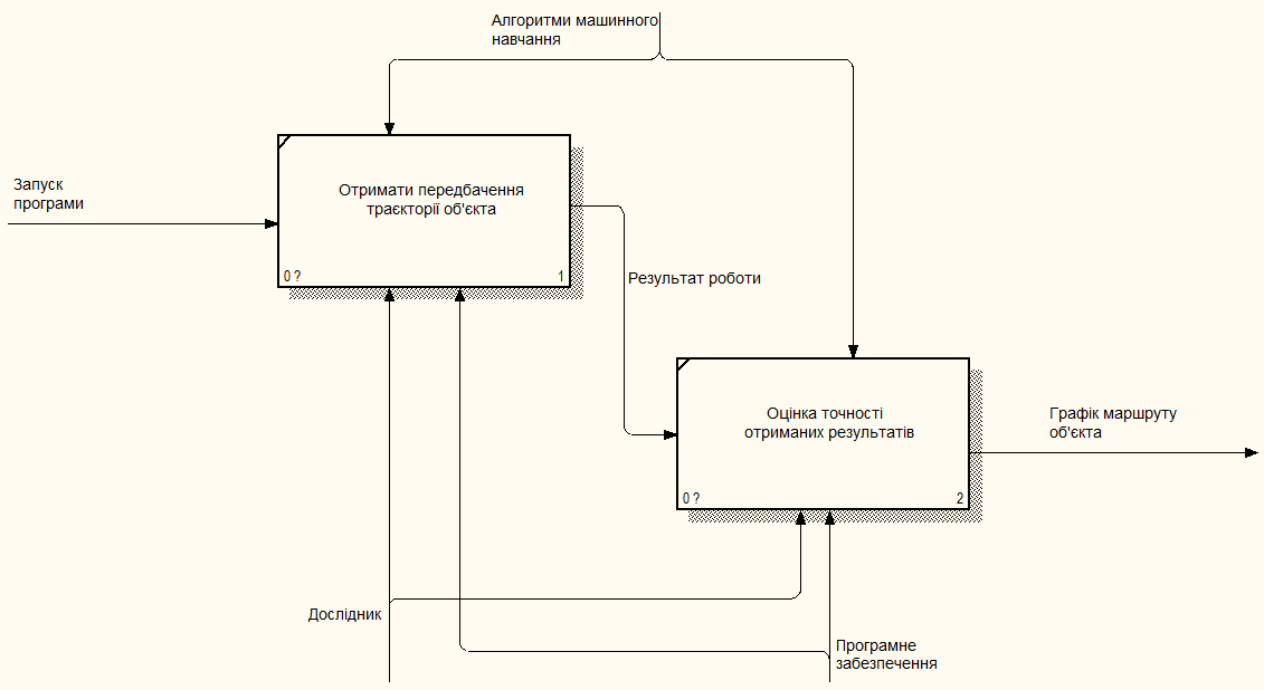

Рисунок 1.5 – Декомпозиція роботи «Отримати результат» (рівень А3)

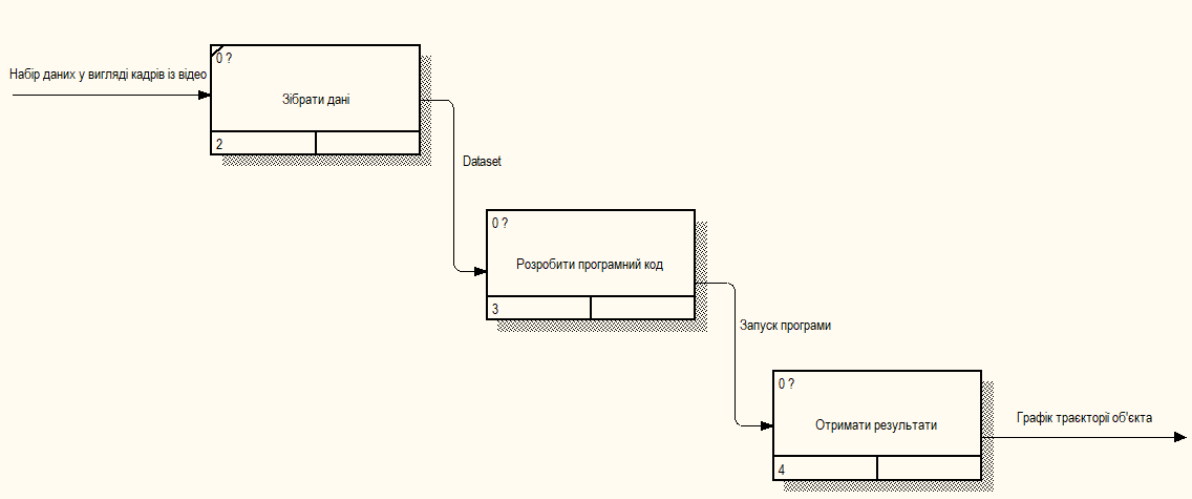

Рисунок 1.6 – Опис роботи «Визначити траєкторію об’єкта по відеозображенню» (рівень А0 у нотації IDEF3)

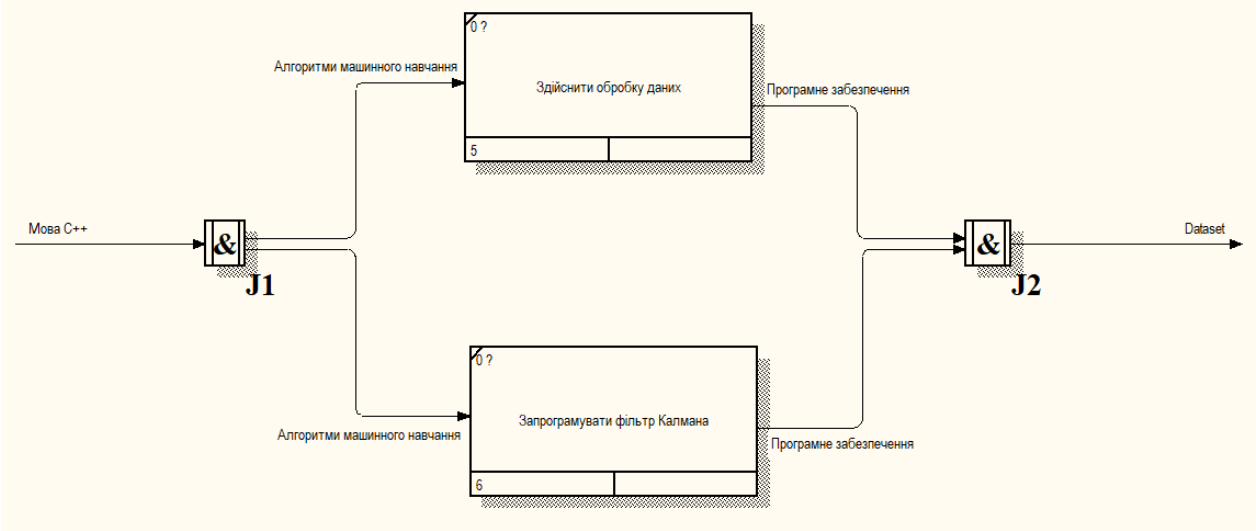

Рисунок 1.7 – Опис роботи «Розробити програмний код»
(рівень А2 в нотації IDEF3)

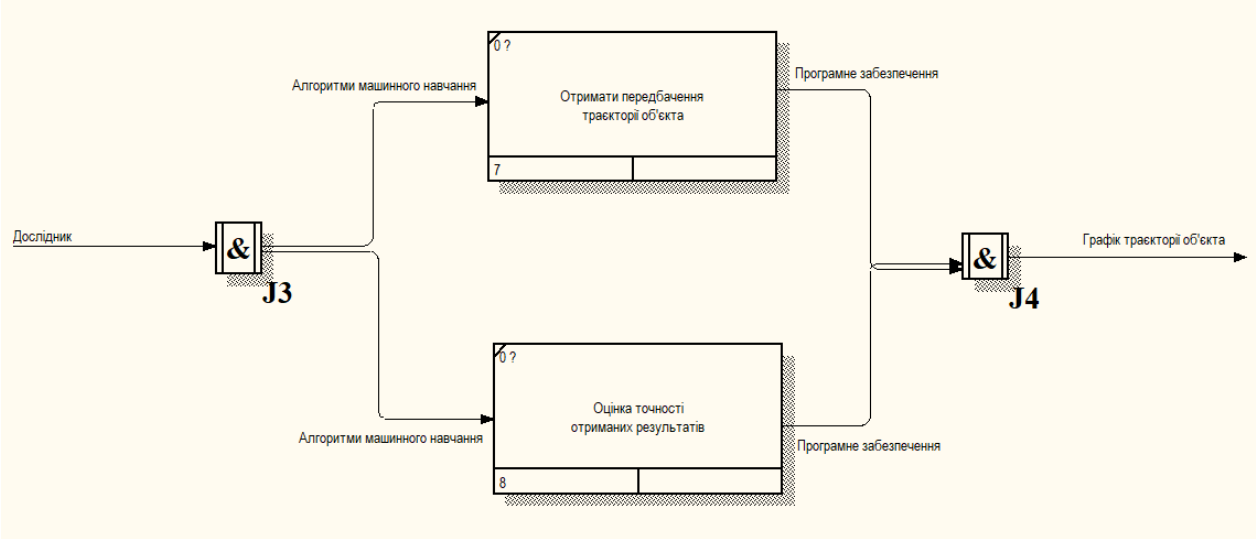

Рисунок 1.8 – Опис роботи «Отримати результат»
(рівень А3 у нотації IDEF3)

1.1.4 Інформаційна модель

Інформаційні моделі відображають різні типи систем об'єктів, в яких реалізуються різні структури взаємодії і взаємозв'язку між елементами системи.

Інструменти методології DFD дозволяють відображати джерела і адресати даних, ідентифікувати процеси і групи даних, що зв'язують в потоки одну

функцію з іншого, і ефективно використовуються для опису процесів при впровадженні процесного підходу до управління організацією, так як дозволяє максимально знизити суб'єктивність опису бізнес процесів [2]. Крім того, нотація DFD дозволяє описувати потоки документів і потоки ресурсів.

Діаграма дерева вузлів показує ієрархію робіт в моделі і дозволяє розглянути всю модель цілком, але не показує взаємозв'язку між роботами.

Діаграма дерева вузлів для всіх вузлів моделі показана на рисунку 1.9.

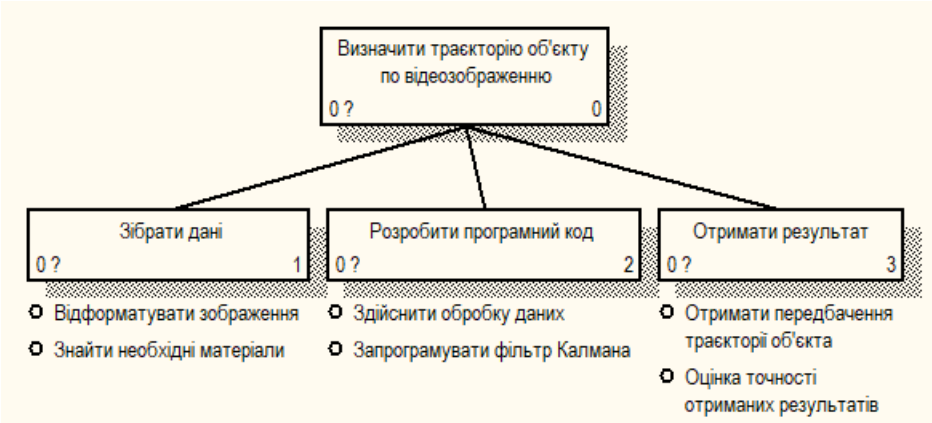

Рисунок 1.9 – Діаграма дерева вузлів

1.2 Аналіз сценаріїв вирішення проблеми визначення траєкторії руху об’єкта по відеозображенню

1.2.1 Модель аналізу проблеми

Математична модель має бути вирішена при застосуванні різних алгоритмів з подальшим їх порівнянням з точки зору точності висновків та особливостей програмної реалізації. В результаті очікується визначення найбільш оптимального алгоритму, який може бути застосований для вирішення даної задачі.

Виділимо критерії, які на наш погляд здійснюють найбільший вплив на очікуваний результат. Такими критеріями є:

– швидкодія (К1);

– універсальність (К2);

– надійність (К3);

– ресурсоємність (К4).

Розглянемо перераховані вище критерії більш детально і визначимо, що саме порівнюватиметься в наступних альтернативах. У свою чергу, в якості альтернативи будуть обрані алгоритми визначення траєкторії.

При порівнянні ресурсоємності альтернатив ми маємо на увазі ресурси обчислювальної машини, які будуть потрібні для успішного завершення алгоритмів без технічних помилок: обсяг оперативної пам'яті, машинний час для виконання обчислень.

Порівнюючи альтернативи на універсальність, нас цікавить можливість використання алгоритмів для досить великих вибірок і вхідні дані, необхідні для алгоритму. Перевагу буде надано тому алгоритму, який потребує менше вхідної інформації.

Надійність алгоритмів: очікуємо, що у всіх випадках ми отримаємо однозначний результат від алгоритму щодо визначення траєкторії об’єкта, в якому ми зацікавлені.

При порівнянні альтернативи на швидкодію, перевагу будемо віддавати алгоритму, який буде мати мінімальний сумарний час роботи.

Розглянемо наступні альтернативи:

– застосувати навчання з підкріпленням;

– застосувати сигма-точковий фільтр Калмана;

– застосувати нейронні мережі.

Відомо, що нейронна мережа досить ресурсомістка, через складність структури. Але, завдяки своїм методам аналізу досить проста у реалізації, так як працює завдяки повторення одноманітних дій, але з використанням попередніх результатів, що значно заощаджує час на опис програми. Однак вона вимагає збереження результатів попередньої ітерації для донавчання, а це вимагає більше оперативної пам’яті.

Сигма-точковий фільтр Калмана досить простий з точки зору математич-

ного розуміння, але для конкретного завдання він вимагає багато розгалужень для більш точного розв'язку, а це значно збільшує час тестування алгоритму. По показнику надійності він поступається нейронній мережі, але є більш надійним за навчання з підкріпленням.

Навчання з підкріпленням – найменш ресурсоємнісним та найпростішим у реалізації алгоритм, порівняно з нейронними мережами та сигма-точковим фільтром Калмана. Однак це, також, найменш точний та надійний метод. За часом відпрацювання алгоритму він прирівнюється до використання сигма-точкового фільтра Калмана, тому що формула потребує набагато більше арифметичних дій. На рисунку 1.10 показано ієрархічну модель процесу аналізу незадоволеностей.

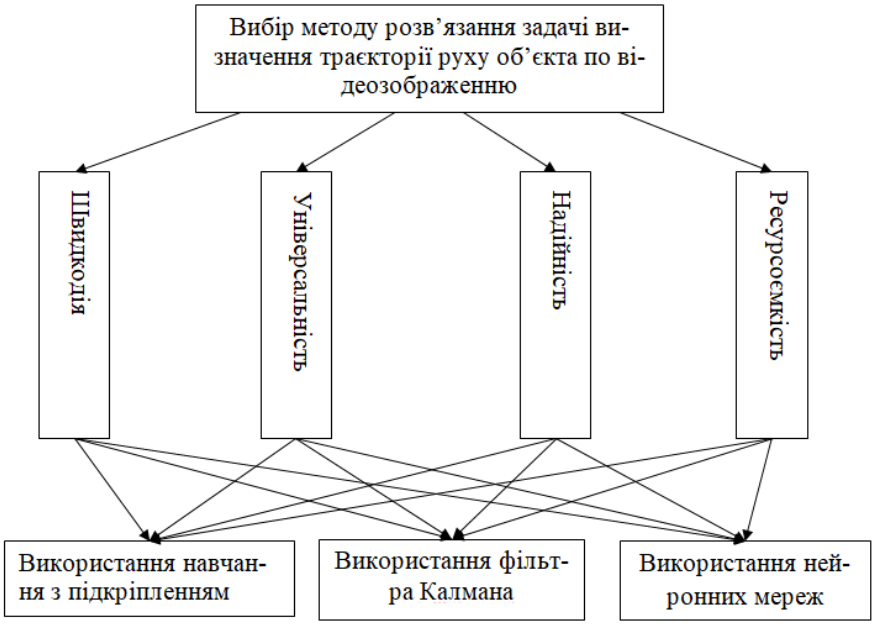

Рисунок 1.10 – Ієрархічна структура задачі вибору методу

1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій

Використовуючи метод попарних порівнянь, ми побудуємо модель процесу аналізу незадоволеностей. Аналіз незадоволеностей включає такі рівні:

– перший рівень – незадоволеності;

– другий рівень – класифікація незадоволеностей;

– третій рівень – характеристики компонентів, які впливають на результат поставленої задачі.

На рисунку 1.10 показано нульовий рівень проблеми.

На першому рівні аналізу проблеми ми побудуємо матрицю попарних порівнянь критеріїв, щоб оцінити вплив кожної незадоволеності на поставлену проблему. Результати представлені в таблиці 1.1.для встановлення відносної важливості критеріїв використовується шкала Сааті [3]. Ця шкала дозволяє поставити у відповідність ступеням переваги одного об’єкта порівняння над іншим деяке число.

Таблиця 1.1 – Матриця попарних порівнянь критеріїв

	К1	К2	К3	К4	Вектор пріоритетів
К1	1,00	8,00	0,20	1,00	0,19
К2	0,13	1,00	0,13	0,17	0,04
К3	5,0	8,00	1,00	4,00	0,59
К4	1,00	6,00	0,25	1,00	0,18

Щоб знайти індекс узгодженості рахуємо суми елементів матриці за стовбцями:

$$y_1 = 1,00 + 0,13 + 5,0 + 1,00 = 7,13,$$

$$y_2 = 8,00 + 1,00 + 8,00 + 6,00 = 23,00,$$

$$y_3 = 0,20 + 0,13 + 1,00 + 0,25 = 1,58,$$

$$y_4 = 1,00 + 0,17 + 4,00 + 1,00 = 6,17.$$

Тоді:

$$\lambda_{max} \approx 7,13 \cdot 0,19 + 23,00 \cdot 0,04 + 1,58 \cdot 0,59 + 6,17 \cdot 0,18 = 4,26$$

та індекс узгодженості:

$$CI^k = \frac{4,26 - 4}{4 - 1} = 0,09.$$

Так як, матриця попарних порівнянь критеріїв є матрицею четвертого порядку, то відношення узгодженості:

$$CR^K = \frac{CI^K}{RI^K} = \frac{0,09}{0,90} = 0,10\,.$$

Далі формуємо матриці попарних альтернатив за кожним критерієм для того, щоб порівняти методи між собою за кожним критерієм окремо.

Таблиця 1.2 – Матриця порівнянь за критерієм К1

К1	А1	А2	А3	Вектор пріоритетів
А1	1,00	0,13	0,17	0,06
А2	8,00	1,00	6,00	0,74
А3	6,00	0,17	1,00	0,20

Щоб знайти індекс узгодженості рахуємо суми елементів матриці за стовбцями:

$$y_1 = 1,00 + 8,00 + 6,00 = 15,00,$$
$$y_2 = 0,13 + 1,00 + 0,17 = 1,29,$$
$$y_3 = 0,17 + 6,00 + 1,00 = 7,17.$$

Тоді:

$$\lambda_{max} \approx 0,06 \cdot 15,00 + 0,74 \cdot 1,29 + 0,20 \cdot 7,17 = 3,26$$

та індекс узгодженості:

$$CI_{K1}^{A} = \frac{3,26-3}{3-1} = 0,13.$$

Так як, матриця попарних порівнянь альтернатив – це матриця третього порядку, то відношення узгодженості:

$$CR_{K1}^{A} = \frac{CI^{k}}{0,58} = 0,22.$$

Таблиця 1.3 – Матриця порівнянь за критерієм К2

К2	А1	А2	А3	Вектор пріоритетів
А1	1,00	6,00	8,00	0,76
А2	0,17	1,00	3,00	0,17
А3	0,13	0,33	1,00	0,07

Щоб знайти індекс узгодженості рахуємо суми елементів матриці за стовбцями:

$$y_1 = 1,00 + 0,17 + 0,13 = 1,29,$$
$$y_2 = 6,00 + 1,00 + 0,33 = 7,33,$$
$$y_3 = 8,00 + 3,00 + 1,00 = 12,00.$$

Тоді:

$$\lambda_{max} \approx 0,76 \cdot 1,29 + 0,17 \cdot 7,33 + 0,07 \cdot 12,0 = 3,07$$

та індекс узгодженості:

$$CI_{K2}^{A} = \frac{3,07 - 3}{3 - 1} = 0,02.$$

Відношення узгодженості:

$$CR_{K2}^{A} = \frac{CI^{k}}{0,58} = 0,04.$$

Таблиця 1.4 – Матриця порівнянь за критерієм К3

К3	А1	А2	А3	Вектор пріоритетів
А1	1,00	0,13	0,20	0,07
А2	8,00	1,00	3,00	0,66
А3	5,00	0,33	1,00	0,27

Щоб знайти індекс узгодженості рахуємо суми елементів матриці за стовбцями:

$$y_1 = 1,00 + 8,00 + 5,00 = 14,00,$$
$$y_2 = 0,13 + 1,00 + 0,33 = 1,46,$$
$$y_3 = 0,20 + 3,00 + 1,00 = 4,20.$$

Тоді:

$$\lambda_{\max} \approx 0,07 \cdot 14,00 + 0,66 \cdot 1,46 + 0,27 \cdot 4,20 = 3,04$$

та індекс узгодженості:

$$CI_{K3}^{A} = \frac{3,04-3}{3-1} = 0,01.$$

Відношення узгодженості:

$$CR_{K3}^{A} = \frac{CI^{k}}{0,58} = 0,03.$$

Таблиця 1.5 – Матриця порівнянь за критерієм К4

К4	А1	А2	А3	Вектор пріоритетів
А1	1,00	0,33	0,25	0,12
А2	3,00	1,00	2,00	0,52
А3	4,00	0,50	1,00	0,36

Щоб знайти індекс узгодженості рахуємо суми елементів матриці за стовбцями:

$$y_1 = 1,00 + 3,00 + 4,00 = 8,00,$$
$$y_2 = 0,33 + 1,00 + 0,50 = 1,83,$$
$$y_3 = 0,25 + 2,00 + 1,00 = 3,25.$$

Тоді:

$$\lambda_{\max} \approx 0,12 \cdot 8,00 + 0,52 \cdot 1,83 + 0,36 \cdot 3,25 = 3,11$$

та індекс узгодженості:

$$CI_{K4}^{A} = \frac{3{,}11-3}{3-1} = 0{,}04.$$

Відношення узгодженості:

$$CR_{K4}^{A} = \frac{CI^{k}}{0{,}58} = 0{,}06.$$

1.2.3 Модель вирішення проблеми

Необхідно розрахувати вектор глобальних пріоритетів альтернатив. Для цього знаходимо добуток:

$$\vec{p} = \begin{bmatrix} 0{,}06 & 0{,}76 & 0{,}07 & 0{,}12 \\ 0{,}74 & 0{,}17 & 0{,}66 & 0{,}52 \\ 0{,}20 & 0{,}07 & 0{,}27 & 0{,}36 \end{bmatrix} \cdot \begin{bmatrix} 0{,}19 \\ 0{,}04 \\ 0{,}59 \\ 0{,}18 \end{bmatrix} = \begin{bmatrix} 0{,}10 \\ 0{,}63 \\ 0{,}27 \end{bmatrix}.$$

Обчислимо індекс узгодженості та відношення узгодженості для всієї ієрархії:

$$CI = 0{,}09 + 0{,}13 \cdot 0{,}19 + 0{,}02 \cdot 0{,}04 + 0{,}01 \cdot 0{,}59 + 0{,}04 \cdot 0{,}18 = 0{,}13,$$

$$RI = 0{,}90 + 0{,}58 = 1{,}48,$$

$$CR = \frac{CI}{RI} = \frac{0{,}13}{1{,}48} = 0{,}09,$$

що теж можна вважати доброю узгодженістю.

Найбільша компонента вектора локальних пріоритетів критеріїв відповідає третьому критерію. Таким чином, за критеріями порівняння ми маємо наступні пріоритети: швидкодія, надійність, ресурсоємність, універсальність.

Порівнюючи альтернативи за обраними критеріями, ми отримали вектор глобальних пріоритетів, де найбільша компонента відповідає другій альтернативі – використанню сигма-точкового фільтра Калмана.

1.3 Змістовна та формальна постановка задачі

1.3.1 Змістовна постановка задачі

Вхідними даними буде захоплене зображення об’єкта (в нашому випадку автомобіля), а створені обмежувальні прямокутники – це вихідні дані. Відстань вимірюється за допомогою лідара з похибкою.

Рисунок 1.11 – Виділення об’єкту обмежувальним прямокутником

Поставлено задачу відстеження траєкторії об’єкта:

– прогноз майбутнього місця розташування об'єкта;

– коригування прогнозу на основі нових вимірювань;

– зниження шуму через неточне виявлення.

Задано значення вимірюваних з похибкою координат положення по X і Y.

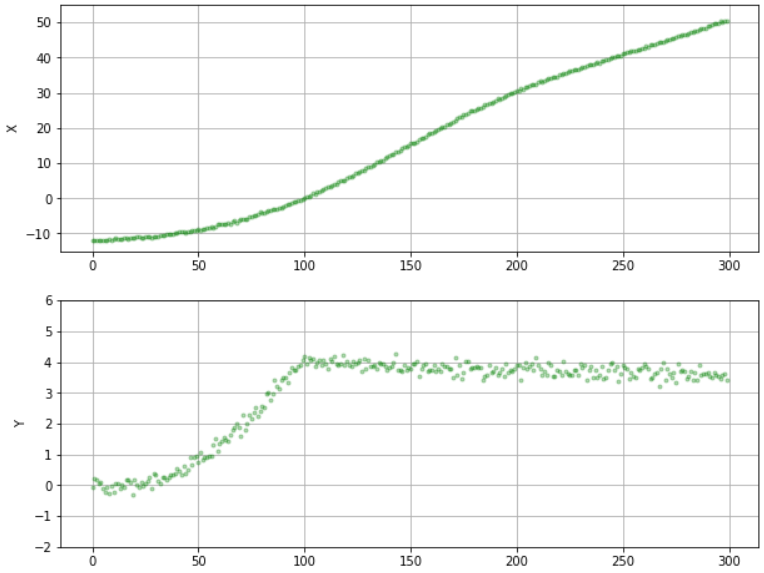

Рисунок 1.12 – Значення вимірюваних з похибкою координат по X і Y

Необхідно обчислити реальне положення об’єкту та розрахувати швидкості для прогнозування шляху.

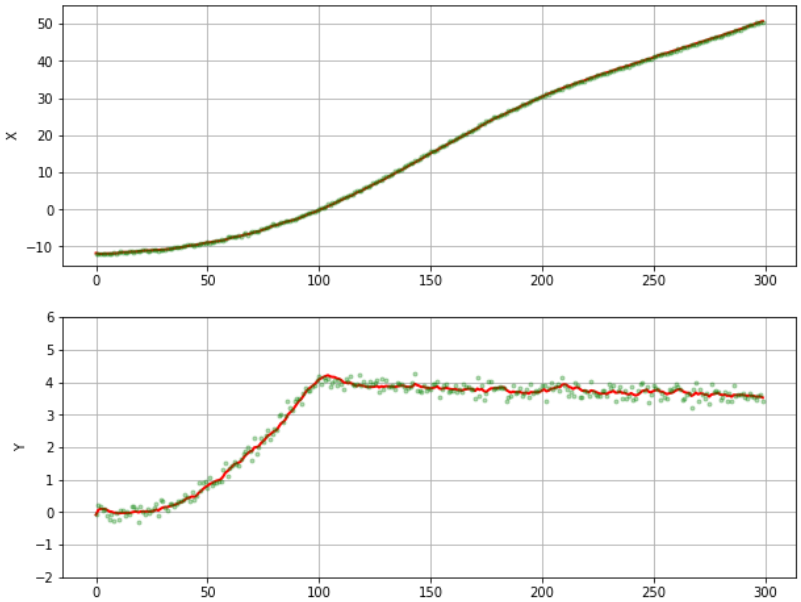

Рисунок 1.13 – Реальне положення об’єкту по X і Y

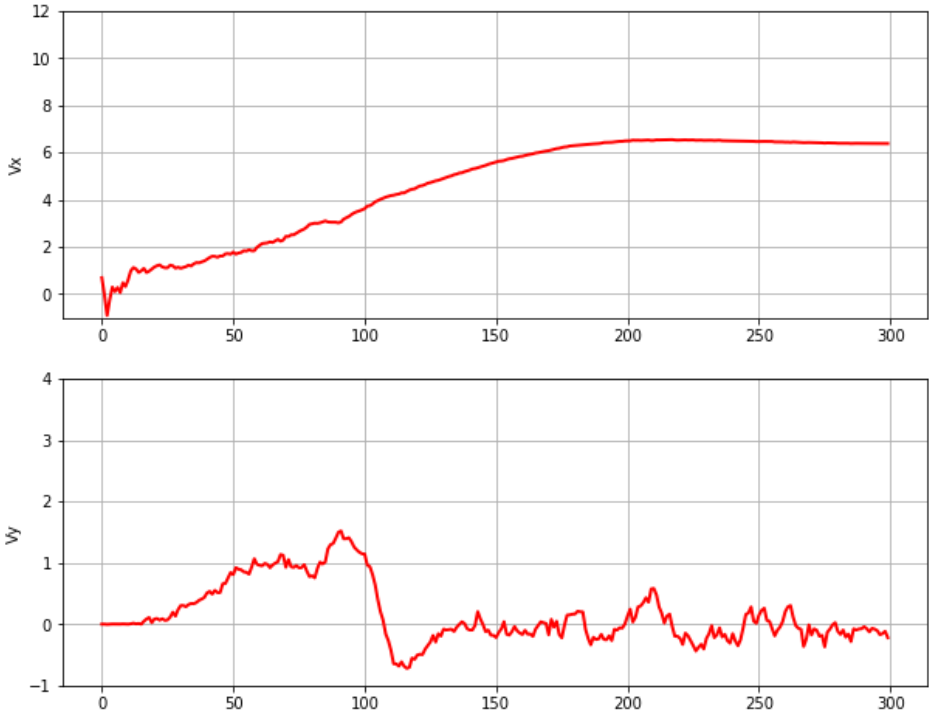

Рисунок 1.14 – Розраховані швидкості по X і Y

1.3.2 Формальна постановка задачі

Нехай V – поток відеоданих, який можна представити у вигляді послідовності зображень $I_0, I_1, I_2, ..., I_{N-1}$, де N – кількість кадрів відео. Положення об’єкта визначається розташуванням прямокутника, в рамках якого він знаходиться. Задача відстеження об’єкта полягає в тому, щоб кожному кадру I_k оригінального відео поставити у відповідність сукупність положень об'єктів B_k. Задача зводиться до побудови відображення φ

$$\varphi : \{I_k, k = \overline{0, N-1}\} \to \{B_k, k = \overline{0, N-1}\}, \tag{1.1}$$

де $B_k = \{b_l^k, l = \overline{0, s_k - 1}\}$ – множина прямокутників, в рамках яких знаходиться об’єкт, виявлених на кадрі I_k. При цьому кожний прямокутник b_l^k:

$$b_l^k = \left((x_1^l, y_1^l), (x_2^l, y_2^l), [s^l, c^l]\right), \quad (1.2)$$

де $(x_1^l, y_1^l), (x_2^l, y_2^l)$ – координати лівого верхнього та правого нижнього кутів прямокутника;

$s^l \in \mathbb{R}$ – вірогідність того, що об'єкт виявлено правильно;

c^l – клас, якому належить об’єкт.

1.4 Постановка задач дослідження

Прогнозування шляху є фундаментальним завданням для оцінки того, як пішоходи або транспортні засоби збираються рухатися в сцені. Оскільки прогнозування шляху як завдання комп'ютерного зору використовує відео в якості вхідного сигналу, різні дані, що використовуються для прогнозування, такі як навколишнє середовище цілі і внутрішній стан цілі, повинні бути оцінені з відео на додаток до прогнозування шляхів.

Провівши детальний аналіз системи та отримавши вихідні дані до роботи сформулюємо задачу дослідження:

– розглянути способи вилучення функцій із відео;

– дослідити методи прогнозування шляху;

– скласти загальну математичну модель, що дозволить вирішити задачу визначення траєкторії руху об’єкта.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДА РОЗВ’ЯЗАННЯ

2.1 Дослідження прогнозування шляху

Прогнозування шляху – це завдання оцінки шляху або траєкторії, по якій буде рухатися ціль (наприклад, пішохід або транспортний засіб). Прогнозування шляху по відео є важливим завданням, якому приділяється велика увага, оскільки очікується, що воно буде мати багато потенційних застосувань, таких як аналіз камер спостереження, автомобілі з автоматичним управлінням і автономна навігація роботів.

Прогнозування шляху має оцінювати набагато більше інформації – такої як інформація про навколишнє середовище, напрямку руху і стан цілей прогнозування – ніж інші прості завдання розпізнавання зображень [4]. В результаті методи прогнозування часто будуються поверх інших завдань комп'ютерного зору, таких як виявлення пішоходів, розпізнавання пішохідних атрибутів і семантична сегментація. Більш того, в завданні прогнозування майбутні спостереження прогнозованих трас недоступні. В задачах виявлення та відстеження пішоходів спостереження з минулого в теперішнє використовуються для визначення місця розташування та відстеження цілі в поточному кадрі відео. Навпаки, завдання прогнозування локалізує і прогнозує розташування цілі в майбутніх кадрах відео, використовуючи зроблені спостереження до теперішнього часу і попередньої інформації про навколишнє середовище і знання про рух цілі.

Прогнозування шляху вивчалося десятиліттями в області робототехніки. На станціях і в аеропортах роботам необхідно переміщатися, не заважаючи присутнім людям, і планувати шлях ефективного руху в навколишньому середовищі. Прогнозування шляху необхідно для досягнення таких завдань. Проте на додаток до інформації з камер роботи можуть використовувати інформацію від датчиків багатьох типів, таких як датчик LIDAR, для отримання тривимірної (3D) геометрії сцени. Оточення, в якому робот може переміщатися, іноді явно вказується у вигляді карти оточення. Це дослідження присвячене методам

прогнозування шляху.

Існує альтернативна задача, яка називається раннім розпізнаванням [5], яка передбачає майбутню поведінку людини в відео. Це завдання передбачає майбутні дії в відео, але виключається з опитування, тому що передбачені категорії є дискретними, тоді як передбачені шляхи є послідовностями безперервних розташувань.

Оскільки завдання передбачення шляху в області комп'ютерного зору є важким і складним, був запропонований цілий ряд різних методів. Загальний підхід показаний на рисунку 2.1. В якості вхідних даних дається відео (або кадр із відео) на додаток до місця розташування цілі в поточному кадрі або послідовності розташування за останні кадри в кілька секунд. Функції, корисні для прогнозування, потім витягуються з відео (або кадрів) для прогнозування шляху в майбутніх кадрах. В огляді на рисунок 2.1 є дві важливі частини: (b) вилучення ознак, де береться багато особливостей, щоб зрозуміти середовище і ціль; і (c) прогнозування шляху, де пропонується безліч методів.

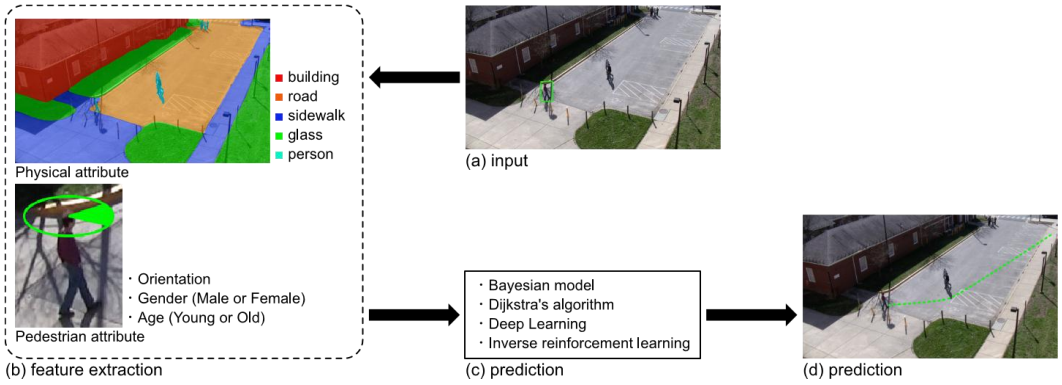

Рисунок 2.1 – Прогнозування шляху

2.2 Методи прогнозування шляху

2.2.1 Байєсівські моделі

Перший підхід використовує онлайн-байєсівські фільтри, такі як фільтр

Калмана (KF) і фільтри часток, і виводить модель для прогнозування шляхів. Таке моделювання являє собою внутрішні стан і спостереження як змінні і визначає ймовірнісні моделі, припускаючи, що спостереження – це зашумлені внутрішні стани. Цей підхід повторює етап прогнозування, який обчислює поточні внутрішні стани з попередніх стані, і етап оновлення, що оновлює поточні стани за допомогою спостережень. У загальній конфігурації внутрішні стани – це фактичні координати пішоходів, а спостереження є координатами, отриманими шляхом виявлення пішоходів. Це відстеження людини, якщо ми застосовуємо підхід для відстеження з минулого в теперішній час, і прогнозування шляху, якщо ми тільки повторюємо етап прогнозування, щоб отримати послідовність координат пішоходу, без етапу оновлення; немає ніяких майбутніх спостережень.

Шнайдер Н. використав розширений фільтр Калмана для оновлення внутрішнього стану пішохода перед автомобілем. Це була рання робота по прогнозованому шляху, що показана, яку примітивну інформацію (наприклад, швидкість ходьби і прискорення) можна використовувати для прогнозування шляхів [7].

Замість використання онлайн-байєсівських фільтрів в деяких роботах використовувалась динамічна байєсівська сіть. Розглядався більш обмежений випадок. Оцінка того, чи буде пішохід йти по проїжджій частині перед автомобілем, на якому встановлена камера. Визначили модель DBN з лінійною динамічною системою, яка перемикається (SLDS), що показана на рисунку 2.2 і яка використовує функції, вилучені з фільму , такі як орієнтація голови пішохода, відстань до автомобіля, і відстань між пішоходом і проїжджою частиною. Цей метод працює краще, ніж використання тільки координат для виявлення пішоходів.

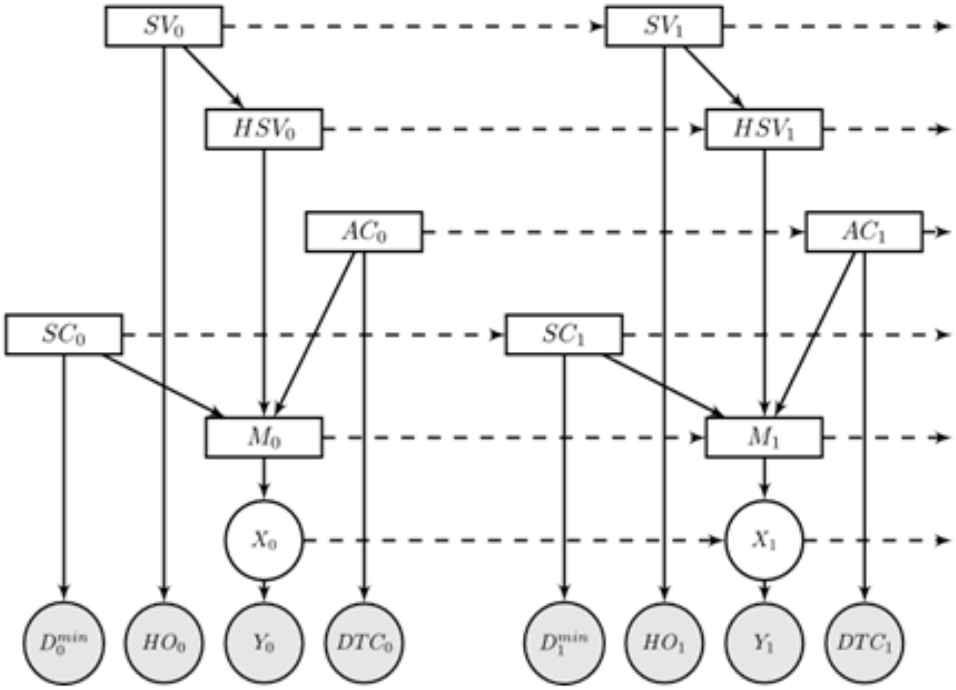

Рисунок 2.2 – Графічна модель DBN з SLDS

2.2.2 Мінімізація енергії

Підхід Байєса, описаний вище, є онлайновим, в якому він оцінює координати пішохода кадр за кадром у відео. Інший (автономний або пакетний) підхід – це метод мінімізації енергії, який оцінює всю послідовність координат одночасно. Цей підхід створює двовимірний сітковий граф сцени і призначає витрати на переміщення до країв на графіку, а потім знаходить комбінацію ребер, що дає мінімальну енергію. Це сформульовано як задача найкоротшого шляху, розв’язана за допомогою методу Дейкстри. Тому точність прогнозу в значній мірі залежить від того, як визначається вартість.

Хуан запропонував метод прогнозування шляху з використанням одного зображення. Спочатку вилучається патч, що містить ціль, щоб оцінити орієнтацію цілі [8]. Далі, вартість переміщення через місце розташування патча оцінюється шляхом порівняння текстури навколишніх патчів. На додаток до цієї вартості, передбачувана орієнтація цілі використовується як обмеження і додається

до граничних ваг. Уокер порівнював текстуру суперпікселів, використовуючи патчі уздовж шляху, який відстежувала ціль без залучення будь-якої навчальної процедури.

Інформація про зовнішній вигляд (текстуру) сцени може бути використана для визначення функції вартості, але також можуть використовуватися об'єкти сцени. Тодорович. припускав, що пішоходи вибрали свою ціль (наприклад, багажник з їжею) відповідно до їх потенційних потреб (голоду), і визначили карти вартості, де пішоходів приваблюють об'єкти на сцені.

2.2.3 Глибоке навчання

Методи глибокого навчання, такі ж як методи, що пов'язані з використанням CNN і довготривалою пам'яттю (LSTM), використовувалися для прогнозування шляху з моменту появи систем глибокого навчання. Методи цього типу беруть в якості вхідних даних координати цілі протягом останніх декількох кадрів і створюють серію координат цілі у декількох послідовних кадрах. Вилучення ознак, описане в останньому розділі, явно не виконується, оскільки вилучення ознак і прогнозування явно не відокремлені в моделях глибокого навчання.

Було запропоновано декілька методів, які використовують LSTM для обробки шляхів, які являють собою послідовності двовимірних координат. Гоель запропонував рівень соціального об'єднання (S-об'єднання) для уникнення зіткнень між пішоходами. Пішохід представлений LSTM, а виходи прихованого рівня LSTMs інших людей підключені до рівня S-об’єднання пішохода. Цей рівень дозволяє LSTM пішохода представляти просторові відносини з людьми, які знаходяться близько (наприклад, відстань один від одного), і таким чином прогнозувати шлях, уникаючи зіткнення.

LSTM має обмеження довготривалої пам'яті; тобто шляхи в далекому майбутньому важко передбачити. Фернандо передбачав необхідність більш продуманої довготривалої пам'яті і запропонував деревовидну мережу пам'яті,

яка ієрархічно вибирає корисну інформацію про минуле, що зберігається в комірках пам'яті, і працює краще, ніж інші моделі LSTM.

Крім LSTM, CNN також використовується для безпосередніх прогнозів. Ванг запропонував поведінку – CNN, яка передбачає майбутній шлях від минулого шляху. Цей метод спочатку створює тривимірні розріджені дані, канали яких зберігають двовимірні координати пішохода останніх декількох кадрів. Розріджені тривимірні дані кодуються за допомогою шарів згортки та об'єднання, а потім декодуються з використанням шарів деконволюції. До кожного каналу закодованої інформації також додані карти зміщення місця розташування для врахування різних поведінок в різних місцях сцени, таких як розташування входів і перешкод.

2.2.4 Зворотнє навчання з підкріпленням

Три вищевикладені підходи є прикладами навчання під наглядом або без нагляду, тоді як підхід, представлений тут, є прикладом навчання з підкріпленням (RL). RL вивчає політику для вирішення дій, які повинні виконувати агенти в поточному стані в середовищі. RL зазвичай визначається як марковський процес прийняття рішень, який вивчає оптимальну політику, що дозволяє агенту приймати найкращі дії, що максимізують винагороду. На рисунку 2.3 показано, що агент RL є ціллю прогнозування, середовище – це сцена, представлена у вигляді відео, статус – місце розташування пішохода, а дія– рух пішохода.

RL необхідно визначити винагороду за перехід з одного стану в інший, що вказує на те, наскільки добре діє агент. Однак важко явно визначити функцію винагороди для практичних задач, таких як задача прогнозування шляху. Ця проблема називається проблемою проектування винагороди, і зворотне навчання з підкріпленням (IRL) є одним з підходів, прийнятих для вирішення проблеми. IRL оцінює винагороди, які відтворюють оптимальні послідовності дій, і визначає дії агента на етапі тестування з розрахунковою винагородою,

так щоб агент міг виконувати подібні дії.

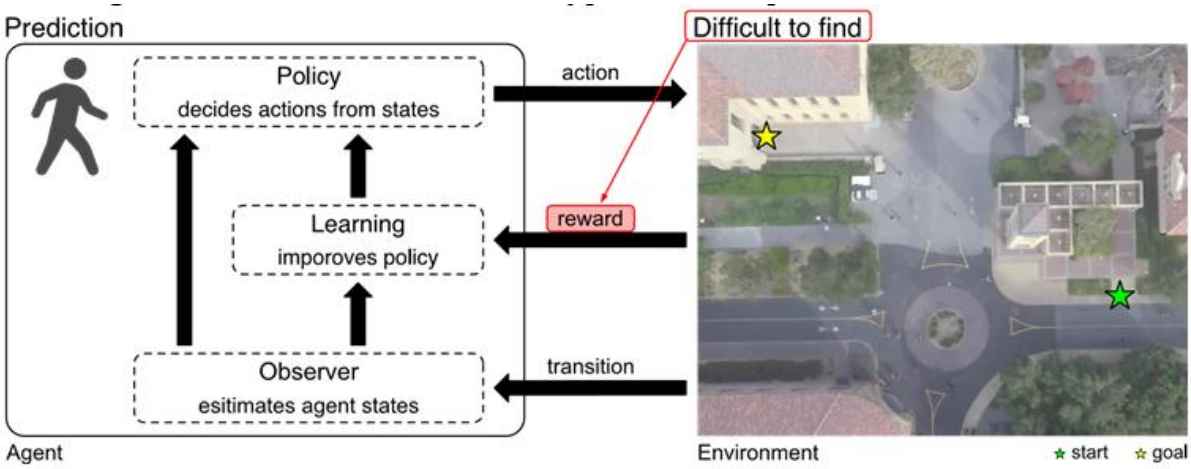

Рисунок 2.3 – Огляд RL

IRL використовується для навчання та управління оптимальним рухом роботів [8]. Хеберт вперше запровадив IRL для прогнозування шляхів на основі бачення. Замість того, щоб оцінювати цільові місця розташування, він оцінив дії, які агент може виконувати в певний час або місце, і передбачив можливі шляхи, послідовно застосовуючи оцінені дії до поточного цільового місця розташування. Таким чином, це завдання, називається прогнозуванням активності, на відміну від прогнозування шляху, яке безпосередньо оцінює місця розташування цілі в майбутньому. Прогнозування діяльності є набагато більш складним завданням, ніж прогнозування шляху, хоча воно має великий потенціал з точки зору адаптації різноманітних прогнозів, до кожного можливого застосування.

Хеберт припускав, що фізичні атрибути сцени сильно впливають на пішохідні шляхи, і використовували атрибути сцени оцінені семантичною сегментацією як карти властивостей. Винагороди кожного атрибута сцени визначаються внутрішнім продуктом особливостей карт і ваги векторів, а оптимальні ваги оцінюються по даних навчання. Для прогнозування послідовності дій, які приходять до заздалегідь визначеної мети, генерується шляхом надання цілі та поточного розташування цільового пішохода. Лі використовував аналогічний підхід для прогнозування шляхів футболістів у відеоіграх. Вей представив тео-

рію ігор, що називається фіктивною грою, щоб передбачити шляхи декількох пішоходів, які досягають цілі, уникаючи зіткнень між пішоходами.

Без будь-яких заздалегідь визначених цілей, Рехдер запропонував мережу призначення для оцінки цілі з використанням останніх декількох кадрів. Передбачувана ціль та екологічні атрибути, отримані за допомогою повністю згорткової мережі, використовувалися для прогнозування пішохідних маршрутів.

2.2.5 Інші підходи

Більшість методів прогнозування можна розділити на один з чотирьох описаних вище підходів, але є й інші підходи. Модель соціальної сили передбачає енергію, яку називають «соціальною силою», яка діє між пішоходами та об'єктами на сцені, і генерує рух пішоходів через взаємодію за допомогою сили. Ямагуті запропонував модель з додатковими станами, такими як переваги пішоходів, швидкість ходьби, цілі та існування інших людей, що йдуть разом. Ця робота мотивувалася прагненням підвищити точність відстеження пішоходів, але виконала прогнозування шляху для оцінки запропонованої моделі. Алахі запропонував соціальні сили декількох класів для уникнення зіткнень. Вони оцінювали «особливості соціальної чутливості», використовуючи відстані між іншими людьми, і застосовували кластеризацію функцій Kmeans для отримання декількох кластерів поведінки уникнення. Кластер цільової поведінки уникнення оцінювався з використанням цільової функції, і шляхи кластера потім проектувалися назад до сцени для прогнозування.

Оптичний потік, вилучений з автомобільних камер, був використаний Келером для прогнозування пішохідних доріжок. Він використовував оптичні потоки над останніми кількома кадрами та обчислював гістограми орієнтації як особливості руху пішоходів. Послідовність гістограм була використана для отримання подібних сцен у навчальному наборі, а шляхи вилучених сцен були зіставлені назад до сцени для прогнозування.

Використання структури марковського процесу було запропоновано Рехдером. Він використовував звичайний розподіл і розподіл фон Мізеса для представлення стану (розташування) і швидкості пішохода, і послідовно оцінював стан, приймаючи продукти цих розподілів на кожному часовому кроці для прогнозування. Щоб підвищити точність, ціль пішохода була оцінена по характеристикам навколишнього середовища для обмеження напрямку руху.

Підхід на основі пошуку, показаний на рисунку 2.4, був запропонований Парком для прогнозування майбутнього шляху у відео, яке відображає вигляд від першої особи. Він спочатку вилучив сюжетні функції, використовуючи AlexNet, а потім знайшли подібні сцени в навчальному наборі, порівнюючи витягнуті характеристики. Шляхи вилучених навчальних зразків були відображені на відео.

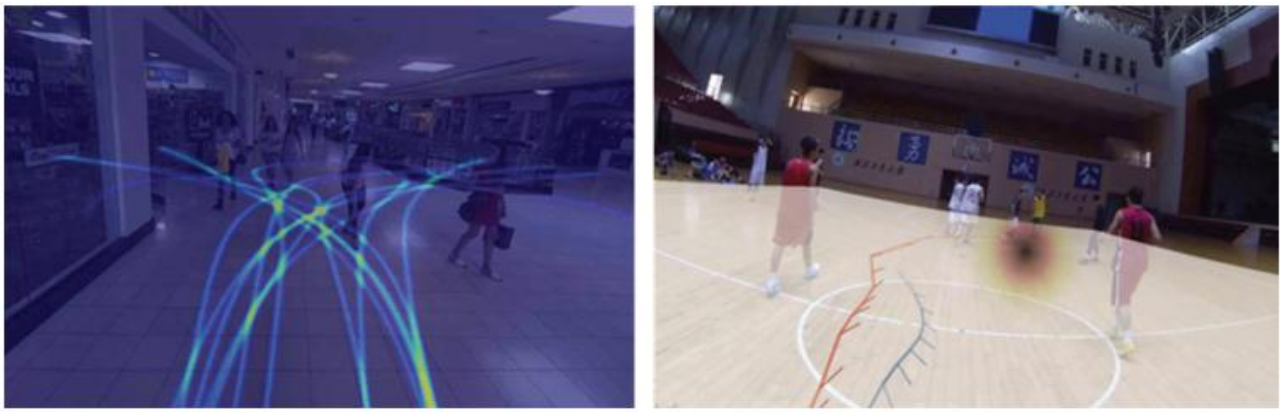

Рисунок 2.4 – Прогнозування з відео від першої особи

Вони передбачали шляхи навіть у сценах з оклюзіями, оцінюючи області, що стоять за оклюзійними об'єктами, такими як стіни та перешкоди. Він поширив цю роботу на прогнозування декількох баскетболістів на ігровій сцені. В одному відео від першої особи вони оцінювали область «спільної уваги», на яку, зазвичай, звертали увагу деякі гравці. Кілька шляхів були передбачені шляхом вибору оптимального шляху кожного гравця та мінімізації цільової функції, визначеної оціненою областю спільної уваги, розташування гравців і шляхів, що проектуються назад до сцени.

2.3 Сигма-точковий Фільтр Калмана

Розширений фільтр Калмана (РФК) – найбільш широко застосовуваний алгоритм оцінки стану для нелінійних систем. Однак РФК може бути важко налаштувати і він часто дає ненадійні оцінки, якщо нелінійності системи є серйозними. Це пояснюється тим, що РФК покладається на лінеаризацію для розповсюдження середнього значення та коваріації стану. То ж давайте розглянемо сигма-точковий фільтр Калмана– розширення фільтра Калмана, що зменшує помилки лінеаризації РФК.

Використання сигма-точкового фільтр Калмана може забезпечити значне покращення порівняно з розширеним фільтром Калмана.

2.3.1 Математичні очікування та коваріації нелінійних перетворень

Розглянемо нелінійне перетворення

$$\begin{aligned} y_1 &= r\,\mathrm{Cos}(\theta), \\ y_2 &= r\,\mathrm{Sin}(\theta). \end{aligned} \tag{2.1}$$

Це стандартне перетворення полярних прямокутних координат. Наприклад, ми можемо мати датчик, який вимірює діапазон r та кут θ, і ми хочемо перетворити виміряні дані у прямокутні координати y_1 та y_2. Перетворення координат можна записати більш загально як

$$y = h(x), \tag{2.2}$$

де y – двоелементний вихід $h(z)$, а двоелементний вектор x визначається як

$$x = \begin{bmatrix} r \\ \theta \end{bmatrix}.$$

Припустимо, що x_1 (яке є діапазоном r) є випадковою величиною із математичним очікуванням 1 та стандартним відхиленням σ_r. Припустимо, що x_2 (що є кутом θ) є випадковою величиною із математичним очікуванням $\frac{\pi}{2}$ і стандартним відхиленням σ_θ. Іншими словами, математичні очікування компонентів x наведені як $\bar{r} = 1$ і $\bar{\theta} = \frac{\pi}{2}$. Крім того, ми будемо вважати, що r і θ незалежні, а їх функції щільності ймовірностей симетричні відносно їх математичних очікувань (наприклад, функція Гауса або однорідна).

2.3.1.1 Середнє значення нелінійного перетворення

Початковий розгляд вищезазначеної проблеми, разом з рівнянням (2.1), змусив би нас думати, що y_1 має математичне очікування 0, а y_2 має математичне очікування 1. Крім того, підхід лінеаризації приведе нас до того ж висновку. Якщо виконати лінеаризацію першого порядку рівняння (2.2) і взяти очікуване значення обох сторін, ми отримаємо

$$\begin{aligned} \bar{y} = \mathrm{E}\left[h(x)\right] &\approx \mathrm{E}\left[h(\bar{x}) + \left.\frac{\partial h}{\partial x}\right|_{\bar{x}} \left(x - \bar{x}\right) \right] = \\ &= h(x) + \left.\frac{\partial h}{\partial x}\right|_{\bar{x}} \mathrm{E}\left(x - \bar{x}\right) = h(\bar{x}) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{2.3}$$

Наше передбачення разом з аналізом лінеаризації першого порядку, приводять нас до такого ж висновку [9]. Однак розглянемо цю проблему більш строго, щоб перевірити наш попередній аналіз. Ми можемо записати r і θ як

$$r = \bar{r} + \tilde{r},$$
$$\theta = \bar{\theta} + \tilde{\theta},$$

де $\tilde{r}$ і $\tilde{\theta}$ – це просто відхилення r і θ від їхніх математичних очікувань. Строгий аналіз математичного значення y_1 може бути виконаний таким чином:

$$\bar{y}_1 = \mathrm{E}\left(r\,\mathrm{Cos}(\theta)\right) = \mathrm{E}\left[(\bar{r} + \tilde{r})\mathrm{Cos}\left(\bar{\theta} + \tilde{\theta}\right)\right] =$$
$$= \mathrm{E}\left[(\bar{r} + \tilde{r})\left(\mathrm{Cos}\left(\bar{\theta}\right)\mathrm{Cos}\left(\tilde{\theta}\right) - \mathrm{Sin}\left(\bar{\theta}\right)\mathrm{Sin}\left(\tilde{\theta}\right)\right)\right].$$

Виконуючи множення, пам'ятаємо, що $\tilde{r}$ і $\tilde{\theta}$ не залежать від симетричних функцій щільності ймовірностей, і приймаючи очікуване значення, це призводить до

$$\bar{y}_1 = \bar{r}\,\mathrm{Cos}\left(\bar{\theta}\right) = 0. \qquad (2.4)$$

Наше передбачення та апроксимація першого порядку $\bar{y}_1$ були підтверджені суворим аналізом. Повторимо процес для y_2:

$$\bar{y}_2 = \mathrm{E}\left(r\,\mathrm{Sin}(\theta)\right) = \mathrm{E}\left[(\bar{r} + \tilde{r})\mathrm{Sin}\left(\bar{\theta} + \tilde{\theta}\right)\right] =$$
$$= \mathrm{E}\left[(\bar{r} + \tilde{r})\left(\mathrm{Sin}\left(\bar{\theta}\right)\mathrm{Cos}\left(\tilde{\theta}\right) + \mathrm{Cos}\left(\bar{\theta}\right)\mathrm{Sin}(\tilde{\theta})\right)\right].$$

Виконуючи множення, пам'ятаємо, що $\tilde{r}$ і $\tilde{\theta}$ не залежать від симетричних функцій щільності ймовірностей, і приймаючи очікуване значення, це призводить до

$$\bar{y}_2 = \bar{r}\,\mathrm{Sin}\left(\bar{\theta}\right)\mathrm{E}\left(\mathrm{Cos}\left(\tilde{\theta}\right)\right) = \mathrm{E}\left(\mathrm{Cos}\left(\tilde{\theta}\right)\right). \qquad (2.5)$$

Припустимо, що $\tilde{\theta}$ рівномірно розподілений між $\pm\theta_m$. У цьому випадку ми можемо обчислити

$$\bar{y}_2 = \mathrm{E}\left(\mathrm{Cos}\left(\tilde{\theta}\right)\right) = \frac{\mathrm{Sin}\left(\theta_m\right)}{\theta_m}. \qquad (2.6)$$

Очікувалося, що ми отримаємо 1, як відповідь на підтвердження рівняння (2.3), але натомість вийшло деяке число менше 1 (звернемо увагу, що $\frac{\mathrm{Sin}\left(\theta_m\right)}{\theta_m} < 1$ для всіх $\theta_m > 0$ та $\lim_{\theta_m \to 0} \frac{\mathrm{Sin}\left(\theta_m\right)}{\theta_m} = 1$). Аналіз виявляє проблему з нашим початковим передбаченням та лінеаризацією першого порядку, яку ми виконували раніше. Математичне очікування y_2 дійсно буде менше 1. Це можна побачити, розглянувши графік із 300 випадково сформованих значень r і θ, де $\tilde{r}$ рівномірно розподілено між $\pm 0{,}01$, а $\tilde{\theta}$ рівномірно розподілено між $\pm 0{,}35$ радіанами . Мала дисперсія $\tilde{r}$ і велика дисперсія $\tilde{\theta}$ призводять до дугоподібного розподілу точок, як показано на рисунку 2.5. В результаті такого дугоподібного розподілу $\bar{y}_2 < 1$.

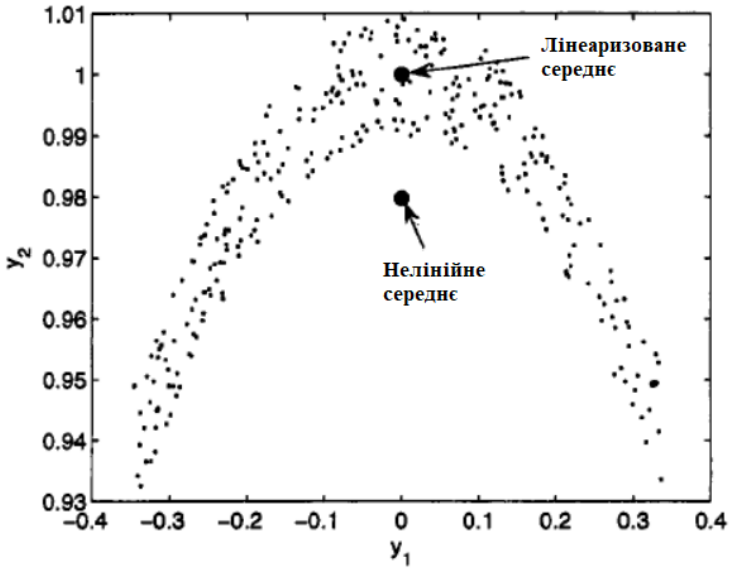

Рисунок 2.5 – Лінеаризоване та нелінійне середнє значення 300 випадково згенерованих точок

Це не приклад фільтрації Калмана. Але оскільки РФК використовує лінеаризацію першого порядку для оновлення математичного очікування стану, цей приклад показує тип помилки, яка може прокрастись в РФК, коли вона застосовується до нелінійної системи [9].

Для більш загального аналізу математичного очікування нелінійного перетворення згадаємо, що $y = h(z)$ можна розкласти в ряд Тейлора в околі $\bar{x}$ наступним чином:

$$y = h(x) = h(\bar{x}) + D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \frac{1}{3!}D_{\tilde{x}}^3 h + ..., \tag{2.7}$$

де $\tilde{x} = x - \bar{x}$. Таким чином, математичне очікування y може бути розширено, як

$$\begin{aligned} \bar{y} &= \mathrm{E}\left[h(\bar{x}) + D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \frac{1}{3!}D_{\tilde{x}}^3 h + ... \right] = \\ &= h(\bar{x}) + \mathrm{E}\left[D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \frac{1}{3!}D_{\tilde{x}}^3 h + ... \right]. \end{aligned} \tag{2.8}$$

Використовуючи $D_{\tilde{x}}h$, ми бачимо, що

$$\mathrm{E}\left[D_{\tilde{x}}h\right] = \mathrm{E}\left[\sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i} h(x)\Big|_{x=\bar{x}} \right] = \sum_{i=1}^{n} \mathrm{E}(\tilde{x}_i) \frac{\partial}{\partial x_i} h(x)\Big|_{x=\bar{x}} = 0,$$

тому що $\mathrm{E}(\tilde{x}_i) = 0$. Так само, ми бачимо, що

$$\mathrm{E}\left[D_{\tilde{x}}^3 h \right] = \mathrm{E}\left[\left(\sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i} \right)^3 h(x)\Big|_{x=\bar{x}} \right] = 0.$$

Це пояснюється тим, що сума у наведеному рівнянні складається лише з

моментів третього порядку $(\mathrm{E}\left(\tilde{x}_1^3\right), \mathrm{E}\left(\tilde{x}_1^2\tilde{x}_2\right), \ldots)$. Ці очікувані значення завжди будуть дорівнювати нулю. Аналогічно всі непарні доданки у рівнянні (2.7) будуть дорівнювати нулю, що призводить до спрощення

$$\bar{y} = h\left(\bar{x}\right) + \frac{1}{2!}\mathrm{E}\left[D_{\tilde{x}}^2 h\right] + \frac{1}{4!}\mathrm{E}\left[D_{\tilde{x}}^4 h\right] + \ldots \tag{2.9}$$

Це показує, чому математичне очікування у рівнянні (2.3) було обчислено неправильно; цей розрахунок був правильним лише до першого порядку. Якщо ми апроксимуємо $\tilde{y}$ для нашого перетворення, від полярного до прямокутного, використовуючи члени до другого порядку з рівняння (2.8), ми отримаємо

$$\bar{y} \approx h\left(\bar{x}\right) + \frac{1}{2!}\mathrm{E}\left[D_{\tilde{x}}^2 h\right] = h\left(\bar{x}\right) + \frac{1}{2}\mathrm{E}\left[\left(\sum_{i=1}^{2} x_i \frac{\partial}{\partial x_i}\right)^2 h(x)\Big|_{x=\bar{x}}\right] =$$

$$= h\left(\bar{x}\right) + \frac{1}{2}\left(\mathrm{E}\left(\tilde{x}_1^2\right)\frac{\partial^2 h(x)}{\partial x_1^2}\bigg|_{x=\bar{x}} + 2\mathrm{E}\left(\tilde{x}_1\tilde{x}_2\right)\frac{\partial^2 h(x)}{\partial x_1 \partial x_2}\bigg|_{x=\bar{x}} + \mathrm{E}\left(\tilde{x}_2^2\right)\frac{\partial^2 h(x)}{\partial x_2^2}\bigg|_{x=\bar{x}}\right) =$$

$$= h\left(\bar{x}\right) + \frac{1}{2}\left(\begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix} + \sigma_\theta^2\begin{bmatrix}-r\,\mathrm{Cos}(\theta)\\-r\,\mathrm{Sin}(\theta)\end{bmatrix}_{x=\bar{x}}\right) = \begin{bmatrix}0\\1\end{bmatrix} + \frac{1}{2}\sigma_\theta^2\begin{bmatrix}0\\-1\end{bmatrix}.$$

Таким чином, ми отримаємо

$$\begin{aligned} \bar{y}_1 &\approx 0, \\ \bar{y}_2 &\approx 1 - \frac{\sigma_\theta^2}{2} = 1 - \frac{\mathrm{E}\left(\tilde{\theta}^2\right)}{2}. \end{aligned} \tag{2.10}$$

Точне значення $\bar{y}_2$ у рівнянні (2.5) дорівнюватиме $\mathrm{E}\left(\mathrm{Cos}\left(\tilde{\theta}\right)\right)$. Наближений вираз, знайдений у рівнянні (2.9), є першими двома ненульовими членами

розкладу ряду Тейлора на $\mathrm{E}\left(\mathrm{Cos}\left(\tilde{\theta}\right)\right)$.

2.3.1.2 Коваріація нелінійного перетворення

Розглянемо коваріацію випадкової величини, яка піддається нелінійному перетворенню. Коваріація y задається як

$$P_y = \mathrm{E}\left[\left(y-\bar{y}\right)\left(y-\bar{y}\right)^T\right]. \tag{2.11}$$

Використаємо рівняння (2.7) та (2.8), щоб записати $\left(y-\bar{y}\right)$ як

$$y-\bar{y} = \left[h\left(\bar{x}\right)+D_{\tilde{x}}h+\frac{1}{2!}D_{\tilde{x}}^2h+...\right]-\left[h\left(\bar{x}\right)+\frac{1}{2!}\mathrm{E}\left(D_{\tilde{x}}^2h\right)+\frac{1}{4!}\mathrm{E}\left(D_{\tilde{x}}^4h\right)+...\right]=$$
$$=\left[D_{\tilde{x}}h+\frac{1}{2!}D_{\tilde{x}}^2h+...\right]-\left[\frac{1}{2!}\mathrm{E}\left(D_{\tilde{x}}^2h\right)+\frac{1}{4!}\mathrm{E}\left(D_{\tilde{x}}^4h\right)+...\right].$$

Підставимо цей вираз у рівняння (2.9) і використаємо той самий тип міркувань, що і в попередньому розділі, щоб побачити, що всі непарні доданки в очікуваному значенні дорівнюють нулю (припускаючи, що $\tilde{x}$ є нульовим математичним очікуванням з симетричною функцією щільності ймовірності). Це призводить до

$$P_y = \mathrm{E}\left[D_{\tilde{x}}h\left(D_{\tilde{x}}h\right)^T\right]+E\left[\frac{D_{\tilde{x}}h\left(D_{\tilde{x}}^3h\right)^T}{3!}+\frac{D_{\tilde{x}}^2h\left(D_{\tilde{x}}^2h\right)^T}{2!2!}+\frac{D_{\tilde{x}}^3h\left(D_{\tilde{x}}\right)^T}{3!}\right]+$$
$$+\mathrm{E}\left(\frac{D_{\tilde{x}}^2h}{2!}\right)\mathrm{E}\left(\frac{D_{\tilde{x}}^2h}{2!}\right)^T+... \tag{2.12}$$

Перший доданок у правій частині наведеного рівняння запишемо як

$$\mathrm{E}\left[D_{\tilde{x}}h\left(D_{\tilde{x}}h\right)^T \right] = \mathrm{E}\left[\left(\sum_{i=1}^{n} \tilde{x}_i \right) \left.\frac{\partial h}{\partial x_i}\right|_{x=\tilde{x}} \left(...\right)^T \right] =$$

$$= \mathrm{E}\left[\sum_{i,j} \tilde{x}_i \left.\frac{\partial h}{\partial x_i}\right|_{x=\tilde{x}} \left.\frac{\partial h^T}{\partial x_j}\right|_{x=\tilde{x}} \tilde{x}_j \right] = \sum_{i,j} H_i E\left(\tilde{x}_i \tilde{x}_j\right) H_j^T = \sum_{i,j} H_i P_{ij} H_j^T , \qquad (2.13)$$

де вектор частинних похідних H_i та очікуване значення P_{ij} визначаються наведеним вище рівнянням. Рівняння у цій формі можна записати як

$$\mathrm{E}\left[D_{\tilde{x}}h\left(D_{\tilde{x}}h\right)^T \right] = \left.\frac{\partial h}{\partial x}\right|_{x=\tilde{x}} P \left.\frac{\partial h^T}{\partial x}\right|_{x=\tilde{x}} = HPH^T , \qquad (2.14)$$

де матриця частинних похідних H та матриця коваріації P визначаються наведеним вище рівнянням. H_i, у рівнянні (2.11) – i-й стовпець H_i, а P_{ij} у рівнянні (2.11) є елементом у i-му рядку та j-му стовпці $P = \mathrm{E}\left(\tilde{x}\tilde{x}^T\right)$. Ми можемо використати це у рівнянні (2.10), щоб записати коваріацію нелінійного перетворення $y = h\left(x\right)$ наступним чином:

$$P_y = HPH^T + E\left[\frac{D_{\tilde{x}}h\left(D_{\tilde{x}}^3 h\right)^T}{3!} + \frac{D_{\tilde{x}}^2 h\left(D_{\tilde{x}}^2 h\right)^T}{2!2!} + \frac{D_{\tilde{x}}^3 h\left(D_{\tilde{x}}h\right)^T}{3!} \right] + \qquad (2.15)$$

$$+ \mathrm{E}\left(\frac{D_{\tilde{x}}^2 h}{2!} \right) \mathrm{E}\left(\frac{D_{\tilde{x}}^2 h}{2!} \right)^T + ...$$

Це повне розкладання рядів Тейлора для коваріації нелінійного перетворення.

У РФК ми використовуємо лише перший доданок цього розширення для апроксимації коваріації помилки оцінки [9]. Наприклад, якщо вимірювання $y = h(x) + v$, то з рівняння $P_y = H(k)P^-(k)H(k)^T + R(k)$ ми бачимо, що коваріація у апроксимується як $P_y = HP_xH^T + R$, де H – часткова похідна h відносно x, а R – коваріація v. Так само, якщо стан поширюється як $x_{k+1} = f(x_k) + w_k$, то з рівняння $P^+(k) = P^-(k) + P_{xy}P_y^{-1}P_{xy}^T$ ми бачимо, що коваріація x приблизно оновлюється як $P_k^- = FP_{k-1}^+F^T + Q$, де F – часткова похідна $f(x)$ відносно x, а Q – коваріація w_k. Однак ці коваріаційні наближення можуть призвести до значних помилок, якщо основні функції $h(x)$ та $f(x)$ є вкрай нелінійними.

Наприклад, розглянемо нелінійне перетворення, введене на початку розділу. Апроксимація лінійної коваріації означала б, що $P_y \approx HP_xH^T$, де H і P_x подані як

$$H = \left.\frac{\partial h^T}{\partial x}\right|_{x=\tilde{x}} = \begin{bmatrix} \mathrm{Cos}(\theta) & -r\,\mathrm{Sin}(\theta) \\ \mathrm{Sin}(\theta) & r\,\mathrm{Cos}(\theta) \end{bmatrix}_{x=\bar{x}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

$$P_x = \mathrm{E}\left(\begin{bmatrix} r-\bar{r} \\ \theta-\bar{\theta} \end{bmatrix} [\ldots]^T \right) = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}.$$

Це дає нам P_y наступним чином.

$$P_y \approx HP_xH^T = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}.$$

Ми отримали наближення P_y. Однак більш ретельний аналіз P_y можна провести за допомогою рівнянь (2.1), (2.4) та (2.6):

$$P_y = \mathrm{E}\left[(y-\bar{y})(y-\bar{y})^T\right] = \mathrm{E}\left[\begin{pmatrix} r\,\mathrm{Cos}(\theta) \\ r\,\mathrm{Sin}(\theta) - \dfrac{\mathrm{Sin}(\theta_m)}{\theta_m} \end{pmatrix}(\ldots)^T\right] =$$

$$= \mathrm{E}\begin{bmatrix} r^2\,\mathrm{Cos}^2(\theta) & r^2\,\mathrm{Cos}(\theta)\mathrm{Sin}(\theta) - r\,\mathrm{Cos}(\theta)\dfrac{\mathrm{Sin}(\theta_m)}{\theta_m} \\ r^2\,\mathrm{Cos}(\theta)\mathrm{Sin}(\theta) - r\,\mathrm{Cos}(\theta)\dfrac{\mathrm{Sin}(\theta_m)}{\theta_m} & \left(r\,\mathrm{Sin}(\theta) - \dfrac{\mathrm{Sin}(\theta_m)}{\theta_m}\right)^2 \end{bmatrix}. \quad (2.16)$$

Знову використаємо наше припущення, що r і θ незалежні, r рівномірно розподілене з математичним очікуванням 1 і стандартним відхиленням σ_r, і $\theta = \frac{\pi}{2} + \tilde{\theta}$, з $\tilde{\theta}$ рівномірно розподіленим між $\pm\theta_m$. Обчислимо

$$\mathrm{E}\left(r^2\right) = 1 + \sigma_r^2,$$

$$\mathrm{E}\left(\mathrm{Cos}^2\left(\tilde{\theta}\right)\right) = \frac{1 - \mathrm{E}\left(\mathrm{Cos}\left(2\tilde{\theta}\right)\right)}{2},$$

$$\mathrm{E}\left(\mathrm{Cos}\left(2\tilde{\theta}\right)\right) = \frac{\mathrm{Sin}\left(2\theta_m\right)}{2\theta_m},$$

$$\mathrm{E}\left(\mathrm{Sin}\left(\theta\right)\right) = \mathrm{E}\left(\mathrm{Cos}\left(\tilde{\theta}\right)\right) = \frac{\mathrm{Sin}\left(\theta_m\right)}{\theta_m}.$$

Використаємо ці вирази у рівнянні (2.15) для обчислення

$$P_y = \begin{bmatrix} \dfrac{1}{2}\left(1+\sigma_r^2\right)\left(1 - \dfrac{\mathrm{Sin}\left(2\theta_m\right)}{2\theta_m}\right) & 0 \\ 0 & \dfrac{1}{2}\left(1+\sigma_r^2\right)\left(1 + \dfrac{\mathrm{Sin}\left(2\theta_m\right)}{2\theta_m}\right) - \dfrac{\mathrm{Sin}^2\left(\theta_m\right)}{\theta_m^2} \end{bmatrix}. \quad (2.17)$$

Ця матриця описує двовимірний еліпс, де $P_y(1,1)$ означає квадрат довжини осі y_1, а $P_y(2,2)$ означає квадрат довжини осі y_2. На рисунку 2.6 показана

лінеаризована коваріація, визначена рівнянням (2.15), і точна коваріація, визначена рівнянням (2.16). Лінеаризована коваріація відцентрована навколо лінеаризованого математичного очікування, а точна коваріація – навколо точного середнього. Лінеаризована коваріація не є дуже хорошим наближенням до точної коваріації, принаймні не в напрямку y_2.

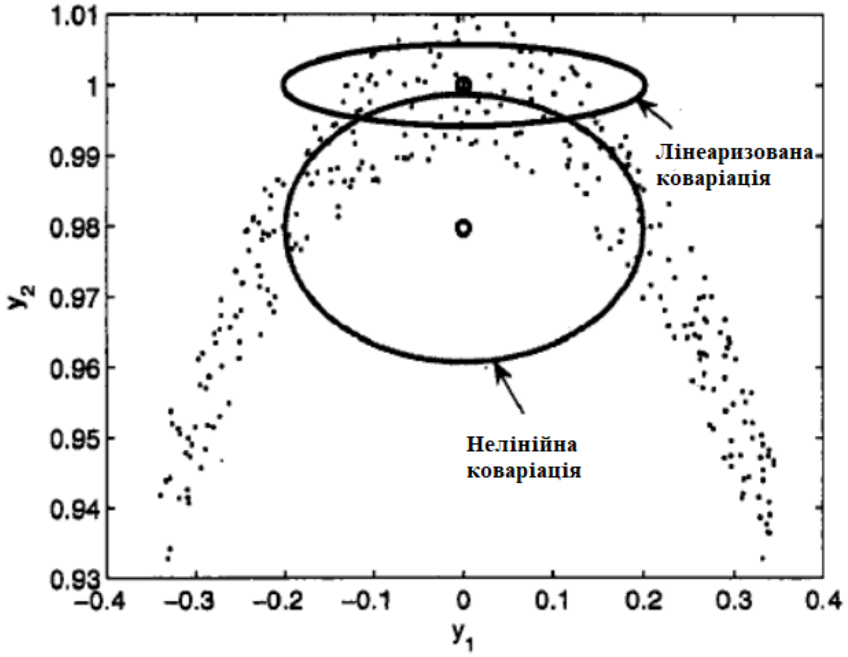

Рисунок 2.6 – Лінеаризоване та нелінійне середнє та коваріація

Це не приклад фільтрації Калмана. Але оскільки РФК використовує лінеаризацію першого порядку для оновлення коваріації стану, цей приклад показує тип помилки, яка може прокрастись до РФК, коли він застосовується до нелінійної системи.

2.3.2 Сигма-точкове перетворення

Проблема нелінійних систем полягає в тому, що важко перетворити функцію щільності ймовірності на основі загальної нелінійної функції. У попередньому розділі ми отримали точні нелінійні перетворення математичного очіку-

вання та коваріації, але лише для простого двовимірного перетворення. Розширений фільтр Калмана працює за принципом, що лінеаризоване перетворення математичних очікувань і коваріацій приблизно дорівнює справжньому нелінійному перетворенню, але в попередньому розділі було виявлено, що наближення може бути незадовільним.

Сигма-точкове перетворення базується на двох основних принципах. По-перше, легко виконати нелінійне перетворення в одній точці (а не всієї функції щільності ймовірності). По-друге, не надто складно знайти множину окремих точок у просторі стану, в якому вибіркова функція щільності ймовірності апроксимує істину функцію щільності ймовірності вектору стану [10].

Припустимо, що ми знаємо середнє значення $\bar{x}$ та коваріацію P вектора x. Потім ми знаходимо набір детермінованих векторів, які називаються сигма-точками, математичне очікування та коваріація яких дорівнюють $\bar{x}$ і P. Далі ми застосовуємо нашу відому нелінійну функцію $y = h(x)$ до кожного детермінованого вектора, щоб отримати перетворені вектори. Математичне очікування та коваріація змінених векторів дадуть хорошу оцінку істинного математичного очікування та коваріації y. Це ключ до перетворення без запаху.

Як приклад, припустимо, що x є $n \times 1$ вектором, який змінюється нелінійною функцією $y = h(x)$. Обираємо $2n$ сигма-точки $x^{(i)}$ наступним чином:

$$
\begin{aligned}
x^{(i)} &= \bar{x} + \tilde{x}^{(i)}, \ i = 1,...,2n, \\
\tilde{x}^{(i)} &= \left(\sqrt{nP}\right)^T, \ i = 1,...,n, \\
\tilde{x}^{(n+i)} &= -\left(\sqrt{nP}\right)^T, \ i = 1,...,n,
\end{aligned}
\tag{2.18}
$$

де $\sqrt{nP}$ а – матричний квадратний корінь nP такий, що $\left(\sqrt{nP}\right)^T \sqrt{nP} = nP$, і $\left(\sqrt{nP}\right)_i$, i-й рядок $\sqrt{nP}^1$. У наступних кількох підрозділах ми побачимо, як мо-

же використовуватися ансамблеве математичне очікування з зазначених вище сигма-точок, для апроксимації математичного очікування та коваріації нелінійно перетвореного вектора.

2.3.2.1 Апроксимація математичного очікування

Припустимо, що ми маємо вектор x з відомим математичним очікуванням $\bar{x}$ і коваріацією P, нелінійною функцією $y = h(x)$, і ми хочемо апроксимувати математичне очікування y. Перетворимо кожну окрему сигма-точку рівняння (2.16), використовуючи нелінійну функцію $h(\cdot)$. А потім візьмемо зважену суму перетворених сигма-точок для апроксимації математичного очікування y. Перетворені сигма-точки обчислюються наступним чином:

$$y^{(i)} = h\left(x^{(i)}\right), \ i = 1,...,2n. \tag{2.19}$$

Справжнє математичне очікування y позначається як $\bar{y}$. Апроксимоване математичне очікування y позначається як $\bar{y}_u$ і обчислюється наступним чином:

$$\bar{y}_u = \sum_{i=1}^{2n} W^{(i)} y^{(i)}. \tag{2.20}$$

Вагові коефіцієнти $W^{(i)}$ визначаються наступним чином:

$$W^{(i)} = \frac{1}{2n}, \ i = 1,...,2n. \tag{2.21}$$

Таким чином, рівняння (2.18) можна записати у вигляді

$$\overline{y}_u = \frac{1}{2n}\sum_{i=1}^{2n} y^{(i)}\,. \tag{2.22}$$

Тепер обчислимо значення $\overline{y}_u$, щоб побачити, наскільки воно відповідає справжньому математичному очікуванню y. Щоб зробити це, ми спочатку використовуємо рівняння $f(x) = f(\overline{x}) = D_{\overline{x}} f + \frac{1}{2!} D_{\overline{x}}^2 f + \frac{1}{3!} D_{\overline{x}}^3 f + \ldots$, щоб розкласти кожний елемент в рівнянні (2.20) у ряд Тейлора в околі $\overline{x}$. Це призводить до

$$\begin{aligned} \overline{y}_u &= \frac{1}{2n}\sum_{i=1}^{2n}\left(h(\overline{x}) + D_{\tilde{x}^{(i)}} h + \frac{1}{2!} D_{\tilde{x}^{(i)}}^2 h + \ldots \right) = \\ &= h(\overline{x}) + \frac{1}{2n}\sum_{i=1}^{2n}\left(D_{\tilde{x}^{(i)}} h + \frac{1}{2!} D_{\tilde{x}^{(i)}}^2 h + \ldots \right). \end{aligned} \tag{2.23}$$

Для будь-якого цілого числа $k \geq 0$ ми маємо

$$\sum_{j=1}^{2n} D_{\tilde{x}^{(j)}}^{2k+1} h = \sum_{j=1}^{2n}\left[\left(\sum_{i=1}^{n} \tilde{x}_i^{(j)} \frac{\partial}{\partial x_i} \right)^{2k+1} h(x)\Big|_{x=\overline{x}} \right] = \sum_{j=1}^{2n}\left[\left(\sum_{i=1}^{n} \tilde{x}_i^{(j)} \right)^{2k+1} \frac{\partial^{2k+1}}{\partial x_i^{2k+1}} h(x)\Big|_{x=\overline{x}} \right] =$$

$$= \sum_{i=1}^{n}\left[\left(\sum_{j=1}^{2n} \tilde{x}_i^{(j)} \right)^{2k+1} \frac{\partial^{2k+1}}{\partial x_i^{2k+1}} h(x)\Big|_{x=\overline{x}} \right] = 0\,,$$

тому що з рівняння (2.16) $\tilde{x}^{(j)} = -\tilde{x}^{(n+j)}$ $(j = 1,\ldots,n)$. Отже, усі непарні доданки у рівнянні (2.21) дорівнюють нулю, і ми маємо

$$\overline{y}_u = h(\overline{x}) + \frac{1}{2n}\sum_{i=1}^{2n}\left(\frac{1}{2!} D_{\tilde{x}^{(i)}}^2 h + \frac{1}{4!} D_{\tilde{x}^{(i)}}^4 h + \ldots \right) =$$

$$= h(\bar{x}) + \frac{1}{2n}\sum_{i=1}^{2n}\frac{1}{2!}D^2_{\tilde{x}^{(i)}}h + \frac{1}{2n}\sum_{i=1}^{2n}\left(\frac{1}{4!}D^4_{\tilde{x}^{(i)}}h + \frac{1}{6!}D^6_{\tilde{x}^{(i)}}h + ...\right). \qquad (2.24)$$

Подивимось на другий доданок у правій частині наведеного рівняння:

$$\frac{1}{2n}\sum_{i=1}^{2n}\frac{1}{2!}D^2_{\tilde{x}^{(i)}}h = \frac{1}{2n}\sum_{k=1}^{2n}\frac{1}{2!}\left(\sum_{i=1}^{n}\tilde{x}_i^{(k)}\frac{\partial}{\partial x_i}\right)^2 h(x)\Big|_{x=\bar{x}} =$$

$$= \frac{1}{4n}\sum_{k=1}^{2n}\sum_{i,j=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i\partial x_j}h(x)\Big|_{x=\bar{x}} = \frac{1}{4n}\sum_{i,j=1}^{n}\sum_{k=1}^{2n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i\partial x_j}h(x)\Big|_{x=\bar{x}} =$$

$$= \frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i\partial x_j}h(x)\Big|_{x=\bar{x}},$$

де знову був використаний той факт із рівняння (2.16), що $\tilde{x}^{(k)} = -\tilde{x}^{(k+n)}\ (k = 1,...,n)$. Підставимо $\tilde{x}_i^{(k)}$ та $\tilde{x}_j^{(k)}$ із рівняння (2.16) у наведене вище рівняння, щоб отримати

$$\frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} = \frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\left(\sqrt{nP}\right)_{k_i}\left(\sqrt{nP}\right)_{k_j}\frac{\partial^2 h}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} =$$

$$= \frac{1}{2n}\sum_{i,j=1}^{n}nP_{ij}\frac{\partial^2 h(x)}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} = \frac{1}{2}\sum_{i,j=1}^{n}P_{ij}\frac{\partial^2 h(x)}{\partial x_i\partial x_j}\Big|_{x=\bar{x}}.$$

Таким чином, рівняння (2.22) можна записати у вигляді

$$\bar{y}_u = h(\bar{x}) + \frac{1}{2}\sum_{i,j=1}^{n}P_{ij}\frac{\partial^2 h}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} + \frac{1}{2n}\sum_{i=1}^{2n}\left(\frac{1}{4!}D^4_{\tilde{x}^{(i)}}h + \frac{1}{6!}D^6_{\tilde{x}^{(i)}}h + ...\right). \qquad (2.25)$$

Справжнє математичне очікування y задається рівнянням (2.9), як

$$\bar{y}_u = h(\bar{x}) + \frac{1}{2!}\mathrm{E}\left[D_{\tilde{x}}^2 h\right] + \frac{1}{4!}\mathrm{E}\left[D_{\tilde{x}}^4 h\right] + \ldots \qquad (2.26)$$

Запишемо другий доданок у правій частині наведеного рівняння таким чином:

$$\frac{1}{2!}\mathrm{E}\left[D_{\tilde{x}}^2 h\right] = \frac{1}{2!}\mathrm{E}\left[\left(\sum_{i=1}^{n}\tilde{x}_i\frac{\partial}{\partial x_i}\right)^2 h(x)\Big|_{x=\bar{x}}\right] = \frac{1}{2!}\mathrm{E}\left[\sum_{i,j=1}^{n}\tilde{x}_i\tilde{x}_j\frac{\partial^2 h}{\partial x_i \partial x_j}\Big|_{x=\bar{x}}\right] =$$

$$= \frac{1}{2!}\sum_{i,j=1}^{n}\mathrm{E}\left(\tilde{x}_i\tilde{x}_j\right)\frac{\partial^2 h}{\partial x_i \partial x_j}\Big|_{x=\bar{x}} = \frac{1}{2!}\sum_{i,j=1}^{n}P_{i,j}\frac{\partial^2 h}{\partial x_i \partial x_j}\Big|_{x=\bar{x}}.$$

Звернемо увагу, що $\bar{y}$ можна записати з рівняння (2.24) як

$$\bar{y} = h(\bar{x}) + \frac{1}{2}\sum_{i,j=1}^{n}P_{i,j}\frac{\partial^2 h}{\partial x_i \partial x_j}\Big|_{x=\bar{x}} + \frac{1}{4!}\mathrm{E}\left[D_{\tilde{x}}^4 h\right] + \frac{1}{6!}\mathrm{E}\left[D_{\tilde{x}}^6 h\right] + \ldots.$$

Порівнюючи це з рівнянням (2.23), ми бачимо, що $\bar{y}_u$, (наближене математичне очікування y) відповідає справжньому математичному очікуванню y до третього порядку, тоді як лінеаризація відповідає істинному математичному очікуванню y лише до першого порядку. Якщо ми обчислимо $\bar{y}_u$, використовуючи рівняння (2.16), (2.17) і (2.20), тоді значення $\bar{y}_u$ буде відповідати дійсному математичному очікуванню y аж до третього порядку. Найбільша складність цього алгоритму – квадратний корінь матриці, необхідний у рівнянні (2.16). Але перетворення без запаху має обчислювальну перевагу в тому, що матрицю лінеаризації H не потрібно обчислювати [10]. Звичайно, найбільшою перевагою перетворення без запаху (щодо лінеаризації) є підвищена точність перетворення математичного очікування.

2.3.2.2 Апроксимація коваріації

Припустимо, що ми хочемо апроксимувати коваріацію нелінійно перетвореного вектора x. Маємо n-елементний вектор x із відомим математичним очікуванням $\bar{x}$ і коваріацією P, і маємо відому нелінійну функцію $y = h(x)$. Оцінимо коваріацію y. Позначимо оцінку як P_u і будемо використовувати наступне рівняння:

$$P_u = \sum_{i=1}^{2n} W^{(i)} \left(y^{(i)} - y_u \right) \left(y^{(i)} - y_u \right)^T = \frac{1}{2n} \sum_{i=1}^{2n} \left(y^{(i)} - y_u \right) \left(y^{(i)} - y_u \right)^T, \qquad (2.27)$$

де вектори $y^{(i)}$ – це перетворені сигма-точки, що відповідають рівнянню (2.17), а $W^{(i)}$ вагові коефіцієнти такі ж, як ті, що наведені у рівнянні (2.19). Розширивши це наближення за допомогою рівнянь $f(x) = f(\bar{x}) = D_{\bar{x}} f + \frac{1}{2!} D_{\bar{x}}^2 f + \frac{1}{3!} D_{\bar{x}}^3 f +$ та (2.22), вийде наступне:

$$P_u = \frac{1}{2n} \sum_{i=1}^{2n} \left(h\left(x^{(i)} \right) - y_u \right) \left(h\left(x^{(i)} \right) - y_u \right)^T = \frac{1}{2n} \sum_{i=1}^{2n} \Big[h(\bar{x}) + D_{\bar{x}^{(i)}} h + \frac{1}{2} D_{\bar{x}^{(i)}}^2 h +$$
$$+ \frac{1}{3!} D_{\bar{x}^{(i)}}^3 h + \ldots - h(\bar{x}) - \frac{1}{2n} \sum_{j=1}^{2n} \left(\frac{1}{2} D_{\bar{x}^{(j)}}^2 h + \frac{1}{4!} D_{\bar{x}^{(j)}}^4 h + \ldots \right) \Big] [\ldots]^T.$$

Множення цього рівняння дає

$$P_u = \frac{1}{2n} \sum_{i=1}^{2n} \Big\{ \left(D_{\bar{x}^{(i)}} h \right) (\ldots)^T + \underbrace{\left[\left(\frac{1}{2} D_{\bar{x}^{(i)}} h \right) \left(D_{\bar{x}^{(i)}}^2 h \right)^T \right]}_{0} + [\ldots]_0^T +$$
$$+ \frac{1}{4!} \left(D_{\bar{x}^{(i)}}^2 h \right) (\ldots)^T - \underbrace{\left[D_{\bar{x}^{(i)}} h \left(\frac{1}{2n} \sum_j \frac{1}{2} D_{\bar{x}^{(i)}}^2 h \right)^T \right]}_{0} - [\ldots]_0^T +$$

$$+\frac{1}{4n^2}\left(\sum_j D^2_{\bar{x}(j)}h\right)(...)^T-\left[\frac{1}{4n}D^2_{\bar{x}(i)}h\left(\sum_j D^2_{\bar{x}(j)}h\right)^T\right]-[...]^T+$$

$$+\left[D_{\bar{x}(i)}h\left(\frac{1}{3!}D^3_{\bar{x}(j)}h\right)^T\right]+[...]^T+\Bigg\}.$$

Деякі доданки у наведеному вище рівнянні дорівнюють нулю, як зазначено вище, оскільки $\tilde{x}^{(i)}=-\tilde{x}^{(i+n)}$ для $i=1,...,n$. Тож апроксимацію коваріації можна записати як

$$P_u=\frac{1}{2n}\sum_{i=1}^{2n}\left(D_{\bar{x}(i)}h\right)(...)^T+HOT,$$

де HOT означає члени вищого порядку (тобто члени до четвертої ступеня і вище). Розширення цього рівняння для P_u, нехтуючи членами вищого порядку, дає

$$P_u=\frac{1}{2n}\sum_{i=1}^{2n}\sum_{j,k=1}^{n}\left(\tilde{x}^{(i)}_j\frac{\partial h(\bar{x})}{\partial x_j}\right)\left(\tilde{x}^{(i)}_k\frac{\partial h(\bar{x})}{\partial x_k}\right)^T.$$

Нагадаємо, що $\tilde{x}^{(i)}_j=-\tilde{x}^{(i+n)}_j$ і $\tilde{x}^{(i)}_k=-\tilde{x}^{(i+n)}_k$ для $i=1,...,n$. Отже, апроксимація коваріації стає

$$P_u=\frac{1}{n}\sum_{i=1}^{n}\sum_{j,k=1}^{n}\left(\tilde{x}^{(i)}_j\frac{\partial h(\bar{x})}{\partial x_j}\right)\left(\tilde{x}^{(i)}_k\frac{\partial h(\bar{x})}{\partial x_k}\right)^T=\sum_{j,k=1}^{n}P_{jk}\frac{\partial h(\bar{x})}{\partial x_j}\left(\frac{\partial h(\bar{x})}{\partial x_k}\right)^T=HPH^T,$$

де остання рівність походить від рівняння (2.14). Порівнюючи це рівняння для P_u із справжньою коваріацією y з рівняння (2.15), ми бачимо, що рівняння

(2.27) апроксимує справжню коваріацію y до третього порядку (тобто тільки доданки до четвертої та вищих степенів є неправильними). Це той самий порядок апроксимації, що і метод лінеаризації, як ми розглядали трохи вище. Однак ми б інтуїтивно очікували, що величина похибки апроксимації без запаху у рівнянні (2.27) буде меншою, ніж лінійна апроксимація HPH^T, оскільки апроксимація без запаху принаймні містить правильно підписані доданки до четвертого порядку і вище, тоді як лінійна апроксимація не містить жодних доданків, крім HPH^T.

Перетворення без запаху можна узагальнити наступним чином.

1. Почнемо з n-елементного вектора x з відомим середнім значенням $\overline{x}$ та коваріацією P. Враховуючи відоме нелінійне перетворення $y = h(x)$, ми хочемо оцінити математичне очікування та коваріацію y, позначені як $\overline{y}_u$, та P_u.

2. Форма $2n$ сигма-точкових векторів $x^{(i)}$ виглядає так:

$$x^{(i)} = \overline{x} + \tilde{x}^{(i)} \quad i = 1,...,2n,$$

$$\tilde{x}^{(i)} = \left(\sqrt{nP}\right)_i^T \quad i = 1,...,n,$$

$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP}\right)_i^T \quad i = 1,...,n,$$

де $\sqrt{nP}$ – матричний квадратний корінь з nP такий, що $\left(\sqrt{nP}\right)^T \sqrt{nP} = nP$ і $\left(\sqrt{nP}\right)_i$ це i-й рядок $\sqrt{nP}$.

3. Перетворення сигма-точок наступним чином:

$$y^{(i)} = h\left(x^{(i)}\right) \; i = 1,...,2n.$$

4. Апроксимація математичного очікування та коваріації y наступним

чином:

$$\bar{y}_u = \frac{1}{2n}\sum_{i=1}^{2n} y^{(i)},$$

$$P_u = \frac{1}{2n}\sum_{i=1}^{2n}\left(y^{(i)} - y_u\right)\left(y^{(i)} - y_u\right)^T.$$

2.3.3 Сигма-точковий фільтр Калмана

Узагальнимо перетворення без запаху, розроблене в попередньому розділі, щоб отримати сигма-точковий фільтр Калмана. Алгоритм фільтра Калмана розмножує математичне очікування та коваріацію системи, використовуючи оновлення часу та оновлення вимірювань. Якщо система лінійна, то математичне очікування та коваріація можуть бути точно оновлені за допомогою фільтра Калмана. Якщо система нелінійна, тоді математичне очікування та коваріацію можна приблизно оновити за допомогою розширеного фільтра Калмана. Однак РФК базується на лінеаризації, і попередній розділ показав, що перетворення без запаху є більш точними, ніж лінеаризація для розмноження математичних очікувань та коваріацій. Тому ми просто замінюємо рівняння РФК на сигма-точкове перетворення, щоб отримати алгоритм сигма-точкового фільтра Калмана [10].

Алгоритм сигма-точкового фільтра Калмана можна узагальнити наступним чином.

1. Маємо n-станову нелінійну систему з дискретним часом, задану формулами

$$x_{k+1} = f\left(x_k, u_k, t_k\right) + w_k,$$

$$y_k = h\left(x_k, t_k\right) + v_k,$$

$$w_k \sim (0, Q_k),$$
$$v_k \sim (0, R_k). \qquad (2.28)$$

2. Сигма-точковий фільтр Калмана ініціалізується наступним чином.

$$\hat{x}_0^+ = \mathrm{E}(x_0),$$
$$P_0^+ = \mathrm{E}\left[\left(x_0 - \hat{x}_0^+\right)\left(x_0 - \hat{x}_0^+\right)^T\right].$$

3. Наступні рівняння оновлення часу використовуються для розмноження оцінки стану та коваріації від одного часу вимірювання до іншого.

Для розповсюдження часового кроку з $(k-1)$ до k спочатку необхідно обрати сигма-точки $x_{k-1}^{(i)}$, як зазначено у рівнянні (2.18), з відповідними змінами, оскільки поточне найкраще припущення щодо математичного очікування та коваріації x_k становить $\hat{x}_{k-1}^+$ і P_{k-1}^+:

$$\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^+ + \tilde{x}^{(i)} \quad i = 1,...,2n,$$
$$\tilde{x}^{(i)} = \left(\sqrt{nP_{k-1}^+}\right)_i^T \quad i = 1,...,n,$$
$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_{k-1}^+}\right)_i^T \quad i = 1,...,n.$$

Використаємо відоме нелінійне системне рівняння $f(\cdot)$. Для перетворення сигма-точок у $\hat{x}_k^{(i)}$ вектори, як показано у рівнянні (2.19), з відповідними змінами, оскільки наше нелінійне перетворення є $f(\cdot)$, а не $h(\cdot)$:

$$\hat{x}_k^{(i)} = f\left(\hat{x}_{k-1}^{(i)}, u_k, t_k\right).$$

Поєднаємо $\hat{x}_k^{(i)}$ вектори, щоб отримати апріорну оцінку стану в момент часу k. Це базується на рівнянні (2.22):

$$\hat{x}_k^- = \frac{1}{2n}\sum_{i=1}^{2n}\hat{x}_k^{(i)}.$$

Оцінимо апріорну коваріацію помилки, як показано у рівнянні (2.27). Однак слід додати Q_{k-1} до кінця рівняння, щоб врахувати шум процесу:

$$P_k^- = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{x}_k^{(i)} - \hat{x}_k^-\right)\left(\hat{x}_k^{(i)} - \hat{x}_k^-\right)^T + Q_{k-1}. \qquad (2.29)$$

4. Тепер, коли рівняння оновлення часу зроблені, ми реалізуємо рівняння вимірювання-оновлення

Виберемо сигма-точки $x_k^{(i)}$, як зазначено у рівнянні (2.18), із відповідними змінами, оскільки поточне найкраще припущення щодо математичного очікування та коваріації x_k дорівнює $\hat{x}_k^-$; і P_k^-:

$$\hat{x}_k^{(i)} = \hat{x}_k^- + \tilde{x}^{(i)} \quad i = 1,...,2n,$$

$$\tilde{x}^{(i)} = \left(\sqrt{nP_k^-}\right)_i^T \quad i = 1,...,n,$$

$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_k^-}\right)_i^T \quad i = 1,...,n.$$

Цей крок за бажанням можна опустити. Тобто замість генерування нових сигма-точок ми можемо використовувати сигма-точки, отримані з оновлення часу [10]. Це заощадить обчислювальні зусилля, якщо ми готові пожертвувати продуктивністю.

Використовуйте відоме нелінійне рівняння вимірювання $h(\cdot)$, щоб перетворити сигма-точки в $\hat{y}_k^-$ вектори (прогнозовані вимірювання), як показано у рівнянні (2.19):

$$\hat{y}_k^{(i)} = h\left(\hat{x}_k^{(i)}, t_k\right).$$

Об'єднаємо вектори $\hat{y}_k^{(i)}$, щоб отримати прогнозоване вимірювання в момент часу k. Це базується на рівнянні (2.22):

$$\hat{y}_k = \frac{1}{2n}\sum_{i=1}^{2n}\hat{y}_k^{(i)}.$$

Оцінимо коваріацію прогнозованного вимірювання, як показано у рівнянні (2.27). Однак слід додати R_k до кінця рівняння, щоб врахувати шум вимірювання:

$$P_y = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{y}_k^{(i)} - \hat{y}_k\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T + R_k. \qquad (2.30)$$

Оцінимо перехресну коваріацію між $\hat{x}_k^-$ та $\hat{y}_k$ на основі рівняння (2.27)

$$P_{xy} = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{x}_k^{(i)} - \hat{x}_k^-\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T.$$

Оновлення вимірювань оцінки стану можна виконати, використовуючи звичайні рівняння фільтра Калмана:

$$K_k = P_{xy}P_y^{-1},$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\left(y_k - \hat{y}_k\right),$$

$$P_k^+ = P_k^- + K_k P_y K_k^T .$$

Алгоритм, наведений вище, припускає, що рівняння процесу та вимірювання є лінійними по відношенню до шуму, як показано в рівнянні (2.28). Загалом, рівняння процесу та вимірювання можуть мати шум, який входить в ці рівняння нелінійно. А саме

$$x_{k+1} = f\left(x_k, u_k, w_k, t_k\right),$$

$$y_k = h\left(x_k, v_k, t_k\right).$$

У цьому випадку представлений вище алгоритм сигма-точкового фільтра Калмана не є суворим, оскільки він розглядає шум як доповнення, як це видно з рівнянь (2.29) та (2.30). Для вирішення цієї ситуації ми можемо збільшити шум до вектора стану

$$x_k^{(a)} = \begin{bmatrix} x_k \\ w_k \\ v_k \end{bmatrix}.$$

Тоді ми можемо використовувати сигма-точковий фільтр Калмана для оцінки доповненого стану $x_k^{(a)}$. Сигма-точковий Фільтр Калмана ініціалізується як

$$x_0^{a+} = \begin{bmatrix} E\left(x_0\right) \\ 0 \\ 0 \end{bmatrix},$$

$$P_0^{a+} = \begin{bmatrix} E\left[\left(x_0 - \hat{x}_0\right)\left(x_0 - \hat{x}_0\right)^T\right] & 0 & 0 \\ 0 & Q_0 & 0 \\ 0 & 0 & R_0 \end{bmatrix}.$$

Потім ми використовуємо алгоритм сигма-точкового ФК, представлений вище, за винятком того, що ми оцінюємо розширене математичне очікування та коваріацію, тому ми вилучаємо Q_{k-1} та R_k з рівнянь (2.29) та (2.30).

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір мови програмування та необхідних бібліотек

Практичну частину даної дипломної роботи розроблено на мові програмування Python.

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий набір корисних функцій.

Python підтримує структурне, узагальнене, об'єктно-орієнтоване, функціональне і аспектно-орієнтоване програмування. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках, включаючи пропрієтарні. Є реалізація інтерпретатора для JVM з можливістю компіляції, інші незалежні реалізації. Проект PyPy використовує JIT-компіляцію, яка значно збільшує швидкість виконання Python-програм.

Python – це мова програмування, яка активно розвивається, нові версії з додаванням або зміною мовних властивостей виходять приблизно раз в два з половиною роки. Язик не піддавався офіційній стандартизації, роль стандарту де-факто виконує CPython, що розробляється під контролем автора мови. На даний момент Python займає друге місце в рейтингу TIOBE з показником 12,12%. Аналітики відзначають, що це найвищий бал Python за весь час його присутності в рейтингу.

3.2 Опис програми

В результаті отримано наступні елементи програми:

– unscented_kalman_filter.py – реалізує клас UKF;

– measurements.py – реалізує кілька технологічних і вимірювальних моделей. Цей скрипт буде регулярно оновлюватися.

– ukf_turn-rate_velocity.py – ілюструє відстеження дорожнього транспортного засобу з використанням UKF і моделі постійної швидкості повороту.

Вимоги до класу UKF:

1. Клас UKF повинен бути самодостатнім тобто він повинен включати всі функціональні блоки в межах одного класу, включаючи створення і оновлення сигма-точки.

2. Клас UKF повинен володіти гнучкістю, дозволяючи приймати будь-які форми процесів і функцій вимірювання в якості вхідних даних.

3. Клас UKF повинен мати можливість справлятися як з адитивним, так і з неадитивну шумом процесу. Зокрема, клас UKF повинен мати можливість реалізації збільшення сигма-точок, що часто потрібно для неадитивного шуму процесу.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

В програмі ми реалізуємо сигма-точковий фільтр Калмана для нелінійної поведінки об’єкта. В нашому випадку для автомобіля.

Крок прогнозу сигма-точкового ФК обчислює апріорне значення, використовуючи модель процесу $f(\ )$. Яка є нелінійною, тому ми генеруємо сигма точки χ і відповідні їм ваги W^m, W^c відповідно до деякої функції:

$$\chi = sigma - function(x, P),$$

$$W^m, W^c = weight - function(n, parameters).$$

Кожну точку сигми ми пропускаємо через $f(\chi, \triangle t)$. Це проектує сигма-точки вперед у часі відповідно до моделі процесу, формуючи новий пріор, який є набором сигма-точок, які ми називаємо γ:

$$\gamma = f(\chi, \triangle t).$$

Ми обчислюємо середнє значення та коваріацію пріору, використовуючи перетворення без запаху в перетворених сигма-точках.

$$\overline{x}, \overline{P} = UT(\gamma, w_m, w_c, Q).$$

Маємо рівняння для сигма-точкового перетворення

$$\overline{x} = \sum_{i=0}^{2n} w_i^c \left(\gamma_i - \overline{x}\right)\left(\gamma_i - \overline{x}\right)^T + Q.$$

Крок оновлення відбувається наступним чином.

Фільтри Калмана виконують оновлення у вимірювальному просторі. Таким чином, ми перетворюємо сигма-точки пріору у вимірювання, використовуючи визначену нами функцію вимірювання $h(x)$, яку ми визначаємо як

$$Z = h(\gamma).$$

Ми обчислюємо середнє значення та коваріацію цих точок, використовуючи перетворення без запаху. Нижній індекс z означає, що це середнє значення та коваріація вимірювальних сигма-точок.

$$\mu_z, P_z = UT\left(Z, w_m, w_c, R\right),$$

$$\mu_z = \sum_{i=0}^{2n} w_i^m Z_i,$$

$$P_z = \sum_{i=0}^{2n} w_i^c \left(Z_i - \mu_z\right)\left(Z_i - \mu_z\right)^T + R.$$

Далі обчислюємо залишковий коефіцієнт і коефіцієнт посилення Калмана. Залишок вимірювання z легко обчислити:

$$y = z - \mu_z.$$

Для обчислення коефіцієнта посилення Калмана спочатку обчислюємо перехресну коваріацію стану та вимірювань, яка визначається як:

$$P_{xz} = \sum_{i=0}^{2n} w_i^c \left(\gamma_i - \overline{x}\right)\left(Z_i - \mu_z\right)^T.$$

І тоді коефіцієнт посилення Калмана визначаємо як

$$K = P_{xz} P_z^{-1}.$$

Обчисляємо нову оцінку стану, використовуючи залишок і посилення Калмана

$$x = \bar{x} + Ky.$$

І нову коваріацію обчислюємо як

$$P = \bar{P} - KP_z K^T.$$

Результати програми:

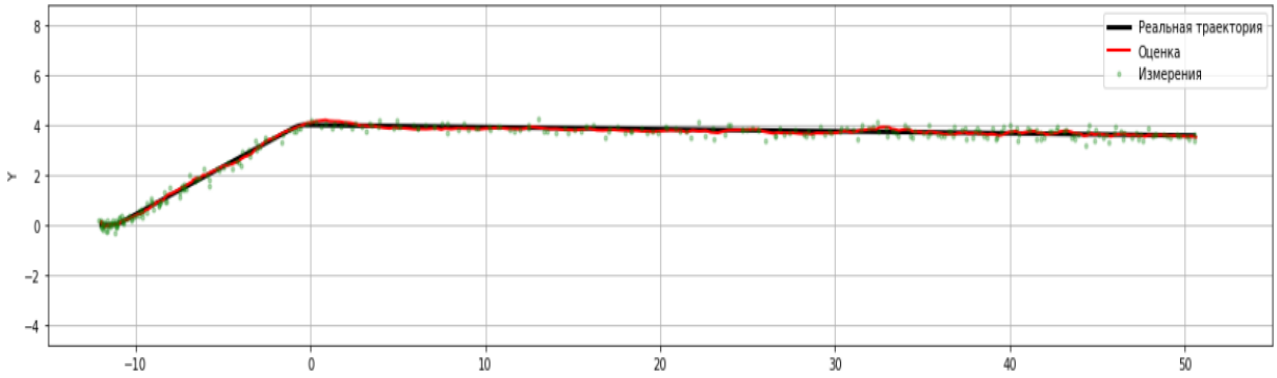

Рисунок 4.1 – Передбачена траєкторія руху автомобіля

На рисунку 4.1 червоною лінією показано значення оцінок, тобто передбачена фільтром траєкторія руху машини. Зелені крапки означають зашумлені показники вимірювання, які ми отримали з датчиків. Чорна лінія – це істинна траєкторія руху автомобіля.

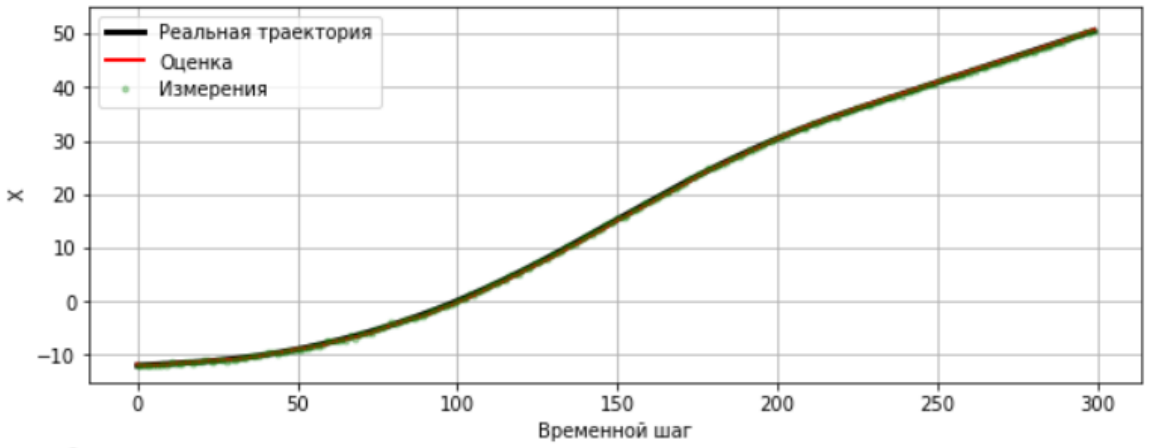

Рисунок 4.2 – Передбачена траєкторія руху автомобіля по осі X

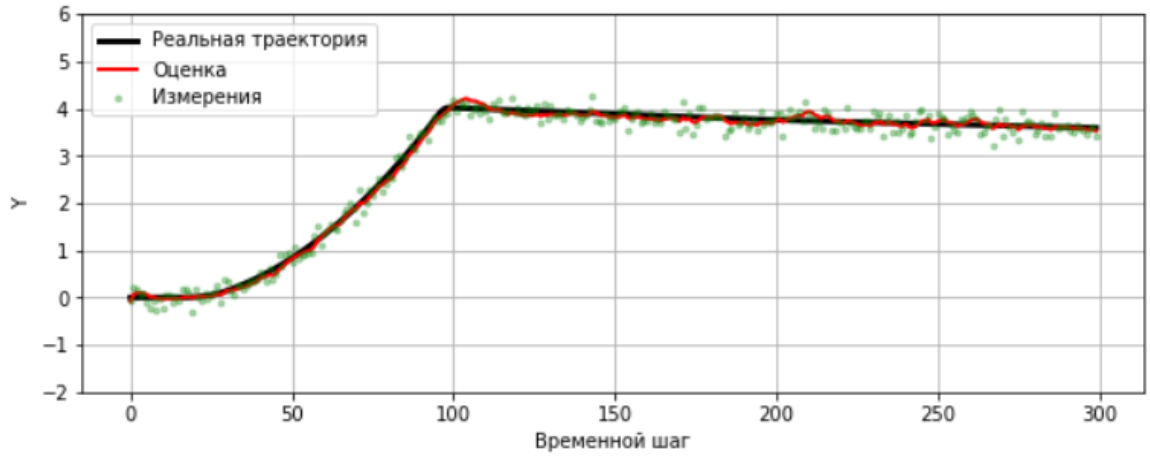

Рисунок 4.3 – Передбачена траєкторія руху автомобіля по осі Y

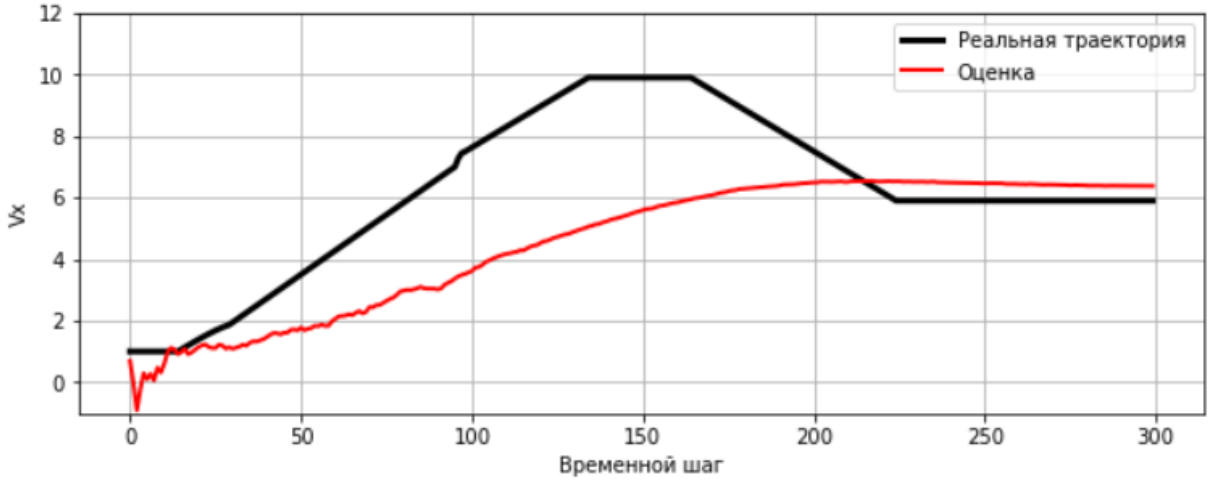

Рисунок 4.4 – Прогнозована і вимірювальна швидкість автомобіля по осі X

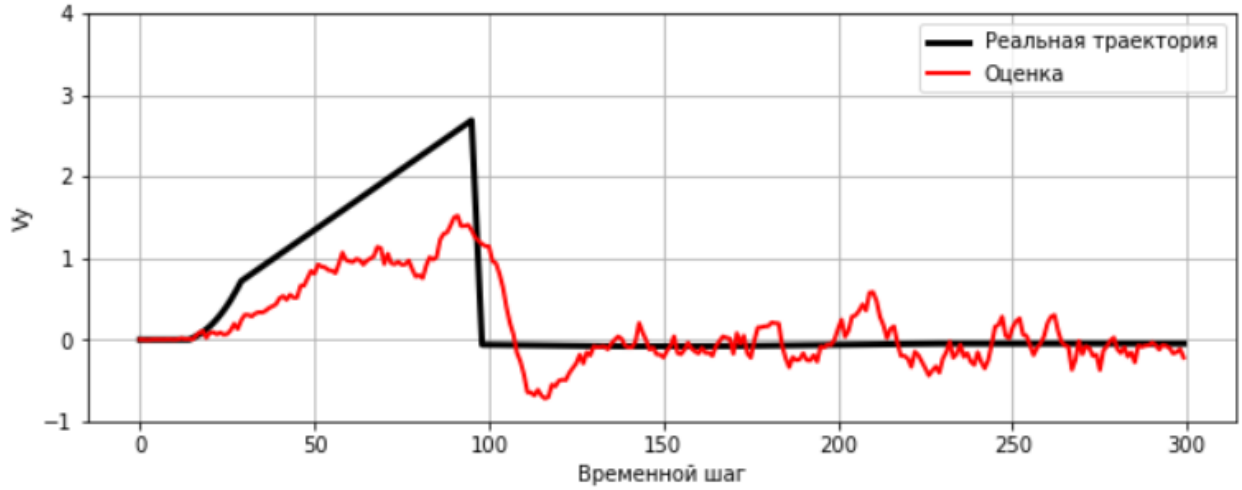

Рисунок 4.5 – Прогнозована і вимірювальна швидкість автомобіля по осі Y

5 АНАЛІЗ МОЖЛИВИХ ЗАСТОСУВАНЬ

В 21 столітті проблема прогнозування шляху є дуже популярною та постійно набуває обертів, тому, що питання навігації і відстеження переміщення об’єктів стають все більш і більш актуальними з кожним днем. У даній роботі було розглянуто оптимальний метод вирішення.

Можна навести такі приклади її застосування:

– система автоматизованої допомоги водію. Тобто, таким чином, можливо запобігати аварійні ситуації на дорозі, за допомогою розпізнавання об’єктів. в разі, якщо людині стало зле, або вона заснула за кермом;

– прогнозування місця розташування об’єктів може підвищити рівень безпеки та може сприяти більшій оперативності спеціальних підрозділів, тому що можливо передбачити місце подій;

– допомога по господарству, зараз набирає обертів популярний пристрій робот-пилосос;

– батьківський контроль – смарт годинник для дітей. З ним батьки завжди будуть знати, де знаходиться їх дитина і що поруч з нею відбувається.

Це лише мала частина того, де можна застосувати дану розробку. На мою думку, технології невпинно ростуть і даний напрямок буде досліджуватись та розвиватись все з більшою силою.

ВИСНОВКИ

Під час виконання атестаційної роботи була здійснена розробка програми визначення траєкторії руху об’єкту з використанням сигма-точкового фільтра Калмана. Цей фільтр дає змогу відстежувати об’єкти, рух яких описаний нелінійною моделлю. Дана програма може бути використана в системах безпеки, навігації машин, стеження за підозрілою активністю.

За останній час зроблений значний прогрес в області трекінгу об’єктів. Приводячи до значної кількості методів та технологій для визначення та відстеження об’єктів. Головним викликом для дослідників є розробка алгоритму, який би забезпечував результат незалежно від того, наскільки зашумлені дані.

Було розроблено алгоритм роботи та архітектуру програми. Для розробки продукту було обрано мову програмування – Python.

На закінчення хочеться додати, що розвиток і тенденції в області трекінгу знаходять застосування в багатьох сферах. Визначення і відстеження об’єктів дає змогу аналізувати об’єкти для подальшої роботи з цими даними в залежності від використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Антонов А. В. Системный анализ. Москва : Высшая школа, 2004. 454 с.

2. Волкова В. Н., Денисов А. А. Основы теории систем и системного анализа. СПб : СПбГТУ, 1997. 510 с.

3. Саати Т. Принятие решений. Метод анализа иерархий. Москва : Радио и связь, 1993. 278 с.

4. Human trajectory prediction in crowded spaces / A. Alahi, K. Goel, V. Ramanathan, A. Robicquet // In: Computer Vision and Pattern Recognition. 2016. №6. С. 961–971.

5. Dalal, N., Triggs, B. Histograms of oriented gradients for human detection // In: Computer Vision and Pattern Recognition. 2005. № 7 (45). С. 886–893.

6. Activity forecasting / Kitani K.M., Ziebart B.D., Bagnell J.A., Hebert M. // In: European Conference on Computer Vision. 2012. № 4 (37). С. 201–214.

7. Human trajectory prediction in crowded spaces / Alahi A., Goel K., Ramanathan V., Robicquet A. // In: Computer Vision and Pattern Recogniton. 2016. №6. С. 961–971.

8. Einicke G., Smoothing, Filtering and Prediction- Estimating The Past, Present and Future. Croatia : InTech, 2012. 363 с

9. Simon D., Optimal State Estimation Kalman, H Infinity, and nonlinear approaches. New Jersey : John Wiley & Sons, 2006. 550 с.

10. Rogger R., Kalman and Bayesian Filters in Python. Toronto : Longman, 2018. 504 с.