

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів

Кафедра систем управління літальних апаратів

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

магістра

(освітньо-кваліфікаційний рівень)

на тему Розробка і дослідження системи управління транспортним роботом при русі у середовищі з перешкодами

ХАІ.301.361.24О.272.00183012 ПЗ

Виконав: студент 2 курсу, групи 361

Галузь знань 27 «Транспорт»

Спеціальність

272 Авіаційний транспорт

Освітня програма

Інтелектуальні транспортні системи

Литовченко Олександр Юрійович

(прізвище та ініціали студента)

Керівник Немшилов Ю. О.

(прізвище та ініціали)

Рецензент Жученко О. С.

(прізвище та ініціали)

Міністерство освіти і науки України
 Національний аерокосмічний університет ім. М. Є. Жуковського
 «Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
(повне найменування)
 Кафедра систем управління літальних апаратів
(повне найменування)
 Рівень вищої освіти другий (магістерський)
 Галузь знань 27 «Транспорт»
(код та найменування)
 Спеціальність 272 Авіаційний транспорт
(код та найменування)
 Освітня програма Інтелектуальні транспортні системи
(найменування)

ЗАТВЕРДЖУЮ
 Завідувач кафедри,
 К.Т.Н., с.н.с
 _____ Костянтин ДЕРГАЧОВ
(підпис) (ім'я та прізвище)
 «__» _____ 2024 р.

ЗАВДАННЯ
 НА КВАЛІФІКАЦІЙНУ РОБОТУ

Литовченка Олександра Юрійовича

(прізвище, ім'я та по батькові)

- Тема кваліфікаційної роботи «Розробка і дослідження системи управління транспортним роботом при русі у середовищі з перешкодами»
 керівник кваліфікаційної роботи
канд. техн. наук, доцент Немшилов Юрій Олександрович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
 затверджені наказом Університету № 1968-уч від «06» листопада 2023 року.
- Термін подання здобувачем кваліфікаційної роботи «08» січня 2024 р.
- Вихідні дані до роботи Математична модель руху БПЛА, вимоги до функціонування системи та проведення аерофотозйомки
- Зміст пояснювальної записки (перелік завдань, які потрібно розв'язати) Вступ, аналіз проблеми, опис об'єкту, математичний опис об'єкту, синтез середовища для моделювання, конструкторська частина, розробка дослідницького процесу, експериментальна частина, економічна частина, заключення, список використаних джерел
- Перелік графічного матеріалу 10 плакатів формату А1 у вигляді слайдів

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стан проблеми	к.т.н., доц. Немшилов Ю. О.	12.09.23	10.01.24
Аналіз даної задачі	к.т.н., доц. Немшилов Ю. О.	12.09.23	10.01.24
Конструктор. частина	к.т.н., доц. Немшилов Ю. О.	12.09.23	10.01.24
Дослідницька частина	к.т.н., доц. Немшилов Ю. О.	12.09.23	10.01.24
Експ.-практ. частина	к.т.н., доц. Немшилов Ю. О.	12.09.23	10.01.24
Економічне обґрунтування	к.т.н., доц. Немшилов Ю. О.	12.09.23	10.01.24

Нормоконтроль _____ Немшилов Ю.О. « 16 » 01 2024 р.
(підпис) (ім'я та прізвище)

7. Дата видачі завдання « 12 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Формулювання теми проекту.	09.09.2023	
2	Розробка технічного завдання	12.09.2023	
3	Огляд стану проблеми і патентний пошук. Математичний опис системи управління.	05.10.2023	
4	Аналіз предметної області	21.10.2023	
5	Конструкторська частина роботи. Дослідницька частина роботи. Економічне обґрунтування розробки	21.12.2023	
6	Експериментально-практична частина	27.12.2023	
7	Оформлення розрахунково-пояснювальної записки і графічного матеріалу. Попередній захист роботи	18.01.2024	
8	Рецензування проекту. Захист проекту в ЕК		

Здобувач

Олександр ЛИТОВЧЕНКО

(підпис)

(ім'я та прізвище)

Керівник кваліфікаційної роботи

Юрій НЕМШИЛОВ

(підпис)

(ім'я та прізвище)

Міністерство освіти і науки України
 Національний аерокосмічний університет ім. М. Є. Жуковського
 «Харківський авіаційний інститут»

Кафедра систем управління літальних апаратів (№301)

«ЗАТВЕРДЖУЮ»
 Завідуючий кафедрою 301
 к.т.н., с.н.с., доцент
Костянтин ДЕРГАЧОВ
 «__» _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ
 на дипломне проектування
Литовченка Олександра Юрійовича

- 1 Тема роботи: Розробка і дослідження системи управління транспортним роботом при русі у середовищі з перешкодами
 затверджена наказом по університету від «06» листопада 2023 р. № 1968-уч .
- 2 Строк здачі студентом закінченої роботи « 08 » січня 2024 р.
- 3 Область застосування розробки: робота з нестійкими космічними об'єктами
- 4 Початкові дані для розроблювальної системи
 - 4.1 Призначення і мета створення системи: створення ефективної системи визначення польоту для проведення аерофотозйомки
 - 4.2 Загальні відомості завдання на дипломний проект, програмні аналоги для відображення керованих рухомих графічних об'єктів
- 5 Технічні вимоги до каналів системи управління
 - 5.1 Питання, що підлягають розробці: формування математичної моделі, вибір виконавчого та вимірювальних пристроїв, синтез напівнатурного моделювання.
 - 5.2 Режим роботи системи (безперервний, циклічний, одноразової дії): автоматичний
- 6 Умови експлуатації системи:
 - 6.1 Кліматичні вимоги до експлуатації (температура середовища, у якій буде працювати система управління, її вологість, вміст хімічно активних компонентів і т.ін.): температура середовища ± 20 градусів за Цельсієм.
 - 6.2 Механічні вимоги (вібрація, тряска, можливі перекося, удари, нахили і т.ін.): можливі вібрація, удари, нахили.
 - 6.3 Наявність перешкод (електричні наведення радіоперешкоди, магнітні впливи): можна знехтувати
- 7 Додаткові функції, реалізовані системою (сигналізація про несправності, реєстрація необхідної інформації, самоконтроль самої системи і т.ін.): Дана система не має додаткових функцій.
- 8 Обсяг виконуваних розроблювачем робіт
 - 8.1 Етапи проведення роботи: I етап: Оцінка стану проблеми.
II етап: Вербальний і математичний опис об'єкту
III етап: Синтез пристрою
 - 8.2 Обсяг розробки по кожному етапу: _____

- 1) стан проблеми і постановка задачі проектування (10 ст) _____
- 2) опис задачі та математичний опис (40ст) _____
- 3) конструкторська частина (12 ст) _____
- 4) розробка дослідницького процесу (12ст) _____
- 5) економічна частина (10 ст) _____
9. Параметри обладнання системи
- 9.1. Габарити розміри визначаються у процесі проектування _____
- 9.2. Маса визначається в процесі проектування _____
- 9.3. Вимоги до конструктивного виконання і розміщенню визначаються в процесі проектування _____
- 9.4. Інші вимоги не передбачаються _____
10. Вимоги безпеки визначаються в процесі проектування _____
11. Дослідницька частина
- 11.1. Розробка дослідницького процесу розробка процедури дослідження контролера з периферійним обладнанням _____
- 11.2. Умови і вимоги процедури перевірки апаратних засобів _____
- 11.3. Очікуваний результат створення повністю працездатної та повнофункціональної моделі _____
12. Економічна частина
- 12.1. Розробити (розрахувати, отримати): трудомісткість виконання роботи, розрахунок витрат на виконання роботи, договірна ціна роботи. _____
- 12.2. Умови і вимоги єдиничне виробництво, розрахунок проводиться за статтями калькуляції _____
- 12.3. Очікуваний результат собівартість виробу грн. з урахуванням усіх витрат _____
13. Перелік графічних матеріалів та їх формат 10 слайдів формату А4:
- 1) опис об'єкту автоматичного управління, математична модель об'єкту, результат дослідження об'єкту. _____
- 2) середовище розробки. _____
- 3) дослідницька частина, економічна частина. _____
- 4) висновок. _____
14. Мова підготовки пояснювальної записки (захисту) українська _____

Керівник роботи
к.т.н., доц. Немшилов Ю. О.
(П.І.Б.)

«12» вересня 2023 р.

Прийняв до виконання
Литовченко О.Ю.
(П.І.Б. студента)

«12» вересня 2023 р.

Погоджено з питань:

конструкції

_____ к.т.н., доц. Немшилов Ю. О.
_____ (П.І.Б.)

_____ «12» вересня 2023 р.

дослідницької частини

_____ к.т.н., доц. Немшилов Ю. О.
_____ (П.І.Б.)

_____ «12» вересня 2023 р.

економіки

_____ к.т.н., доц. Немшилов Ю. О.
_____ (П.І.Б.)

_____ «12» вересня 2023 р.

РЕФЕРАТ

Даний документ являє собою пояснювальну записку до дипломного проекту: 109 с., 75 рис., 10 табл., 1 додаток, 26 джерел.

Актуальність теми обумовлено тим, що системи управління рухом для роботів надають їм можливість автономно аналізувати оточуюче середовище і приймати рішення стосовно напрямку переміщення. Ці системи забезпечують самостійний аналіз навколишнього середовища та прийняття відповідних рішень щодо руху.

Мета проведення дослідження полягає в розробці моделі мобільного робота для лабораторних експериментів, обладнаного системою управління рухом за допомогою сучасних методів і підходів для розв'язання подібних завдань. Для досягнення цієї мети важливо визначити і реалізувати алгоритм орієнтації мобільного робота у просторі за допомогою візуальної інформації, а також гарантувати отримання параметрів просторового положення об'єкта.

Завдання передбачають аналіз поточного стану наукових досліджень, аналіз та створення системи управління рухом транспортного робота, вибір елементної бази для розробки та створення моделі мобільного робота, проведення напівнатурних і натурних випробувань розробленої системи, а також розрахунок собівартості запропонованої схеми.

НЕЧІТКИЙ РЕГУЛЯТОР, НАВІГАЦІЯ, СИСТЕМА УПРАВЛІННЯ,
КОЛІСНИЙ РОБОТ, ОПТИЧНИЙ ПОТІК, ТЕХНІЧНИЙ ЗІР

ЗМІСТ

СПИСОК СКОРОЧЕНЬ	10
ВСТУП.....	11
1 СТАН ПРОБЛЕМИ ПРОЕКТУВАННЯ.....	12
1.1 Дослідження стану проблеми	12
1.2 Аналіз технічного завдання	13
1.3 Аналітичний огляд науково-технічної літератури та патентів	14
Висновки до розділу 1	Error! Bookmark not defined.
2 АНАЛІЗ І СИНТЕЗ системи об'єктивного контролю руху транспортного робота	18
2.1 Мета, задачі, структура та вибір шляху розв'язку проблеми	18
2.2 Система комп'ютерного зору	20
2.3 Побудова алгоритму стереобачення для визначення відстані.....	29
2.4 Алгоритм пошуку подібності SURF	33
2.5 Визначення оптимального шляху. Алгоритм Дейкстри	38
2.6 Основні задачі руху.....	40
2.7 Властивості моделювання та управління..	Error! Bookmark not defined.
2.8 Цільова машина.....	Error! Bookmark not defined.
2.9 Відстеження траєкторії	48
2.10 Математична модель колісного робота ...	Error! Bookmark not defined.
2.11 Висновки до розділу	68
3 КОНСТРУКТОРСЬКА ЧАСТИНА	69
3.1 Вибір версії робота AlphaBot.....	69
3.2 Вибір плати управління	73
3.3 Опис плати управління мобільним роботом.....	76
3.4 Опис плати L298n.....	77
3.5 Висновки до розділу 3	78
4 ДОСЛІДНИЦЬКА ЧАСТИНА.....	79
4.1 Модельні випробування.....	79
4.2 Стабілізація стану.....	Error! Bookmark not defined.
4.3 Висновки до розділу 4	93
5 ЕКСПЕРИМЕНТАЛЬНО-ПРАКТИЧНА ЧАСТИНА	94
5.1 Завдання і цілі виконання експериментальної розробки	94
5.2 Опис лабораторної установки	Error! Bookmark not defined.

5.3 Аналіз результатів проведення експериментів і впровадження розробки.....	Error! Bookmark not defined.
5.4 Висновки до розділу 5	Error! Bookmark not defined.
6 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ	99
6.1 Мета економічної частини.....	99
6.2 Короткий опис апарату.....	99
6.3 Компанія виробника	Error! Bookmark not defined.
6.4 Сегментування ринку	Error! Bookmark not defined.
6.5 Розрахунок собівартості і ціни розробки безпілотного літального апарату	101
6.6.Висновок до розділу 6	106
ВИСНОВКИ	107
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	108
ДОДАТОК А	Error! Bookmark not defined.

СПИСОК СКОРОЧЕНЬ

EEPROM	– пам'ять з електричним стиранням та програмуванням;
GPIO	– вхід/вихід загального призначення;
I ² C	– стандарт протоколу послідовного передавання даних;
ICSP	– програмування в серійному режимі в колі;
IR	– інфрачервоний;
SPI	– серійний периферійний інтерфейс;
SRAM	– статична оперативна пам'ять з довільним доступом;
SURF	– прискорені стійкі особливості;
UART	– універсальний асинхронний приймач/передавач;
USB	– універсальна послідовна шина.
АЦП	– аналогово-цифровий перетворювач;
ДР	– датчик руху;
ДЗ	– датчик зображень;
КБ	– коригуючий блок;
КМР	– колісний мобільний робот;
МК	– мікроконтролер;
ОАУ	– об'єкт автоматичного управління;
ОЗП	– оперативна пам'ять;
ОП	– оптичний потік;
ОС	– оптична система;
ПП	– підсилювач потужності;
ПС	– пристрій сполучення;
С	– сервопривід;
САУ	– система автоматичного управління;
СК	– система координат;
СКЗ	– система комп'ютерного зору;
СУ	– система управління;
ШІМ	– широтно-імпульсна модуляція.

ВСТУП

Прогрес у сучасній мікропроцесорній техніці забезпечив придатні умови для розвитку компактних мобільних роботів, управління якими може здійснюватися дистанційно або автоматично. На даний момент було проведено значну кількість досліджень та випробувань, що призвели до розробки нових алгоритмів управління, що забезпечують виконання різноманітних нетривіальних операцій, таких як уточнення карти місцевості, планування траєкторії обходу перешкод, проникнення в важкодоступні зони, рух по складних криволінійних траєкторіях із забезпеченням потрібної орієнтації елементів робота в просторі, за допомогою мобільних роботів. У зв'язку із цим збільшується потреба в створенні зразків роботизованих комплексів дослідного й дослідно-промислового застосування.

Проблеми динаміки, навігації й управління, які існують в робототехнічних системах, вимагають створення інноваційних підходів, які ґрунтуються на доцільному використанні комп'ютерних методів. До таких методів відносяться численні експерименти зі складними математичними моделями, системи аналітичних обчислень, візуальне моделювання, застосування нечіткої логіки та технічного зору, а також комп'ютерна анімація тощо. Ключовою тенденцією сучасної науки і техніки є складні технічні системи. Складні технічні системи - це загальна назва нового класу систем, що виник у процесі розвитку теоретичного знання та інформаційних технологій взагалі. До цього дослідники могли аналізувати тільки ті явища, що точно описані, а також процеси лише з малою кількістю змінних.

Предмет дослідження - колесна мобільна роботизована платформа. Об'єктом дослідження є методи та алгоритми забезпечення руху роботизованої платформи на підставі візуальної інформації. Метою дослідження є розробка моделі мобільного робота для лабораторних випробувань, оснащеного системою об'єктивного контролювання руху за допомогою сучасних методів й підходів до вирішення задач такого роду. Для досягнення поставленої мети необхідно обрати й реалізувати алгоритми орієнтації мобільного робота у просторі із допомогою візуальної інформації. Також треба забезпечити отримання параметрів просторового положення об'єкта, вирішення задачі управління рухом об'єкта у замкнутому просторі, переміщення до певного місця та повернення в початкове положення.

1 СТАН ПРОБЛЕМИ ПРОЕКТУВАННЯ

1.1 Загальна характеристика проблеми

Для мобільних роботів питання реалізації програмних і апаратних засобів для системи технічного зору, які забезпечує оцінювання координат та обрахування маршрутів та задач управління, стають практично важливими. До того ж, розвиток систем управління, що базуються на штучному інтелекті, визнається одним із найперспективніших напрямків. До цієї категорії входять і системи, що здатні до об'єктивного контролю динамічних об'єктів.

Розробка системи об'єктивного контролю руху мобільного робота представляє собою складні та нелінійні задачі управління. Для втілення таких систем автоматичного управління, які працюють з об'єктами, які погано формалізовані, часто використовують пристрої та алгоритми керування і навігації, ґрунтовані на методах нечіткої логіки (fuzzy logic). Ці методи відрізняються від класичних, використовуючи "людський" підхід та рішення задач управління.[1]

Теорія нечітких множин, яку запропонував американський математик Лотфі Заде більше ніж 50 років тому, дозволяє описувати якісні, неточні поняття й наші знання про навколишнє середовище, а також оперувати такими знаннями для отримання нової інформації. Методи побудови інформаційних моделей, які базуються на цій теорії, розширюють традиційні сфери застосування комп'ютерів, утворюючи самостійний напрямок науково-прикладних досліджень - а саме нечітке моделювання.

Інтелектуальна система автоматичного управління (САУ) має вміння навчатися, адаптуватися та накопичувати знання про об'єкт управління (ОУ). При створенні інтелектуальної САУ, система майже завжди містить базовий набір знань, отриманий від фахівців галузі та експертів, що представлені у відповідності із обраною мовою подання знань та методом опису їх нечіткості. Оскільки знання або досвід людини мають вербальний характер та не завжди є чіткими, найбільш перспективними є інтелектуальні САУ. САУ використовуються задля представлення знань людини про властивості й принципи управління об'єктом.[2]

Доцільність конкретного підходу до представлення нечітких знань в інтелектуальних САУ обумовлена тим, що цей математичний апарат працює із лексичними категоріями оцінок, сприйняття й способами міркування людини. Отже, з нечіткими лінгвістичними термінами, і відповідно до аксіоматики управління складними об'єктами, які слабо структуровані, всю інформацію про об'єкт й методи управління ним можна висловлювати за допомогою звичайної мови. Крім того, такий підхід до представлення нечітких знань значно полегшує початкове «навчання» інтелектуальної системи, що розробляється підбіраною групою експертів. Це можливо завдяки тому, що апарат нечітких

множин, який оперує лінгвістичними змінними, дозволяє найбільш точно втілювати машинну інтерпретацію знань експертів [3].

Передумовою застосування нечітких систем управління в широкому розумінні є, з одного боку, наявність невизначеності, пов'язаної як із відсутністю інформації, так і зі складністю самої системи й неможливістю або недоцільністю опису традиційними методами. З іншого боку, це передбачає наявність об'єкта, який потребує управління, впливів на нього, збурень тощо, а також наявність інформації якісного характеру. Однак для застосування нечітких систем управління необхідна наявність невизначеності та об'єкта, що потребує управління. Паралельно з завданням управління, постає завдання орієнтації та навігації об'єкта автоматичного управління, зокрема, використовуючи алгоритми технічного зору.

1.2 Аналіз технічного завдання

У рамках дипломного проекту магістра необхідно реалізувати систему, яка володіє навичками об'єктивного контролю руху транспортного робота.

На сьогодні значне розповсюдження серед систем управління мобільними платформами отримали системи, обладнані вимірювальними приладами інерціального характеру, такими як акселерометри, датчики кутових швидкостей, магнітометри та інші, а також системи, що використовують супутникові системи навігації.

Але, незважаючи на переваги використання таких систем, існують недоліки, пов'язані з втратою ефективності в обмеженому просторі, наприклад, всередині приміщення, через відсутність прямого видимості супутників. Таким чином, необхідно враховувати, що система може працювати неефективно без корекції сигналу протягом тривалого періоду через проблеми інтегрування та чутливість до електромагнітного поля, вібрацій, ударів тощо.

Альтернативою є системи, що базуються на використанні алгоритмів технічного зору (ТЗ), та здатні працювати в закритому просторі з постійною точністю протягом всього періоду [2]. При розробці системи для мобільного робота вважається, що система орієнтації на основі технічного зору є найбільш за все придатною, оскільки вона призначена для тестових випробувань та роботи в приміщенні.

Оскільки оптичні системи (ОС) не накопичують помилок протягом тривалого періоду, їх можна використовувати як альтернативне джерело вимірювання динамічних параметрів об'єкта, зокрема параметрів руху. Показники

цієї вимірювальної системи можна використовувати як сигнал для корекції параметрів вимірювання.

Візуальна інформація, надана ОС вимірювання, дозволяє визначати просторові відносини. Завдяки цьому, маючи ряд послідовних кадрів або відео та інформацію про їх час формування, можна визначати параметри динаміки.

У даній системі технічного зору використовується непересувна камера, що контролює робочий простір мобільного робота. За допомогою відео-інформації, що подається камерою, формується план та визначаються вільні зони простору, які можна використовувати для переміщення робота. Після цього програмно здійснюється пошук попередньо підготовленого об'єкта між початковою (старт) та кінцевою (фініш) точками маршруту. За отриманою інформацією система визначає найкоротший маршрут, який з'єднає крайні точки, при умові, що такий маршрут існує.

Оскільки приблизна модель шуканого об'єкта вже відома, для кожного набору параметрів орієнтації є можливість отримати проектний контур моделі, відомий як еталонний контур, що включає силует об'єкта та його видимі ребра. Для кожної точки зразка обчислюється відстань до найближчого контуру на зображенні. Після цього відбувається пошук параметрів орієнтації, що призводить до мінімізації суми цих відстаней по всьому зразку.

Система взаємодіє з різноманітними об'єктами, розташованими на складній підстильній поверхні. Більше того, об'єкти можуть мати різні форми та будь-яке поєднання кольорів. Виділення об'єктів в таких умовах представляє собою достатньо складний процес. Для цього використовується декілька фільтрів різної розмірності: медіанний, усереднюючий та фільтр Гаусса. Також були використані процедури фільтрації об'єктів за площею та об'єднання декількох об'єктів в один [4].

1.3 Огляд науково-технічної літератури та патентів

Для дійсно високоякісного аналітичного огляду в сучасному світі існує безліч пошукових систем, що значно спрощують роботу дослідника. Раніше пошукових систем для наукових публікацій було багато, найвідоміші це Scirus, GetCITED та Google Scholar. Утім, тепер їхня кількість істотно скоротилася. Головна перевага таких сервісів – зручність, адже вся необхідна інформація накопичується в одному місці від різних джерел й окремих видань. Достатньо створити коректний запит й потім лише відібрати потрібні ресурси [5].

Теорія неякісного управління є однією з гілок теорії інтелектуальних систем. Вона активно використовується в даний час для синтезу неяких регуляторів,

гібридних регуляторів, деяких пошукових систем автоматичної оптимізації, деяких пристроїв оцінювання і фільтрації. Методи, які розвиваються в даній теорії, базуються на математичній теорії нечітких множин, а також побудовану на цій основі нечітку логіку, яка дозволяє оперувати невизначеною або невідомою інформацією, не інтерпретується в кількісних термінах. При управлінні складними процесами, що не мають точного кількісного опису, у порівнянні з традиційними – нечіткі системи мають кращу перешкодо захищеність, швидкодію й точність завдяки детальному опису реального середовища, де вони функціонують.

Обробка деякої інформації та особливо деякий висновок давно використовуються в різних інтелектуальних системах, однак найбільш широке поширення такі системи отримали у галузі управління.

Концептуальна схема такої логіки лежить в основі так званого нечіткого контролера, що використовується в інтелектуальних системах обробки невизначеної інформації, а саме в «інтелектуальних» системах управління. Для «розумних» систем управління на сьогодні характерно інтегрування систем, які оснащені оптичними приймачами, камерами, відео інформація з яких влучно використовується для підбору раціональних законів управління [5].

Постійне підвищення вимог до сучасних систем технічного зору (СКЗ), які функціонують у все більш складних, невизначених та мінливих умовах спостереження, все це робить надзвичайно перспективними дослідження в області розробки систем покращеного і синтезованого спостереження, які в свою чергу працюють із датчиками різної фізичної природи. Вони забезпечують значне підвищення ефективності як у людино-машинних, так і у автоматичних системах управління авіаційних й космічних апаратів, наземних, надводних, також підводних роботів.

Для СКЗ ключовими дослідженнями є ті, що стосуються розробки методів опису двовимірних фігур. Вирішення подібних задач стало одним з вагомих етапів формування описів того, що відбувається навколо, й наближує процес, який протікає в даний час, до вирішення загальної проблеми «розуміння» ситуації [6].

Для висвітлення тематики досліджуваної сфери в першу чергу було проведено огляд літератури з питань використання нечіткої логіки для систем управління мобільними роботами різного застосування, зокрема для СКЗ. Дослідження це відбувалося за допомогою використання пошукових систем мережі Internet.

На сьогодні питання щодо «інтелектуальних» систем керування з інтегрованими СКЗ у основі здобуло значного поширення як в Україні, так і у світі. Дедалі частіше проводяться різноманітні конференції, актуальними проблемами для вирішення яких є системи технічного зору у системах управління.

Оптичний потік представляє собою швидкість візуального руху поперек сітківки в радіанах в секунду, інакше кажучи, це картина руху навколишніх предметів щодо спостерігача. Метод оптичного потоку дає можливість визначати відстань, засновану на відносному русі оточуючих об'єктів. Якщо об'єкт знаходиться далеко, то його рух повільніший порівняно з об'єктом, який перебуває ближче.

Послідовності зображень дозволяють прі цьому спостерігати зміни сцени. Динамічні явища, що спостерігаються через певні часові проміжки, пояснюються рухом об'єктів, спостерігача або обох. Присутні зміни в послідовності зображень можуть використовуватися як характерні ознаки для виявлення рухомих об'єктів чи для визначення їх траєкторій. На випадок руху спостерігача в межах нерухомої сцени, зміни у зображеннях дозволяють визначити траєкторію спостерігача у тривимірному просторі [7].

Зміни інтенсивності або відтінку пікселів зображення виявляються значущими характерними ознаками, корисними для виявлення та розпізнавання об'єктів. У найпростішому випадку нерухома камера спостерігає за постійним фоном. Рухомі на цьому тлі об'єкти призводять до змін інтенсивності / відтінку пікселів в відповідних областях зображень.

Нерухома камера дозволяє отримувати інформацію про рух одного чи декількох об'єктів. Рухомі об'єкти також можна відстежувати протягом певного часу для визначення траєкторії, що, в свою чергу, може надати деяку інформацію про поведінку об'єкта.

Рухома камера породжує зміни на зображеннях через власний рух, навіть якщо тривимірна сцена непорушна. Рухома камера використовується в різних цілях. По-перше, вона дозволяє збирати більше відомостей про оточуюче середовище, ніж доступно з єдиного точкового спостереження. Наприклад, панорамна камера використовується щоб отримати зображення широкого кута (панорамного) виду сцени.

По-друге, надаються дані для визначення відносної глибини об'єктів сцени, оскільки зображення близьких до камери об'єктів змінюються швидше, ніж віддалених. По-третє, ще забезпечуються дані для сприйняття чи вимірювання тривимірної форми близьких об'єктів. При обробці фото- або відеозйомки часто потрібно визначити моменти часу, коли камера переміщувалася або змінювалася її фокусна відстань (масштаб зображення). У даному випадку зміст сцени несуттєвий, важливо визначити саме спосіб спостереження сцени.

У складних задачах аналізу руху по зображеннях використовуються декілька датчиків й сцени із кількома рухомими об'єктами, де буває важко визначити якийсь постійний один фон. Наприклад, це може бути завдання управління колісним роботом в щільному транспортному потоці [8].

Для мобільних роботів пропонуються рішення задач подолання перешкод. Завдяки цьому досягається можливе зменшення часу виявлення цілі мобільним

роботом під час подолання різних нерухомих перешкод шляхом застосування підходу на основі нечіткої логіки.

Оскільки під час переміщення мобільного робота виникає купа проблем та недоліків, особливо це буває при русі в невідомому середовищі, то необхідно використовувати відомі методи побудови й планування переміщення робота в малодослідженому або невідомому середовищі для оптимізації руху. Пропонуються методи, які базуються на поєднанні сучасних нейронних мереж, штучного інтелекту та нечіткої логіки, що вагомо ускладнює процес підготовки до роботи в цілому, тому управління потрібно змінити так, щоб максимально спростити задачу обчислювальної техніки за допомогою введення нової гілки оцінювання вхідних даних за допомогою оптичних систем «зору».

Окрім огляду сучасної літератури згідно теми дипломного проекту, було також виконано патентний пошук за допомогою українського інституту інтелектуальної власності. З аналітичного огляду, що наведено в таблиці 1, необхідно відзначити, що корисні моделі під номерами 2, 3 та 5 відносяться до саме розробки робота, реалізації всіх фізичних та механічних властивостей. Тим часом система управління роботами взагалі не розглядається.

Корисні моделі під номерами 6, 7 та 10, у своєму вирішенні пропонують практичну реалізацію прикладного пристрою управління тією чи іншою системою, де не дозволяється використовувати результати розробок для більш загальних цілей.

Для більш детального розгляду підходять патентні розв'язки під номерами 1, 4, 8 та 9. Однак 8 й 9 вирішують проблеми удосконалення управління великими промисловими апаратами, які вимагають потужного обладнання та габаритних компонентів.

1.4 Постановка задач проектування

З проведеного дослідження випливає, що зараз у всьому світі та особливо на території України активно шукають рішення проблеми забезпечення роботизованих мобільних апаратів системою управління належного військового часу рівня. Оскільки робота подібних роботів є необхідною у безлічі напрямків, таких як робота в важкодоступних місцях, ведення робіт у місцях, небезпечних для життя людини, особлива увага приділяється системам навігації із використанням алгоритмів комп'ютерного зору. Системи що контролюють рух транспортного робота особливо актуальні сьогодні, оскільки вони дозволяють роботів самостійно аналізувати навколишнє оточення і приймати рішення щодо напрямку руху, забезпечуючи повну автономію роботи та не становлять загрози у разі чого життю оператора.

2 АНАЛІЗ І СИНТЕЗ СИСТЕМИ УПРАВЛІННЯ ТРАНСПОРТНОГО РОБОТА

2.1 Мета, задачі, структура та вибір шляху розв'язку проблеми

Мета цього дипломного проекту полягає в вивченні алгоритму формування траєкторії руху об'єкта через середовище, яке наповнене перешкодами, на основі технічного зору. Розроблена система сприятиме вирішенню проблеми орієнтації колісних роботів у просторі з великою кількістю перешкод.

У цьому дослідженні будуть розглянуті наступні завдання, вирішення яких буде здійснюватися за допомогою розглянутого мобільного робота:

– завдання, де необхідний огляд або постійний моніторинг об'єкта без участі людини;

– завдання огляду аварійних об'єктів і попередній контроль стану об'єктів, що постраждали від аварії.

Управління об'єктом такого типу фактично зводиться до управління сервоприводами. У конструкції використовуються InteractiveServoMotor, які вимірюють поворот осі мотора в градусах з точністю $\pm 1^\circ$ або в повних обертах.

Система управління складається з основного блоку, що відповідає за управління сервоприводами відповідно до команд оператора або контуру управління більш високого рівня, а також коригуючого блоку, в якому величини для управління сервоприводами перетворюються відповідно до показань блоку датчиків. До отримання нової команди система циклічно виконує останню отриману команду. Таким чином, при формуванні оператором або контуром управління, сигналу щодо типу руху платформи в основному блоці сервоприводів формуються залежно від ширини керуючих імпульсів у часі. Далі, в блоці корекції, використовуючи свідчення блоку датчиків, ці залежності перетворюються так, щоб робот продовжував рух заданого раніше типу, не втрачаючи його ефективності.[8]

Структура розробленої системи представлена на рис. 2.1.

Алгоритми управління роботом будуються виходячи зі структури системи за наступним принципом - формується керуюча команда x (T_0), яка потім передається на вхід інтерфейсу взаємодії з системою. Після отримання команди формуються величини для управління сервомоторами і величини затримок в їхній роботі, і формується керуючий сигнал f (m). Системою управління формуються відповідно до управління сервомоторами x_i (t), після чого опитуються датчики і коригуючим блоком формуються коефіцієнти для сигналу управління сервомоторами. Після цього керуючий сигнал, перетворений відповідно до показань датчиків, надходить на вхід аналогового перетворювального блоку, який керує рівнем сигналу відповідно до характеристик сервоприводів.

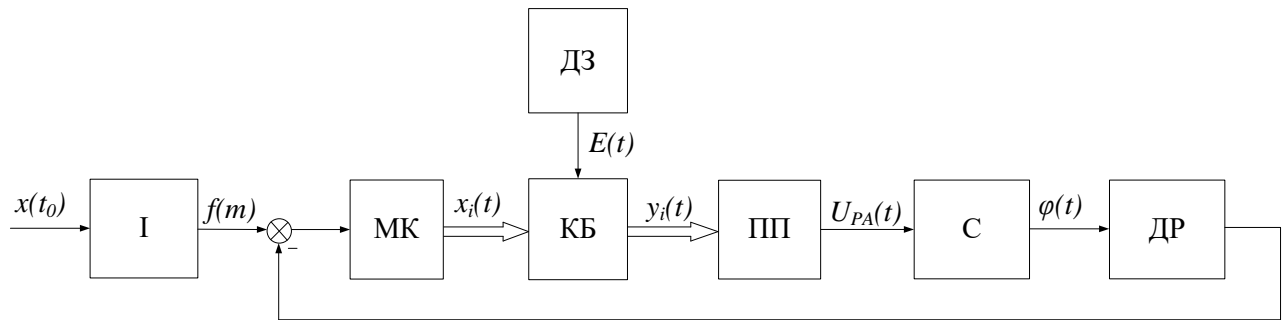


Рисунок 2.1 – Функціональна схема розробленої системи

У системі є такі елементи, а саме :

I – інтерфейс взаємодії із системою;

КБ – корегуючий блок;

ДЗ – датчики зображень;

ДР – датчики руху;

ПП – підсилювач потужності;

МК – мікроконтролер (обчислювальний блок);

С – сервомотори;

$U_{PA}(t)$ – напруга підсилювача потужності;

$E(t)$ – оптичні зображення;

$f(m)$ – керуючий сигнал на вході системи;

T_0 – період дискретизації при передачі даних;

$x(t_0)$ – вхідний сигнал системи у двійковому виді;

$x_i(t)$ – залежності для управління сервоприводами;

$y_i(t)$ – сигнали управління сервоприводами;

$\varphi(t)$ – кут повороту осі сервомотора.

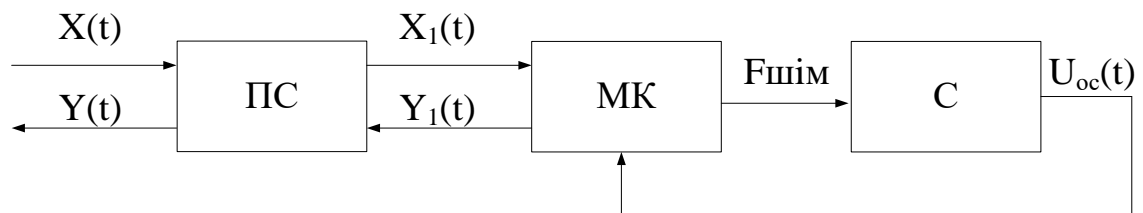


Рисунок 2.2 – Загальна структура пристрою

Структура складається із:

Fшім – широтно модульовані сигнали управління сервомоторів;

$Y_1(t)$ – вихідні сигнали контролера;

$X_1(t)$ – управляючі сигнали контролера;

$Y(t)$ – вихідні сигнали;

$X(t)$ – управляючі сигнали;

МК – мікроконтролер;

ПС – пристрій сполучення;

С – сервомотори;

$U_{oc}(t)$ – напруга зворотного зв'язку сервомоторів.

2.2 Система комп'ютерного зору

Як вже було відзначено, у ролі системи навігації, що виконує функції корегування значень, використовується оптична система з використанням алгоритмів систем комп'ютерного зору (СКЗ).

Технічний зір – сукупність програмно-апаратних засобів, що гарантують отримання зображень, їх обробку та безпосередньо видачу результату, який можна використовувати в подальших наукових технічних рішеннях.[9]

При використанні СКЗ, однією з нагальних задач є розпізнавання з вхідного зображення камери та ідентифікація орієнтованих об'єктів, що вказані. СКЗ використовуються для вирішення задачі пошуку об'єкта орієнтації. У випадку дипломного проектування об'єктом орієнтації є QR-мітка. Для вирішення завдання пошуку QR-мітки за допомогою камери, необхідно спочатку розібратися із специфікою саме даного графічного коду (рис. 2.3).



Рисунок 2.3 – Приклад QR-коду

QR-код – розшифрування означає Швидкий Відгук, що в свою чергу дослідно перекладається як «раптова реакція». Цей вигляд штрих-коду представляє собою двовимірне графічне зображення у формі квадрата, що включає в себе 3 маркера, які розташовані в двох верхніх кутах та лівому нижньому, та інший простір, який заповнений чорними та білими пікселями. Маркером називається область графічного коду, що складається з трьох квадратних контурів, що вкладені один в одного та мають відношення сторін, що відповідає пропорції 7:5:3 (рис. 2.4).

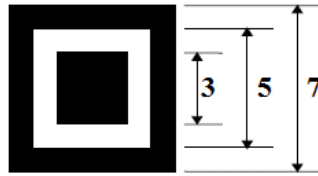


Рисунок 2.4 – Зовнішній вигляд маркерів

Ці дані представляють собою носії інформації, які можуть містити в собі дані у вигляді тексту в половині сторінки формату А4. Крім того, значущою перевагою такого коду є те, що при його пошкодженні чи відсутності якої-небудь частини (пікселя), можливість зчитування інформації з QR-коду не втрачається.

Суть функціонування системи комп'ютерного зору полягає в тому, що при включенні робота, система навігації проводить пошук об'єкта орієнтування для задання напрямку руху робота.[11]

Другим етапом запропонованої системи корекції є визначення відстані до об'єкта для уникнення зіткнення з можливими перешкодами.

Останнім кроком є формування сигналу управління для повороту ходової частини робота відносно предмета орієнтування та виконання переміщення «до» та «від» об'єкту.

Визначення відстані до перешкоди можна здійснити за допомогою системи визначення відстані. Така система будується на основі принципів стерео бачення. Перевага такої системи – це, зокрема, невелика вартість, а при наявності потужного обчислювача, яким можна оснастити роботизовану платформу, можна виконувати обробку зображення.

Для вирішення проблеми визначення місцезнаходження QR-мітки із вхідного відео-потоків необхідно виконати алгоритм, що складається із контурного аналізу отриманого зображення, знаходження ідентифікаційних маркерів та подальшої обробки з ними.

Алгоритм знаходження QR-коду на отриманому зображенні, запропонований в даній роботі, можна розглядати як кілька етапів обробки:

- 1) захоплення зображення з камери;
- 2) перетворення отриманого зображення в градації сірого;
- 3) виявлення контурів та робота з ними:
 - знаходження меж на зображенні;
 - апроксимація знайдених контурів;
 - пошук серед знайдених контурів чотирикутних маркерів;
 - визначення орієнтації маркерів відносно одне одного;
 - визначення чотирьох кутів маркерів;
- 4) знаходження кута коду без маркера;

5) ввідновлення перспективи зображення;

6) графічне виділення детектованого QR-коду в режимі реального часу, вивід орієнтації в консоль та запис до каталогу проекту отриманого під час роботи з зображенням.[12]

У навігаційних та обробних системах часто виникає завдання визначення переміщень об'єктів за допомогою оптичного сенсора, зокрема відеокамери. З використанням оптичної системи обробки послідовності кадрів, необхідно відтворити модель, що відображена на цих кадрах, та відслідковувати зміни, які відбуваються з ними протягом часу.

Якщо потрібно отримати інформацію про зміщення положення об'єкту відносно попереднього за час фіксації кадрів, то можна стверджувати про існування оптичного потоку у методі, який буде використовуватися.

Перед тим як приступити до будь-якої роботи з зображенням, необхідно провести його попередню обробку. Оскільки абсолютне вирішення проблеми за допомогою масочних фільтрів неможливе з ряду причин, необхідно створити фільтр із відповідними властивостями на основі аналогії з одновірним фільтром Калмана.[13]

Найпростішим прикладом подібної фільтрації є рекурсивний фільтр Калмана однорідної стаціонарної гауссівської послідовності, що має експоненціальну кореляційну функцію (2.1).

$$B_x(r) = D_x \cdot \exp(-\alpha \cdot |r|) = D_x \cdot F^{|r|}, \quad (2.1)$$

де: $\alpha > 0$ $F = \exp(-\alpha)$ при – коефіцієнт одношагової кореляції послідовності, визначений параметром α , який має зміст ширини спектра послідовності; D_x – дисперсія послідовності.

У наявності дисперсії на тлі гаусівського білого шуму оптимальний каузальний фільтр реалізується рекурентним алгоритмом, що в режимі фільтрації має наступний вигляд:[14]

$$x_i^* = F \cdot x_{i-1}^* + A \cdot (y_i - F \cdot x_{i-1}^*). \quad (2.2)$$

Імпульсна характеристика описаного фільтра має експоненціальний вигляд (2.3).

$$a_i = A \cdot [(1 - A) \cdot F]^i, i = 0, 1, 2, \dots, \quad (2.3)$$

де A – параметр, що знаходиться в межах $0 < A < 1$, коефіцієнт підсилення фільтра Калмана.

Перший компонент в алгоритмі (2.2) визначає внесок у аналіз сигналу на поточний момент фільтрації, що враховує аналіз сигналу на попередній момент. Другий компонент враховує вплив поточного спостереження та називається "новою інформацією". Коефіцієнт підсилення визначає чутливість фільтра до цієї нової інформації. При високому рівні шуму параметр має значення, близьке до нуля. Це призводить до збільшення параметра, наближаючи його значення до одиниці. Це означає, що імпульсна характеристика подовжується, а, відповідно, полоса пропускання частот звужується. Важливо врахувати, що фільтрація не тільки послаблює вплив шуму, але також вносить динамічне спотворення в корисний сигнал. Механізм цих спотворень полягає в нерівномірному передачі на вихід фільтра різноманітних спектральних компонентів сигналу. Робота фільтра Калмана особливо корисна при зникненні шуму на вході, оскільки при таких умовах зникають і динамічні перешкоди фільтрації.

Зображення, яке пройшло етап попередньої обробки, можна використати для реалізації алгоритмічних методів різноманітного призначення. Розглянемо метод виявлення зміщення між сусідніми кадрами, який дозволяє отримати інформацію щодо зміщення не для груп, а для кожного окремого пікселя в зображенні. Метод можливий к застосуванню за умови, що швидкість змін потоку оптичної інформації не критична. Описаний метод називають диференційним, оскільки його основа складається з обрахування часткових похідних по горизонтальному й вертикальному напрямку інформації. Оскільки будова математичної моделі лише часткових похідних функції є недостатньо щоб визначити зміщення, то має місце використання методу Лукаса-Канаде.

Основне положення метод трактується наступним чином: хай значення пікселів переходять од одного до іншого кадру без змін. Тоді й можна стверджувати, що пікселі, що відносяться до одного й того ж об'єкта, можуть направлятися у будь-яку сторону. При цьому значення пікселя лишається незмінним. На практиці буває поява незначних проблем, адже із появою нового кадру виникають різні глобальні умови. До подібних умов якраз відносяться наступні: освітлення й освітленість самого об'єкта, який рухається.

Математично метод Лукаса-Канаде можна записати наступним чином:[15]

$$I(x, y, t) = I(x + u_x, y + u_y, t + 1), \quad (2.4)$$

де: I – функція яскравості пікселів та положення на кадрі i в часі; x та y – координати пікселя у площині кадру; u_x, u_y – зміщення; t – номер кадру у послідовності.

Формула має сенс при умови, що між двома сусідніми кадрами є одиничний відрізок часу.

Для двомірного уявлення запишемо розклад у ряд Тейлора для виразу $I(x + u_x, y + u_y, t)$ із спрощеннями:

$$I(x + u_x, y + u_y, t) = I(x, y, t) + \vec{u} \nabla I(x, y, t), \quad (2.5)$$

де $\vec{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$ – вектор зміщення.

Відповідно до припущення (2.4) маємо:

$$I(x + u_x, y + u_y, t + 1) = I(x, y, t + 1). \quad (2.6)$$

Тоді формула (2.5) набуває вигляду:

$$I(x + u_x, y + u_y, t + 1) = I(x, y, t) + \vec{u} \nabla I(x, y, t). \quad (2.7)$$

Тоді можна записати наступний вираз:

$$I(x + u_x, y + u_y, t + 1) - I(x, y, t + 1) + \vec{u} \nabla I(x, y, t) = 0. \quad (2.8)$$

Оскільки між двома кадрами проходить одиничний інтервал часу, то можна стверджувати, що вираз $I(x, y, t) - I(x, y, t + 1)$ – це похідна часу, а отже:

$$\frac{\partial I(x, y, t)}{\partial t} + u_x \frac{\partial I(x, y, t)}{\partial x} + u_y \frac{\partial I(x, y, t)}{\partial y} = 0. \quad (2.9)$$

З рівняння отримуємо, що сума часткових похідних має рівнятися нулю, а тому необхідно зробити ще одне припущення: що сусідні пікселі зміщуються на однакову відстань.

Необхідно знайти такі u_x й u_y , які мінімізують похибку:

$$E(u_x, u_y) = \sum_{i,j} g(x_i, y_j) \left[u_x \frac{\partial I(x_i, y_j, t)}{\partial x} + u_y \frac{\partial I(x_i, y_j, t)}{\partial y} + \frac{\partial I(x_i, y_j, t)}{\partial t} \right]^2, \quad (2.10)$$

де g – функція вагових коефіцієнтів для пікселів.

Щоб знайти мінімум $E(u_x, u_y)$ використовуємо метод найменших квадратів, часткові похідні по u_x й u_y обраховуються по відповідним формулам:

$$\frac{\partial E(u_x, u_y)}{\partial u_x} = \sum_{i,j} g(x_i, y_j) \left[u_x \left(\frac{\partial I}{\partial x} \right)^2 + u_y \frac{\partial I}{\partial y} \frac{\partial I}{\partial x} + \frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right] = 0. \quad (2.11)$$

$$\frac{\partial E(u_x, u_y)}{\partial u_y} = \sum_{i,j} g(x_i, y_j) \left[u_y \left(\frac{\partial I}{\partial y} \right)^2 + u_x \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right] = 0. \quad (2.12)$$

Математична форма рівнянь (2.11) і (2.12) має вигляд:

$$M \vec{u} = \vec{b}, \quad (2.13)$$

де:

$$M = \begin{bmatrix} \sum_{i,j} g(x_i, y_j) \left(\frac{\partial I}{\partial x} \right)^2 & \sum_{i,j} g(x_i, y_j) \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \sum_{i,j} g(x_i, y_j) \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \sum_{i,j} g(x_i, y_j) \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix}; \quad (2.14)$$

$$\vec{b} = - \begin{bmatrix} \sum_{i,j} g(x_i, y_j) \frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \\ \sum_{i,j} g(x_i, y_j) \frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \end{bmatrix}; \quad (2.15)$$

$$\vec{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \quad (2.16)$$

u_x та u_y мінімізують помилку E :

$$\vec{u} = M^{-1} \vec{b}. \quad (2.17)$$

Оскільки при знаходженні зміщення для кожного пікселя впливають сусідні з ним пікселі, при реалізації цього метода доцільно попередньо обрахувати похідні кадра по горизонталі й вертикалі.

Нехай $x_{i,j}$ – це значення яскравості зображення – іншими словами корисного сигналу на перетині i -ого рядка і j -го стовпця. Тоді зображення, яке подається на вхід фільтра буде описуватися такою моделлю:

$$y_{i,j} = f(x_{i,j}, n_{i,j}), \quad i = \overline{0, I-1}, \quad j = \overline{0, J-1}. \quad (2.18)$$

де: $n_{i,j}$ – значення перепони в точці з координатами (i, j) ; $f(x_{i,j}, n_{i,j})$ – функція, яка описує взаємодію сигналу й завади; I, J – відповідно кількість рядків і стовпців у кадрі.

У залежності від типу площини зображень розрізняють казуальну, неказуальну й напівказуальну фільтрацію зображень. Поняття казуальності (причинно-наслідкової залежності) пов'язують з залежністю координат поточної точки i, j та точок, які входять у площину. Якщо обидві координати всіх точок площини не перевищують вказаних координат поточної точки, то площина і її обробка називаються каузальними.

Найбільш поширеним критерієм оптимальності, який застосовується для оцінки якості обробки, є критерій мінімуму середнього квадрата помилок. До застосування до фільтрації його вираз має такий вигляд:[17]

$$E \left\{ \left[x(i, j) - \sum_{(i_1, j_1) \in S} a_{i_1, j_1} \cdot y(i - i_1, j - j_1) \right]^2 \right\} = \min_{a(\cdot)} \quad (2.19)$$

де $E\{\cdot\}$ – символ математичних очікувань.

Лінійне і відносно коефіцієнтів a_{i_1, j_1} , що потрібно знайти, алгебраїчне рівняння має вид:

$$B_{xy}(k, l) = \sum_{i_1, j_1} \sum_{\in S} a_{i_1, j_1} \cdot B_y(k - i_1, l - j_1), \quad (2.20)$$

Кількість невідомих рівнянь тотожно рівне кількості точок n_S в площині S та являється скінченною. Якщо рівняння (2.20) диференціювати по невідомим $n_S - 1$, то можна отримаємо ще $n_S - 1$ рівнянь, які відрізняються одне від іншого лівими частинами $B_{xy}(\cdot)$ та коефіцієнтами $B_y(\cdot)$ в правих частинах. В результаті утворюється система n_S лінійних алгебраїчних рівнянь із n_S невідомими, що називається рівнянням Віннера-Хопфа:[5]

$$\begin{cases} B_{xy}(k, l) = \sum_{i_1, j_1} \sum_{\in S} a_{i_1, j_1} \cdot B_y(k - i_1, l - j_1), \\ k, l \in S. \end{cases} \quad (2.21)$$

З рівняння (2.21) знайдемо імпульсну характеристика для лінійного фільтра, яка мінімізує квадрат помилок фільтрації:

$$D_{\varepsilon} = D_x \left[1 - \sum_{i_1, j_1} \sum_{\in S} a_{i_1, j_1} \cdot B_x(i, j) \right]. \quad (2.22)$$

Маскова фільтрація зображень за наявністю білого шуму базується на наступному. Як вже раніше зазначалося, поширеним видом завади являється білий шум, який адитивно впливає на зображення. Зображення, за яким йде спостереження, в цьому випадку має наступний вигляд:

$$y_{i,j} = x_{i,j} + n_{i,j}, \quad (2.23)$$

де: $i = \overline{0, I-1}, j = \overline{0, J-1}$.

Кореляційна функція шуму $n_{i,j}$ описується таким виразом:

$$B_n(i, j) = D_n \cdot \delta_i \cdot \delta_j. \quad (2.24)$$

У виразі (2.24) D_n – це дисперсія шуму, а $\delta_k = \begin{cases} 1, & \text{при } k = 0 \\ 0, & \text{при } k \neq 0 \end{cases}$ – символ Кронекера.[8]

Нехай вхідний сигнал являється центрованим, тобто він має математичне очікування, а зображення x й шум n взаємно незалежні, тоді для кореляційної функції вхідного сигналу будемо вважати справедливим наступний вираз:

$$B_y(i, j) = D_x \cdot R_x(i, j) + D_n \cdot \delta_i \cdot \delta_j, \quad (2.25)$$

де D_x – це дисперсія, а $R_x(i, j)$ – нормована кореляційна функція корисного сигналу.

З виразу (2.25) видно, що у заданих умовах взаємна кореляційна функція B_{xy} співпадає із кореляційною функцією корисного сигналу B_x . Тому рівняння Віннера-Хопфа (2.21) набуває наступного вигляду:

$$\begin{cases} q^2 \cdot R_x(k, l) = q^2 \cdot \sum_{i_1, j_1} \sum_{\in S} a_{i_1, j_1} \cdot R_x(k - i_1, l - j_1) + a_{k, l}, \\ k, l \in S \end{cases}, \quad (2.26)$$

де $q^2 = D_x/D_n$ – відношення дисперсії сигналу шуму.

Перетворюючи вираз (2.22) для помилок фільтрації, маємо:

$$a_{0,0}/q^2 = 1 - \sum_{i_1, j_1} \sum_{\in S} a_{i_1, j_1} \cdot R_x(i_1, j_1) + a_{0,0}. \quad (2.27)$$

Порівнюючи вирази (2.27) й (2.22) остаточно маємо:

$$\delta_{\varepsilon}^2 = a_{0,0}/q^2, \quad (2.28)$$

де $\delta_{\varepsilon}^2 = D_{\varepsilon}/D_x$ – відносний середній квадрат помилок фільтрації.

Таким чином, задля виявлення помилок фільтрації потрібно знати відношення сигнал/ шум (також складова рівняння Віннера-Хопфа) та значення оптимальної імпульсної характеристики у точці (0,0).

Щоб розв'язуючи рівняння (2.26), скористатися існуючими програмними середовищами обчислювальних машин, необхідно виконати його впорядковане перетворення до канонічного векторно-матричного виду. Для цього необхідно сукупність n_S невідомих величин $a_{i,j}$ представити в вигляді вектору A . Аналогічно множину величин, які утворюють ліві частини виразу (2.26), необхідно представити в вигляді вектору Q , а множину коефіцієнтів правої частини в вигляді матриці F розміру $n_S \times n_S$. Тоді рівняння і його розв'язок набудуть наступного вигляду:[18]

$$Q = F \cdot A, A = F^{-1} \cdot Q. \quad (2.29)$$

На практиці для цифрової обробки візуальної інформації широко використовується маскова фільтрація. Її варіація є однією з форм двомірної фільтрації. У якості маски використовується набір вагових коефіцієнтів, що визначені в усіх точках площини S , зазвичай симетрично оточуючих активну точку кадра. Розповсюдженим типом площини, який часто застосовується на практиці, є квадрат 3×3 з робочим елементом в центральній частині зображення. Застосовують різноманітні варіанти масок, одним із евристичних виборів є рівномірна маска, всі дев'ять вагових коефіцієнтів якої мають значення $1/9$. Такий вибір коефіцієнтів відповідає умові збереження середньої яскравості і, отже, в процесі обробки немає необхідності центрувати зображення.

При використанні маскової фільтрації необхідно враховувати різноманіття підходів до оцінки якості, як за допомогою людського зору, так і застосованими кількісними показниками. Тому деякі результати, які в математичному відношенні вважаються критичними, можуть бути прийнятними візуально для так званого оцінювання людським око. Однак, незважаючи на це, цифрова обробка зображень залишається важливою в різноманітних інформаційних системах з автоматичним прийняттям рішень, які ґрунтуються на такій обробці.[19]

Автоматизована або напівавтоматизована система оцінює якість зображення за допомогою математичних показників, тому результат системи може відрізнитися за показниками якості від систем спостереження, в яких задіяна

людина. Функціонування таких систем, де відсутнє людське око, повністю підпорядковане математичним критеріям, і якість їх роботи оцінюється тільки за математичними показниками. Зрозуміло, що і якість зображень, що використовуються в цих системах, також повинна оцінюватися виключно за математичними критеріями.

У цілому використання процедур фільтрації призводить до значного зменшення рівня шуму на зображенні [22].

2.3 Побудова алгоритму стерео-бачення для визначення відстані

Об'єднання зображень, які реєструє пара людських очей, і використання їх відмінностей дозволяє отримати уявлення про глибину об'єкта.

За допомогою камер, розташованих на роботизованій платформі, є можливість створення оптичної системи для визначення відстані до об'єкта, що перебуває в полі зору цих камер.

Базою розроблюваної оптичної системи для визначення відстані до об'єкта є алгоритм комп'ютерного стереозору. Цей алгоритм потрібно програмно описати, щоб обчислювач роботизованої платформи міг його використовувати.

Стереозор включає два процеси: об'єднання деталей, які фіксуються двома або більше камерами, та відновлення їх тривимірного образу. Останній процес є досить простим: образ відповідних точок можна знайти як точку перетину променів, що проходять через ці точки та центри відповідних діафрагм камер. Таким чином, якщо характерну точку зображення можна спостерігати в будь-який момент часу, стереозор реалізується просто. При наявності відкаліброваного стереопристрою та двох визначених точок зображення завдання відновлення відповідності точки образу – це досить простий процес, що пояснюється геометрією визначення відстані до об'єкта.[23]

Якщо в полі зору двох камер, ідентичних одна одній, на відстані D від осі, де розташовані камери, то ці камери мають однакове значення фокусної відстані ($f_1=f_2=f$). Фокусна відстань вимірюється в мм. Відстань між оптичною віссю камер 1 та 2 в такому випадку відповідає значенню L , мм.

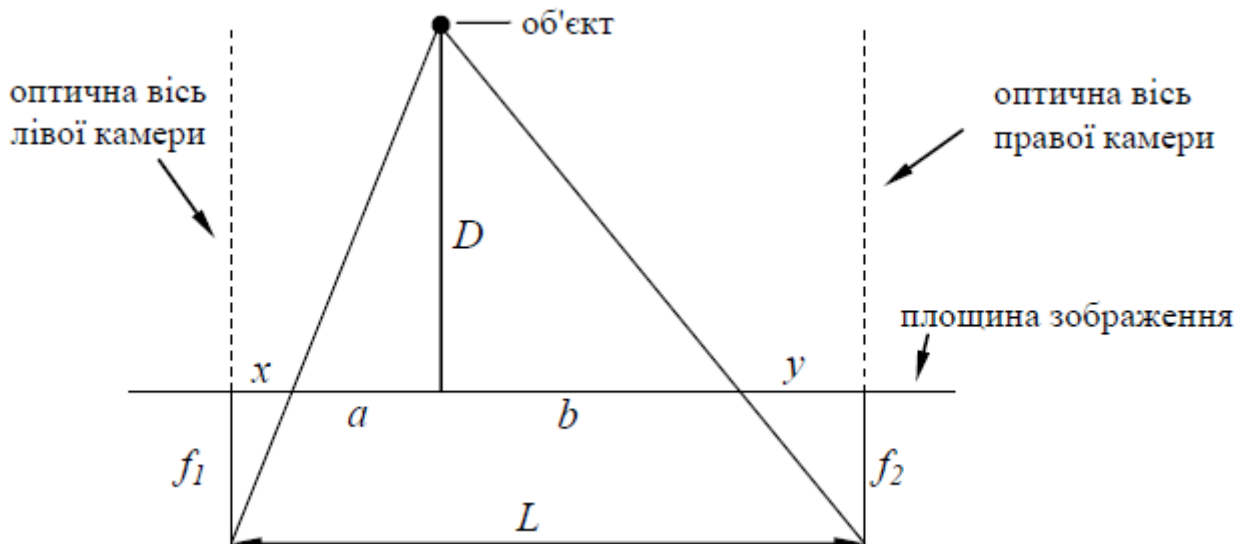


Рисунок 2.5 – Геометрична форма визначення відстані в напрямку об'єкту

На рисунку 2.5 прийняті такі позначення: x – горизонтальне зміщення першого зображення, мм; y – горизонтальне зміщення другого зображення, мм; $H = x + y$ – горизонтальне зміщення на двох зображеннях.

Враховуючи геометричну модель (на рис. 2.5) можна записати наступні співвідношення:

$$L = x + y + a + b ,$$

$$\frac{f}{x} = \frac{D}{a} \text{ та } \frac{f}{y} = \frac{D}{b},$$

$$x + y + x \frac{D}{f} + y \frac{D}{f} = (x + y) \left(1 + \frac{D}{f} \right) \rightarrow \quad (2.30)$$

$$\rightarrow D = \left(\frac{L}{x + y} - 1 \right) f.$$

Рівняння (2.30) являються математичною формою визначення відстані до відомого об'єкта за відомими параметрами.

Щоб визначити горизонтальне зміщення обох тих зображень чи їх сумарного зміщення (H) необхідно розробити алгоритм на основі кореляційно-екстремального методу визначення подібності зображень.

Алгоритм визначення зміщення кадрів складається з десяти кроків.

I крок. Отримується зображення розмірами 640x480 точок від правої й лівої камер. Це значення конвертується у форматі *JPEG*.

II етап. Здійснюється процес розфокусування двох зображень за допомогою усереднюючого фільтру розміром 10×10 .

III етап. Праве та ліве зображення розділяється на сім однакових частин. По вертикалі – $[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8]$, по горизонталі – $[y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7 \ y_8]$, де x_i, y_j – координати пікселів зображення. Таким чином, отримується сітка із визначеним кроком по вертикалі та горизонталі (n_1 та m_1 , відповідно).

IV етап. Створюється початкова нульова матриця Corr із розмірами, які відповідають розмірам розфокусованого зображення. Якщо розмір вихідного зображення становить 480×640 точок, то розфокусоване зображення буде мати розміри 337×636 точок. Конкретна матриця буде використовуватися для запису значень кореляційної функції, що отримуються на подальших етапах.

V етап. Виконання циклічного переміщення правого зображення вниз на величину, яка відповідає значенню x_i , а також вправо – y_j . Значення зміщень на початковому етапі дорівнюють одиниці (1 піксель). Графічний алгоритм циклічного переміщення подано на рисунку 2.6.

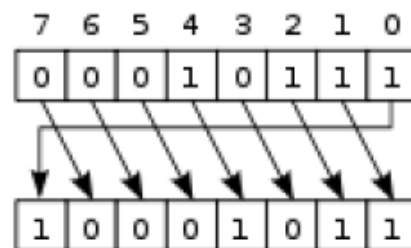


Рисунок 2.6 – Циклічно зміщення вправо рядка зображення

VI етап. Обчислення кореляційних функцій між вихідним лівим зображенням та циклічно переміщеним зображенням. Переміщення виконується для всіх можливих варіацій x_i, y_j . Значення, отримані в результаті обчислення кореляційних функцій, фіксуються у відповідних вузлах сітки матриці Corr .

VII етап. Знаходження максимальних значень кореляційних функцій у матриці відповідних вузлів сітки та визначення координат зображення ($x_{\text{max}}, y_{\text{max}}$), що відповідають цим максимальним значенням.

VIII етап. Виділення підобласті розміром приблизно 84×158 пікселів навколо точки, що відповідає максимальному значенню функції кореляції з координатами, визначеними на попередньому етапі. Ця підобласть має значно менші розміри, що дозволяє детальніше вивчити область навколо максимального значення функції кореляції.

IX етап. Розділення підобласті з етапу VIII на 7 однакових частин по вертикалі та горизонталі. Розділення відбувається аналогічно до етапу III. Формується нова сітка з кроком n_2 для вертикалі та m_2 – для горизонталі, відповідно.

X етап. Кроки з V по VII повторюються, а всі знайдені значення кореляційної функції складають одну матрицю Corr. Цикл повторюється до тих пір, поки кроки сітки не будуть знову дорівнювати одиниці ($n_q=m_p=1$).

Після завершення циклу матриця Corr матиме форму, яку зображено на рисунку 2.7. На площині кореляційної функції відбудеться певна модифікація (рисунок 2.8).

З рисунків 2.7 та 2.8 можна виділити явний пік – це найбільша концентрація точок усієї функції кореляції. Координати точки піку показують зміщення правого зображення відносно лівого у двох напрямках (H – горизонтальне зміщення; V – вертикальне).

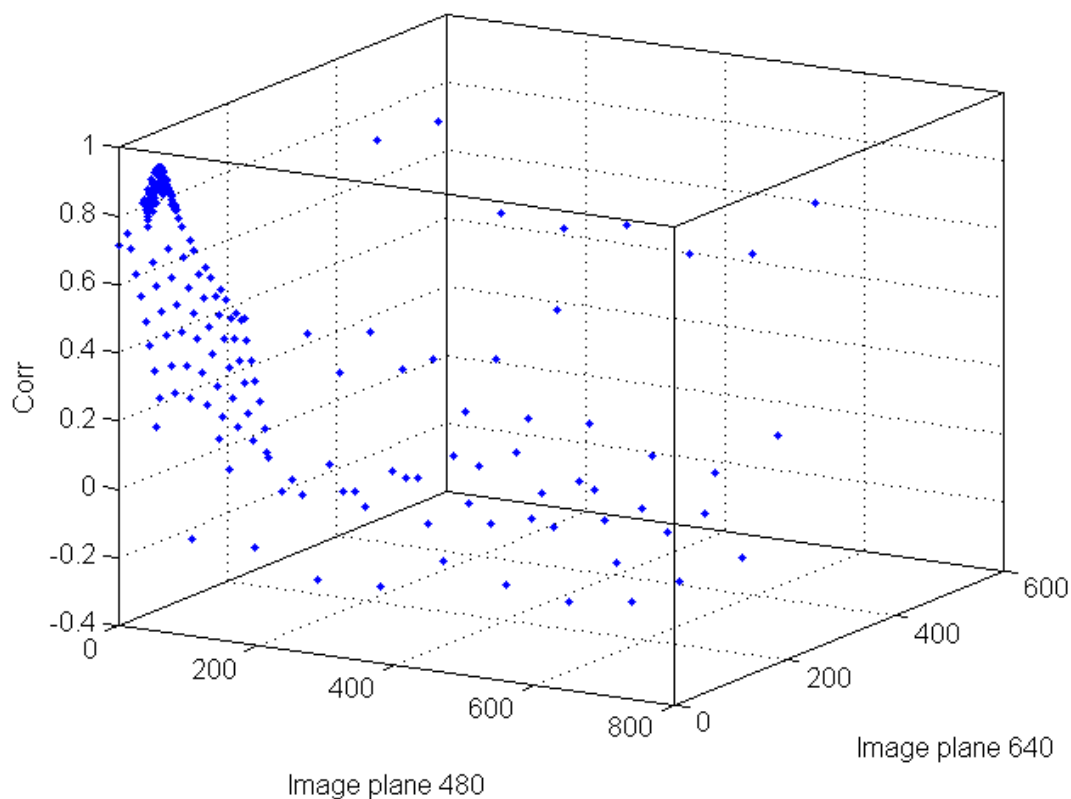


Рисунок 2.7 – Тривимірне зображення результуючої кореляційної функції

Для визначення відстані до об'єкта за формулою (2.30) відіграє роль тільки горизонтальне зміщення. Адже вертикальне зміщення виникає з-за недоліки, які були не враховані при встановленні камер і їх калібруванні.

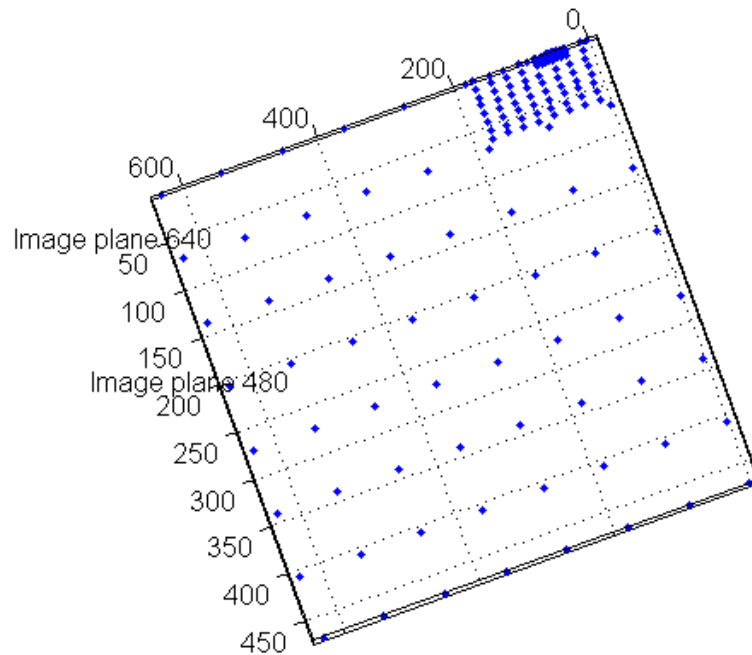


Рисунок 2.8 – Вигляд кореляційної камери в 2D

Наведений метод був описаний за допомогою програмних процедур для забезпечення можливості подальших тестів.

Для підтвердження розробленого методу необхідно провести експериментальне дослідження для визначення відстані до об'єкта.

Оскільки при визначенні відстані між осями двох камер виникають певні труднощі, то встановлення відстані можна виконати приблизно. Це реалізується за допомогою прямих вимірювань. Крім того, виявлення фокусної відстані камер може бути важливим, якщо це значення не відомо безпосередньо від виробника. Для уникнення цих труднощів обрано метод визначення співвідношення між горизонтальним зміщенням для двох зображень H та відстанню D , що представляє відстань до об'єкта, експериментально. Зміщення обчислюється програмою, а відстань до об'єкта вимірюється вручну, за допомогою рулетки.

2.4 Алгоритм пошуків подібності SURF

Один із варіантів вирішення завдання сегментації потоків даних полягає впливі послідовності ключових кадрів відео. Поняття визначення ключових кадрів, у свою чергу, ґрунтується на формалізації опису відеокадру: визначенні дескрипторів зображення та їх порівнянні за визначеною метрикою. Дескриптори мають бути унікальними та володіти певними властивостями, такими як інваріантність до різних перетворень та стійкість. Очевидно, що при розв'язанні реальних задач дескриптори не завжди володіють усіма унікальними характеристиками, тому вибирається той, який найкращим чином відповідає

пріоритетній функціональності, враховуючи особливості предметної області та технічні характеристики зображень.

Залежно від математичного апарату, що використовується, алгоритми пошуку ключових точок зображення можуть бути спрямовані на виявлення кутів та опис зображення в термінах регіонів. До першої групи можна віднести, наприклад, детектор Харріса та метод FAST, до другої – SIFT, SURF, BRIEF. Процес побудови можна умовно розділити на виявлення ключових точок зображення та безпосереднє формування дескриптору.

Аналіз відеоданих потребує значних обчислювальних ресурсів. Головною перевагою використання сигнатури у вигляді дескриптору є ефективність обчислень. Однак такий підхід не надає необхідної інформації для представлення складної семантики зображення. Для вирішення цієї проблеми можна використовувати конкатенацію дескрипторів окремих областей, що робить завдання розділення зображення на окремі об'єкти та зберігання інформації про їх просторовий розподіл більш досяжним.

При аналізі відеоданих важливо враховувати, що послідовні відеокадри можуть відрізнятися незначно, і кількість спільних ключових точок може бути значною. З урахуванням цього, важливо детально відбирати метрику порівняння, яка дозволяє виявляти невеликі зміни.

Із розмаїття існуючих методів розпізнавання образів виділяється той, який дозволяє наблизитися до вирішення задачі розпізнавання. Метод Speeded Up Robust Features (SURF) є одним з найефективніших та швидких алгоритмів сучасності. До того ж цей алгоритм вже досить поширений, і математичні бібліотеки для його реалізації є доступними.

Суть методу пошуку стійких ознак полягає в аналізі зразка та сцени, на якій може знаходитися зразок, який необхідно віднайти. Ключові точки та невеликі ділянки навколо них виділяються на зразку. Ключовою точкою вважається та, яка має певні ознаки, відрізняються від основної маси точок, такі як краї ліній, невеликі кола, різкі перепади освітленості, кути і т. д. Враховуючи, що ключові точки завжди присутні на зразку, можна звести його пошук до виявлення на сцені ключових точок зразку. Оскільки ключові точки суттєво відрізняються від основної маси точок, їх кількість буде суттєво меншою, ніж загальна кількість точок зразку. В цілому принцип вибору ключових точок не важливий. Головне, щоб їх було не забагато та вони були присутніми на зображення зразку завжди. В процесі обробки візуальної інформації зустрічаються поняття, що потребують додаткового уточнення.

Масштаб. Зображення різняться за своїм масштабом. Ті предмети, які здаються однаковими, насправді можуть мати різні розміри на різних зображеннях.

Розташування. Об'єкт, який може перебувати в різних місцях на зображенні.

Фон і перешкоди. Предмет, що вважається окремим, на зображенні не виділений і знаходиться на задньому плані поруч з іншими предметами. Крім того, зображення може бути спотворене або забруднене будь-якими видами викривлення та завад.

Проектування, обертання та кут огляду. Зображення представляє собою лише двовимірну проекцію існуючого тривимірного світу. Тому обертання об'єкта та зміна кута огляду суттєво впливають на його двовимірну проекцію – зображення. Один і той же об'єкт може мати абсолютно різний вигляд в залежності від обертання чи відстані до нього.

Метод SURF може вирішувати дві задачі – виявлення особливих точок на зображенні та створення їх дескрипторів, що залишаються стійкими до масштабу та обертання.

Для пошуку ключових точок на зображенні використовується інтегральне уявлення зображень:

$$H = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(x, y), \quad (2.31)$$

де $I(i, j)$ – яскравість пікселя початкового зображення.

Кожен елемент матриці $H[x, y]$ являє собою суму пікселів в прямокутнику від $(0; 0)$ до $(x; y)$. Розрахунок матриці займає послідовний час, що пропорційний кількості пікселів в зображенні. Маючи інтегральну матрицю, можна доволі швидко обрахувати суму яскравості пікселів довільних прямокутників областей зображення відповідно формули:

$$H_{ABCD} = H(A) + H(C) - H(B) - H(D), \quad (2.32)$$

де $ABCD$ – це прямокутник.

Для впізнання особливих точок використовується гессіан, що є представником матриці Гессе. Детермінант гессіана досягає екстремуму в зонах максимальних змін градієнту яскравості. Після визначення ключових точок методом SURF формують для них дескриптори. Дескриптор - це набір із 64 (або 128) чисел для кожної визначальної точки.

Матрицю Гессе для двовимірної функції та її детермінант визначають за таким способом:

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}; \quad (2.33)$$

$$\det(H) = \frac{\partial^2 f}{\partial x^2} \cdot \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2.$$

Застосування гессіана використовується для виявлення локального мінімуму чи максимуму яскравості на зображенні. В цих точках значення гессіана досягає екстремуму.

Розрахунок матриці Гессе полягає в обчисленні лапласіана-гауссіана. Елементи матриці Гессе обчислюються як сума добутків значень пікселів зображення на фільтри - бінаризовану апроксимацію лапласіана-гауссіана (Fast-Hessian) (2.33).

Мотивація застосування бінаризованих фільтрів:

1) Дискретизований лапласіан-гауссіан має досить великий розкид значень детермінанту при обертанні зразка. Максимальна похибка можлива в області повороту на 45 градусів.

2) Фільтр для лапласіана-гауссіана має неперервний характер. У фільтрі майже в усіх пікселів різні значення яскравості, що не дозволяє ефективно використовувати такий потужний механізм розрахунку, як інтегральна матриця зображення.

Отже, в SURF гессіан обчислюється наступним чином:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0,9D_{xy})^2, \quad (2.34)$$

де D_{xx} , D_{yy} , D_{xy} – сума добутків по фільтрам, які зображені на рисунку 2.7. Коефіцієнт 0,9 має теоретичне обґрунтування й коректує приблизний характер обрахувань.

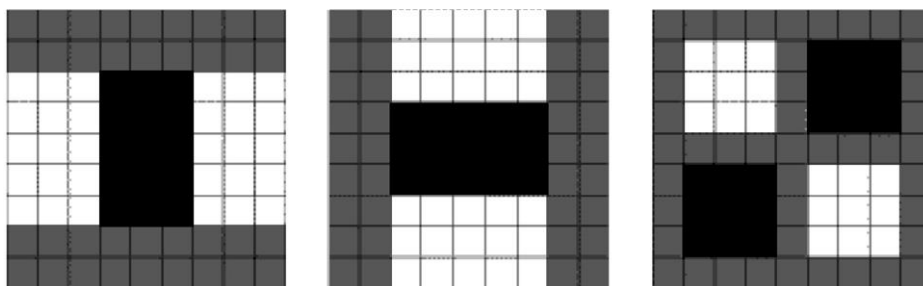


Рисунок 2.9 – Фільтри, які використовуються для знаходження матриці Гессе в SURF

На зображенні 2.9 світлі області відповідають значенню +1, темні – 2 (на третьому фільтрі – 1), середні – нульові. Розмір зображення у просторі - 9x9 пікселів.

Оскільки гессіан є похідною та залежить тільки від перепаду яскравості, але не від абсолютного її рівня, він залишається незмінним щодо зсуву яскравості зображення. Таким чином, зміна рівня освітлення зразку не впливає на виявлення ключових точок.

Крім того, властивості гессіана такі, що він досягає максимуму в точці як білої плями на чорному тлі, так і чорної – на білому. Таким чином, метод реагує як на темні, так і на світлі особливості зображення.[25]

Оскільки гессіан не є інваріантним до масштабу, необхідно вибирати різні масштаби фільтрів та послідовно застосовувати їх до конкретного пікселя. При аналізі зображень до 10 мегапікселів використовуються 5-7 гессіанів різної розмірності – октави (рис. 2.10). Це пов'язано з тим, що при аналізі реальних зображень розмір об'єктів може змінюватися в чітких розмірах, враховуючи обмеження розміром зображення.

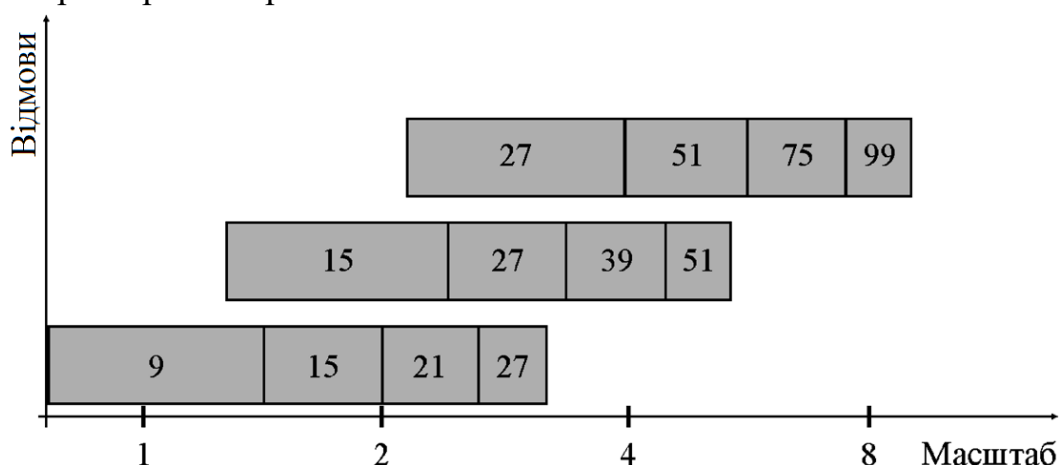


Рисунок 2.10 – Перші три октави по методу SURF

На рисунку 2.10 цифри у прямокутниках вказують розмір фільтра Fast-Hessian; логарифмічна шкала знизу показує масштаби, що охоплюються октавами.

Для виявлення локального екстремуму гессіана потрібно проаналізувати багатомірну матрицю розміром $W \times H \times I$, де W та H – розмір зображення, яке аналізується, по горизонталі та вертикалі, а I – припустимий масштаб (пропорційний кількості октав).

Для визначення точки справжнього екстремуму використовується інтерполяція знайдених гессіанів кубом $3 \times 3 \times 3$ квадратичною функцією (враховуючи умову пошуку - кількість октав не може бути менше трьох). Потім обчислюється похідна (методом кінцевих різниць сусідніх точок). Якщо вона близька до нуля, точка справжнього екстремуму знайдена. Якщо похідна велика, то її потрібно зменшити та повторювати це, доки вона не стане менше заданого

порогу. Якщо в процесі ітерацій відбувається віддалення від початкової точки, то вважається, що точка екстремуму була визначена неправильно і це означає, що вона вже не вважається особливою.

Визначення орієнтації особливих точок та обчислення дескриптора. Розрахунок орієнтації особливих точок необхідний для забезпечення інваріантності дескрипторів особливих точок відносно обертання. З цією метою визначається переважна орієнтація перепадів яскравості в особливій точці. Спочатку обчислюються точкові градієнти в пікселях, які межують з особливою точкою. Для розгляду беруться пікселі навколо радіуса bs навколо особливої точки, де s – масштаб особливої точки (перша октава, $s=2$).

Незважаючи на те, що SURF використовується для пошуку об'єктів на зображенні, він сам не працює безпосередньо з об'єктами; SURF не виділяє об'єкт із фону, розглядаючи зображення як єдине ціле та шукаючи особливості цього зображення. При цьому особливості можуть бути як всередині об'єкта, так і на фоні, а також на точках границі об'єкта та фону. Це призводить до того, що метод неефективно працює для об'єктів простої форми чи з слабо вираженою текстурою. У межах таких об'єктів метод, скоріше за все, не знаходить особливих точок, або вони знаходяться тільки на границі об'єкта із фоном. Це може призвести до того, що об'єкт не може бути ефективно визначений на іншому зображенні чи фоні.

Алгоритм пошуку об'єктів інтерфейсу на цифрових зображеннях застосовується з високою ефективністю у задачах автоматизованого аналізу фотоданих. Серед переваг методу слід відзначити інваріантність до обертання відносно площини захоплення та масштабування. Однак використання методу не дозволяє виділити об'єкт із фону, а лише визначає його місцезнаходження та розмір. Оскільки алгоритм розглядає зображення як єдине ціле, його використання може бути малоефективним при пошуку об'єктів простої форми чи із слабо вираженою текстурою. Також слід відзначити, що обмежена швидкість обробки зображень обмежує використання алгоритму при аналізі відеоданих в реальному режимі часу, з частотою кадрів, що перевищує 0,3 fps.

2.5 Визначення оптимального шляху, алгоритм Дейкстри

Завдяки методу Дейкстри можна виконати пошук найкоротшого шляху від одного до інших вузлів графа (рис. 2.11).

Цей алгоритм застосовується виключно до графів, що не містять ребер із від'ємною вагою. Даний метод знаходить широке використання в програмуванні та технологіях, зокрема, у протоколах динамічної маршрутизації (OSPF) та протоколах маршрутизації проміжних систем (IS-IS - протокол внутрішніх шлюзів).

Кожній вершині призначається мітка відповідно до алгоритму - це мінімальна відома відстань від вершини "a".

Метод Дейкстри працює крок за кроком, на кожному етапі він "відвідує" одну вершину і намагається зменшити мітки. Робота алгоритму завершується, коли всі вершини відвідані.

Ініціалізація виконується наступним чином: мітка вершини "a" встановлюється на нуль, а мітки всіх інших вершин - на нескінченність. З цього припущення випливає, що відстань від вершини "a" до інших вершин є невідомою. Усі вершини, крім "a", позначаються на графі як невідвідані.

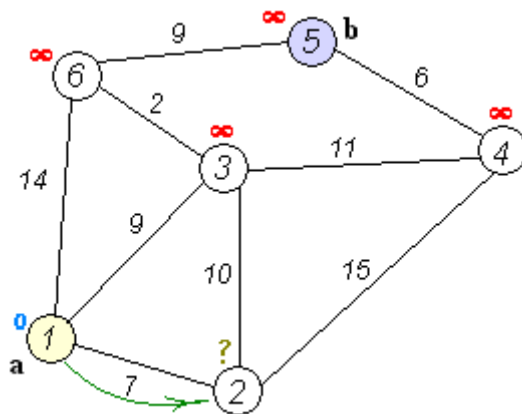


Рисунок 2.11 – Приклад графу для роботи алгоритмом Дейкстри

Етап алгоритму Дейкстри. Якщо всі вершини вже пройдені, алгоритм завершується. У протилежному випадку, серед тих вершин, які ще не були відвідані, обирається вершина u з мінімальним значенням мітки. Розглядаються всі можливі маршрути, в яких вершина u є кінцевою. Такі вершини, до яких є ребра з вершини u , називають сусідами цієї вершини. Для кожного сусіда вершини u , крім тих, що вже відзначені як пройдені, аналізується нова довжина маршруту, яка є сумою значень поточної мітки u та довжини ребра, яке з'єднує вершину u з конкретним сусідом. Якщо отримане значення довжини менше, ніж значення мітки сусіда, необхідно замінити мітку отриманим значенням довжини. Після розгляду всіх значень сусідів, вершина u відзначається як вже відвідана. Далі алгоритм циклічно повторюється до завершення.

Алгоритм завершується, коли жодна вершина не залишається для обробки.

Об'єктом дослідження є також проблема управління рухом колісних мобільних роботів (КМР).

КМР часто використовуються в промисловій та сервісній робототехніці, особливо там, де необхідні гнучкі можливості руху на достатньо рівних основах та поверхнях. Диференційний та синхронний привод, триколісний чи автомобільний привід, а також рульовий привід є найбільш поширеними для роботів із одним корпусом.

При відсутності перешкод основні завдання руху, призначені для КМР, полягають у переміщенні робота від однієї точки до іншої, а також русі по заданій траєкторії.

2.6 Основні задачі руху

Основними завданнями переміщення для КМР, які розглядаються в рамках дипломного проекту, у середовищі без перешкод є наступні:

а) рух "точка – точка": робот повинен досягти бажаної точки, визначеної конфігураціями як ціль, виходячи з визначеної початкової точки;

б) вказана траєкторія: контрольна точка на роботі повинна рухатися вздовж траєкторії, визначеної в декартовій площині, виходячи з початкової точки.

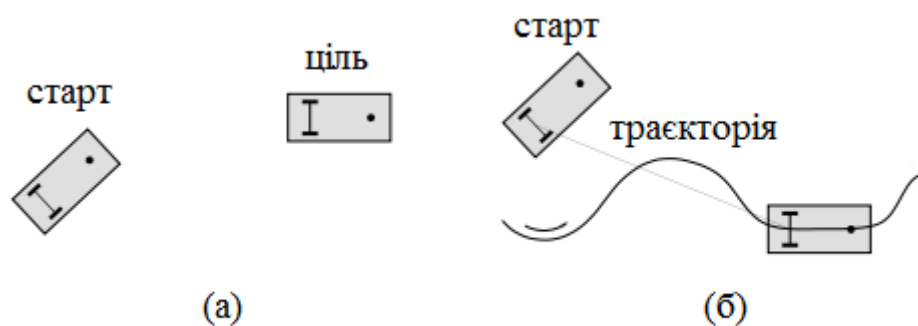


Рисунок 2.12 – Основні задачі по руху:

(а) рух «точка – точка»

(б) рух по вказаній траєкторії

Виконання цих завдань може бути досягнуто за допомогою команд оберненого зв'язку, форсування або їх комбінації. Дійсно, розв'язання оберненого зв'язку проявляє характерну для цього методу стійкість. Проте, особливо у випадку руху "точка – точка", створення оберненого зв'язку не дозволяє вирішити завдання зі складною структурою; використання контролерів цих методів призводить до незадовільних перехідних характеристик.

При використанні стратегії оберненого зв'язку задача "точка – точка" призводить до виникнення проблеми керування станом робота в просторі. Стабілізація положення є значущим завданням. Без обмеження простору ціль для руху "точка – точка" може бути розглянута як початок n -мірного простору конфігурації робота. При використанні траєкторного руху може виникнути проблема початкової похибки, спричинена запуском КМР поза траєкторійним середовищем.

Завдання управління асимптотичною траєкторією полягає в стабілізації нуля:

$$e_p = (e_x, e_y). \quad (2.35)$$

На практиці відстеження стає менш складною задачею, ніж регулювання КМР. Це обумовлено кількістю змінних, які підлягають контролю (кількість управляючих входів та виходів). Для одноколісного робота доступні лише дві команди, тоді як для визначення його конфігурації необхідні три – координати x , y та орієнтація θ . Отже, регулювання положення КМР за заданою конфігурацією передбачає порівняння з нулем трьох незалежних похибок конфігурації.

2.7 Властивості моделювання та управління

Нехай $q \in Q$ – n -вектор узагальнених координат для колісного мобільного робота. Системи відрізняються наявністю $n-m$ неінтегруючих диференціальних обмежень на узагальнені швидкості такого виду:

$$A(q)\dot{q} = 0. \quad (2.36)$$

Для КМР вони виникають через качіння без умов ковзання у коліс. Всі можливі миттєві переміщення можуть бути згенеровано наступним чином:

$$\dot{q} = G(q)\omega, \omega \in IR^m, \quad (2.37)$$

де стовпці $g_i, i = 1, \dots, m$ матриці $G(q) = n \times m$ вибрані так, аби захватити нульовий простір матриці $A(q)$. Різноманітні варіанти можливі у G , у відповідності з фізичною інтерпретацією, що може бути представлена «вагою» $\omega_1, \dots, \omega_m$. Рівняння (2.37), що називається кінематичною моделлю системи (першого порядку), являє собою безступінчасту нелінійну систему.

Сама проста модель КМР –одноколісний велосипед, що відповідає одному вертикальному колесу, яке обертається в площині. Узагальнені координати $q = (x, y, \theta) \in Q = IR^2 \times SO^1(n = 3)$. Обмеження, де колесо не може ковзати у поперечному напрямку, задається у вигляді зверху (рис. 2.13).

$$\dot{x} \sin\theta - \dot{y} \cos\theta = 0. \quad (2.38)$$

Таким чином, кінематична модель виглядає:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = g_1(q)v + g_2(q)\omega = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \quad (2.39)$$

де v та ω – лінійна швидкість колеса та його кутова швидкість навколо вертикальної осі, відповідно .

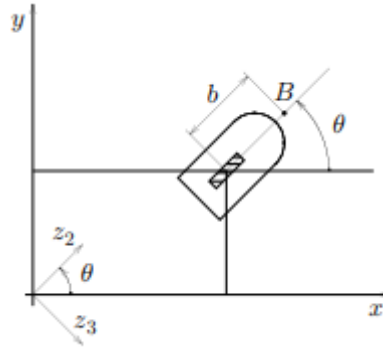


Рисунок 2.13 – Змінні одноколісного мобільного робота

Ці параметри вважаються доступними входами управляючими ($m = 2$).

Система (2.39) відображає ряд структурних управляючих властивостей, більшість яких фактично мають більше спільного для рівняння (2.37).

Керованість по точкам.

Дотична лінеаризація рівняння (2.39) в будь-якій точці q_e є лінійна система, яка не піддається контролю:

$$\dot{\tilde{q}} = \begin{bmatrix} \cos \theta_e \\ \sin \theta_e \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \quad \tilde{q} = q - q_e \quad (2.40)$$

Це означає, що лінійний контролер ніколи не досягає стабілізації положення навіть в локальному понятті. Щоб дослідити керованість одноколісного робота, необхідно використовувати інструменти із теорії нелінійного управління. Легко перевірити, що умова ранжування доступності виконується глобальна (при будь-якому q_e), оскільки:

$$\text{rank} [g_1 \ g_2 \ [g_1, g_2]] = 3 = n, \quad (2.41)$$

$$[g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2 = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix}.$$

Оскільки система дрейфує, з умови (2.41) випливає її управління.

Управління також може бути продемонстроване конструктивно, тобто шляхом порівняння явної послідовності маневрів, що приводить робота від будь-якої початкової конфігурації (x_s, y_s, θ_s) до будь-якої бажаної конфігурації цілі $[(x)_g, y_g, \theta_g)$. Оскільки одноколісний робот може обертатися сам по собі, це завдання досягається простим початковим обертанням на $[(x)_s, y_s)$ до тих пір, поки не відбудеться орієнтація на $[(x)_g, y_g)$, а потім буде переведено в положення цілі та завершено обертанням на $[(x)_g, y_g)$, щоб вирівняти θ із θ_g .

Щодо стабілізації системи (2.39) у точці, невдача попереднього лінійного аналізу вказує на те, що експоненціальна стійкість не може бути досягнена за допомогою гладкого зворотного зв'язку.

Фактично для отримання регулятора стабілізації положення потрібно або відмовитися від вимоги безперервності, або застосовувати змінні в часі закони управління.

Рух по траєкторії

Враховуючи бажаний декартовий рух для одноколісного робота, може бути зручно створити існуючу траєкторію стану $q_d(t) = (x_d(t), y_d(t), \theta_d(t))$. Щоб ця умова виконувалася, останній параметр повинен задовільняти обмеження відносно руху транспортного засобу або відповідати рівнянню (2.39).

Визначаючи похибку відстежування стану - $\tilde{v} = v - v_d$ та $\tilde{\omega} = \omega - \omega_d$, тангенц лінеаризація системи (2.39) відносно еталонної траєкторії:

$$\dot{\tilde{q}} = \begin{bmatrix} 0 & 0 & -v_d \sin\theta_d \\ 0 & 0 & v_d \cos\theta_d \\ 0 & 0 & 0 \end{bmatrix} \tilde{q} + \begin{bmatrix} \cos\theta_d & 0 \\ \sin\theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} = A(t)\tilde{q} + B(t) \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix}. \quad (2.42)$$

Оскільки при лінеаризації система змінюється з часом, необхідно та достатньою умовою керованості є те, що здатність бути керованим відносна. Однак більш простим способом аналіз коли проведено шляхом визначення похибки відстежування стану через матрицю обертання:

$$\tilde{q}_R = \begin{bmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q}. \quad (2.43)$$

Використовуючи рівняння (2.41), маємо:

$$\dot{\tilde{q}} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_R + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix}. \quad (2.44)$$

Коли v_d та ω_d стали, приведена вище лінійна система стає інваріантною в часі та управленою, оскільки наша матриця має ранг 3 за умови, що або v_d , або ω_d не дорівнюють нулю.

$$C = [B \ AB \ A^2B] = \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_d^2 & v_d \omega_d \\ 0 & 0 & -\omega_d & v_d & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.45)$$

А отже кінематична система (2.39) може бути локально стабілізована лінійним оберненим зв'язком відносно траєкторії, що складається із лінійних чи кутових параметрів, що виконуються із сталою швидкістю.

Отже, кінематичну систему (2.39) можна локально стабілізувати лінійним зворотнім зв'язком відносно траєкторії, яка складається із лінійних або кутових параметрів, що змінюються із постійною швидкістю.

Лінеаризованість зворотного зв'язку

Виходячи з обчислень у пункті 1.7.2, систему (2.37) неможливо перетворити в лінійну. Для рівняння (2.39) з умовою керованості (2.40) впливає, що розподіл, породжений векторними полями g_1 та g_2 , не є інволютивним, що порушує необхідну умову лінеаризації зворотного зв'язку повного стану.

Проте, коли матриця $G(q)$ в рівнянні (2.37) має повний ранг стовпця, m рівнянь завжди можуть бути лінійно перетворені за допомогою зворотного зв'язку в прості інтегратори. Вибір лінеаризованих виходів не є єдиним і може бути використаний для спеціальних цілей.

Для точної лінеаризації можна також звернутися до більш загального класу зворотного зв'язку із динамічною складовою. У цьому випадку повна лінійна лінеаризація є достатньою для широкого класу КМР. Проте існує потенційна контрольна особливість, яку слід уважно вивчати.

Ланцюгові форми. Існування канонічних форм для кінематичних моделей роботів дозволяє систематично розвивати як відкриті, так і закриті стратегії управління. Ланцюгова форма, що визначає закон управління, є найбільш корисною канонічною структурою для систем із двома входами.

$$\begin{aligned}
 \dot{z}_1 &= u_1; \\
 \dot{z}_2 &= u_2; \\
 \dot{z}_3 &= z_2 u_1; \\
 &\vdots \\
 \dot{z}_n &= z_{n-1} u_1.
 \end{aligned}
 \tag{2.46}$$

Таким чином, двокомпонентна система до $n = 4$ координат може бути перетворена у ланцюговій формі шляхом статичного перетворення зворотного зв'язку. Загалом більшість КМР можуть бути перетворені подібним чином у ланцюг.

Для кінематичної моделі (2.39) одноколісного робота шляхом перетворень та оберненого зв'язку маємо:

$$\begin{aligned}
 z_1 &= \theta ; \\
 z_2 &= x \cos\theta + y \sin\theta; \\
 z_3 &= x \sin\theta + y \cos\theta; \\
 v &= u_2 + z_3 u_1; \\
 \omega &= u_1;
 \end{aligned}
 \tag{2.47}$$

$$\begin{aligned}
 \dot{z}_1 &= u_1; \\
 \dot{z}_2 &= u_2; \\
 \dot{z}_3 &= z_2 u_1,
 \end{aligned}$$

де (z_2, z_3) – розташування одноколісного робота в лівій рамі, яка здійснює обертання. Рівняння (2.47) демонструє, що перетворення в ланцюговій формі не є єдиним. Цільовий апарат. Експериментальне порівняння методів управління, яке реалізується в рамках дипломного проекту, рекомендується проводити на мобільному роботі (рис. 2.14).



Рисунок 2.14 – Модель колісного робота

Опис фізичних характеристик моделі мобільного робота. Запропонована модель – це транспортний засіб із двома колесами. Ця модель включає два колеса з радіусом $r = 9,93$ см, що розташовані на одній осі довжиною $d = 29$ см. Радіус колеса включає також кільце для запобігання прокручування, виготовлене з резини, яка має достатню жорсткість для забезпечення надійного контакту з поверхнею. Центральне колесо використовується як баланс і розташоване в передній частині моделі на відстані 29 см від задньої осі. Алюмінієве шасі містить два двигуни, трансмісійні елементи, електроніку та джерела живлення. Центр мас розташований трошки ближче до задньої осі. Це економічний прототип, тому він має типові недоліки електромеханічних систем, такі як тертя та прокручування коліс, тощо. Ці обмеження впливають на продуктивність управління, але завдяки динаміці та роботі приводів модель є ефективною та придатною для подальших досліджень.

Архітектура системи управління. Алгоритми управління високого рівня можуть бути реалізовані будь-якою доступною мовою програмування для забезпечення зручного користувацького інтерфейсу в режимі реального часу. Персональний комп'ютер взаємодіє із моделлю колісного робота за допомогою радіо-модему. Низькорівневе управління забезпечує виконання команд високого рівня із достатньою швидкістю. Користувацькі алгоритми інтерполяції розроблені на персональному комп'ютері для зменшення впливу похибок квантування та затримок зв'язку при відновленні положення робота із інформації, що надається камерою. Для всіх систем управління враховано додаткову фільтрацію команд для врахування динаміки транспортного засобу та привода.

Кінематика мобільного робота. Кінематична модель визначається формулою (2.39), що еквівалентна одноколійній. Фактичні команди - це кутові швидкості ω_R та ω_L правого та лівого колеса відповідно, а не швидкість руху та кутове управління v та ω . Між цими швидкостями існує однозначне відображення.

$$v = \frac{r(\omega_R + \omega_L)}{2};$$

$$\omega = \frac{r(\omega_R + \omega_L)}{d}.$$
(2.48)

Ще була розроблена процедура калібрування для оцінки фактичних радіусів коліс й довжини осі.

Реконструкція поточної конфігурації робота базується на даних одометрії. Нехай Δ_{φ_R} та Δ_{φ_L} – це кутове зміщення коліс, що виміряне протягом вибірки T_s датчиками. Виходячи із рівняння (2.48), маємо лінійне й кутове зміщення робота:

$$\Delta_s = \frac{r}{2} (\Delta_{\varphi_R} + \Delta_{\varphi_L});$$

$$\Delta_\theta = \frac{r}{d} (\Delta_{\varphi_R} - \Delta_{\varphi_L}).$$
(2.49)

Оцінка осей на момент часу $t_k = kT_s$ обраховується наступним чином:

$$\hat{q}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{bmatrix} = \hat{q}_{k-1} + \begin{bmatrix} \cos \bar{\theta}_k & 0 \\ \sin \bar{\theta}_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta_s \\ \Delta_\theta \end{bmatrix},$$
(2.50)

де $\bar{\theta}_k = \hat{\theta}_{k-1} + \frac{\Delta_\theta}{2}$.

Локалізація робота, використовуючи запропоноване передбачення, є досить точною, якщо відсутні ефекти прокручування та люфта коліс. Проте ці явища істотно зменшуються, коли швидкість утримується на достатньо невеликому рівні, а кількість резервних маневрів обмежена.

Контрольні обмеження. У зв'язку з обмеженою швидкістю двигунів, кожне колесо може досягати максимальної кутової швидкості Ω .

З рівняння (2.48) можна отримати оцінки швидкості руху та кутового управління:

$$|v| \leq \Omega_r;$$

$$|\omega| \leq \frac{2\Omega_r}{d}.$$
(2.51)

Однак існує також жорстке обмеження через обмежену роздільну здатність у цифрового приймача. Фактично, розв'язок лінійного переміщення $\Delta_{s_{min}}$ робота має бути обраховано наступним чином:

$$\Delta_{s_{min}} = \frac{2\pi r}{m n_e n_r} = \frac{19,86\pi}{4 \cdot 200 \cdot 20} \approx 0,0039 \text{ см.}$$
(2.52)

Це значення відповідає найменш значимому біту у кодувальника, тож середня похибка квантування буде менше 200 мм. Враховуючи 8-розрядну роздільну здатність вбудованого мікроконтролера швидкості й схеми ШІМ, фактична команда лінійної швидкості має такі порогові значення і рівні насичення:

$$v_m = f_m \Delta_{s_{min}} \approx 0,78 \text{ см/с};$$
(2.53)

$$v_M = v_m \cdot (2^8 - 1) = 198,9 \text{ см/с.}$$

Щоб запобігти максимально можливе прокручування коліс, у програмному забезпеченні керування накладається більш консервативне обмеження на команди швидкостей:

$$|v| \leq v_{max} = 0,3 \text{ м/с;} \quad (2.54)$$

$$|\omega| \leq \omega_{max} = 0,5 \text{ рад/с.}$$

В даному випадку необхідно виконувати необхідне масштабування швидкості, щоб зберегти радіус кривизни, який відповідає номінальним швидкостям v та ω . Фактичні команди v_c та ω_c далі обраховуються шляхом визначення:

$$\sigma = \max \left\{ \frac{|v|}{v_{max}}, \frac{|\omega|}{\omega_{max}}, 1 \right\}, \quad (2.55)$$

та нехай $v_c = v$ та $\omega_c = \omega$, і якщо $\sigma = 1$, то:

$$\begin{aligned} \text{якщо } \sigma = \frac{|v|}{v_{max}}, \text{ то} & \quad v_c = \text{sign}(v)v_{max}; \\ \text{інакше} & \quad v_c = v/\sigma, \omega_c = \text{sign}(\omega)\omega_{max}. \end{aligned} \quad (2.56)$$

2.9 Відстеження траєкторії

Розв'язання завдання асимптотичного відстеження вимагає поєднання номінальних команд форсування та застосування оберненого зв'язку згідно з похибкою. У системах управління, що проходять вивчення, ця похибка визначається відносно опорного стану траєкторії (стану помилки).

Формування команд.

Нехай точка (x, y) одноколісного робота повинна слідувати декартовій траєкторії $(x_d(t), y_d(t))$, при $t \in [0, T]$ ($T \rightarrow \infty$), тоді із кінематичної моделі (2.39) маємо:

$$\theta = \text{ATAN2}(\dot{y}, \dot{x}) + k\pi, \text{ при } k = 0,1, \quad (2.57)$$

де ATAN2 – обернена дотична функція.

Отже, номінальні команди формування такі:

$$v_d(t) = \pm \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)}; \quad (2.58)$$

$$\omega_d(t) = \frac{\dot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)}.$$

Із диференційного рівняння (2.57) можна здійснити обрахування ω_d .

Вибраний знак для v_d буде визначати напрямок руху робота (вперед чи назад). Для точного відтворення за допомогою $v_d(t)$ та $\omega_d(t)$ необхідний декартовий рух $(x_d(t), y_d(t))$ має бути двічі диференційований в проміжку $[0, T]$.

Завдяки властивості моделі робота, яка забезпечує узгодження бажаної вихідної траєкторії $(x_d(t), y_d(t))$ разом із її похідними існує єдина зв'язка траєкторія стану $q_d(t) = (x_d(t), y_d(t), \theta_d(t))$, що може бути легко обрахована за допомогою алгебраїчних операцій, оскільки:

$$\theta_d(t) = ATAN2(\dot{y}_d(t), \dot{x}_d(t)) + k\pi, \text{ при } k = 0, 1, \quad (2.59)$$

де k вибирається таким чином, щоб $\theta_d(0) = \theta(0)$. Якщо $k = 1$, то має місце обернений рух.

Отже, номінальна орієнтація $\theta_d(t)$ може бути розрахована вздовж лінії руху (траєкторії) та використовується для визначення похибки траєкторії стану.

Також при взаємодії із запропонованою моделлю важливо мати на увазі наступне:

Коли шукана лінійна швидкість $v_d(t)$ стає нулем протягом певного моменту часу t^- , ні номінальна кутова швидкість, ні номінальна орієнтація не обчислюються з рівнянь (2.58) та (2.59) відповідно. Це може статися в початковий момент часу, якщо використовується плавний старт чи на межі з геометричним шляхом, що лежить в основі декартової траєкторії $(x_d(t), y_d(t))$. У першому випадку можна використовувати (якщо доступно) диференційну інформацію більш високого порядку про $(x_d(t), y_d(t))$ при $t=0$, щоб визначити узгоджену початкову орієнтацію та команду початкової кутової швидкості. У другому випадку неперервний рух гарантує збереження тієї ж орієнтації, що досягнута в конкретний момент часу t^- ; використовуючи аналіз у рівнянні (2.58), можна також розрахувати значення $\omega_d(t^-)$.

У більш загальному плані еталонну траєкторію можна задати шляхом роз'єднання геометричних аспектів шляху (параметризований скаляр s) із законом синхронізації $s=s(t)$, який використовується для виконання шляху. Дрейфуючий характер кінематичної моделі КМР дозволяє уникнути вищезгаданої проблеми

"нульової швидкості". Для одноколісного робота геометричне відношення може бути записане наступним чином:

$$\begin{bmatrix} \frac{dx}{ds} \\ \frac{dy}{ds} \\ \frac{d\theta}{ds} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v' + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega', \quad (2.60)$$

де команди часу відновлюються так:

$$v(t) = v'(s)\dot{s}(t); \quad (2.61)$$

$$\omega(t) = \omega'(s)\dot{s}(t).$$

Далі знаходять точки нульової швидкості з чітко визначеними геометричними дотичними. Команди прямої швидкості $v_d(s)$ та $\omega_d(s)$ в прямому напрямку обраховуються шляхом заміни у рівняннях (2.58) та (2.59), що зроблені у момент часу із похідними простору.

Конструкція лінійного управління. Найпростіший метод управління по траєкторії базується на дотичній лінеаризації здовж опорної траєкторії.

Визначення похибки відстеження траєкторії визначається як:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}. \quad (2.62)$$

Різниця між e та \tilde{q}_R знаходиться в матриці обертання, що обраховується тут за поточної орієнтації та зі зміною знаку в правій частині.

Використовуючи наступне нелінійне перетворення швидкісних входів (2.63) динаміка помилок вираховується за формулою (2.64).

$$v = v_d \cos e_3 - u_1; \quad (2.63)$$

$$\omega = \omega_d - u_2.$$

$$\dot{e} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (2.64)$$

Лінеаризуючи рівняння (2.64) навколо опорної траєкторії, маємо ті ж самі лінійні змінні, що змінні з часом, як і у рівнянні (2.42).

Закон лінійного оберненого зв'язку має вигляд:

$$\begin{aligned} u_1 &= -k_1 e_1; \\ u_2 &= -k_2 \text{sing}(v_d(t))e_2 - k_3 e_3. \end{aligned} \tag{2.65}$$

Необхідне замкнуте характеристичне рівняння виглядає так:

$$(\lambda + 2\zeta a)(\lambda^2 + 2\zeta a\lambda + a^2), \quad \zeta, a > 0, \tag{2.66}$$

а саме із сталими власними значеннями можуть бути отримані шляхом вибору коефіцієнтів підсилення у рівнянні (2.65) у вигляді:

$$\begin{aligned} k_1 = k_3 &= 2\zeta \sqrt{\omega_d^2(t) + b v_d^2(t)}; \\ k_2 &= b |v_d(t)|, \end{aligned} \tag{2.67}$$

де множник $b > 0$ було введено як додаткова степінь свободи.

Відповідно до пункту 2.5 відбувається наближення до нуля, коли локальна керованість навколо траєкторії втрачається, бо робот зупинив свій рух.

Відносно початкових управляючих входів можна стверджувати, що ця конструкція приводить до нелінійного змінного в часі управління:

$$\begin{aligned} v &= v_d \cos(\theta_d - \theta) + k_1 [\cos \theta (x_d - x) + \sin \theta (y_d - y)]; \\ \omega &= \omega_d + k_2 \text{sing}(v_d) [\cos \theta (x_d - x) + \sin \theta (y_d - y)] + k_3 (\theta_d - \theta). \end{aligned} \tag{2.68}$$

Слід відмітити, що навіть тоді, коли власні значення замкнутого циклу являються сталими та із від'ємною дійсною частиною, цей закон управління не гарантує асимптотичної стійкості похибки відстеження стану e , оскільки система все ще змінна в часі. Однак повний аналіз стійкості на основі Ляпунова може бути здійснено шляхом включення простої нелінійної модифікації, як показано нижче. Нелінійна конструкція відстеження траєкторії.

Для представлення нелінійної конструкції відстеження траєкторії необхідно розглянути рівняння (2.62) та визначити наступне:

$$u_1 = -k_1 (v_d(t), \omega_d(t)) e_1; \tag{2.69}$$

$$u_2 = -\bar{k}_2 v_d(t) \frac{\sin e_3}{e_3} e_2 - k_3(v_d(t), \omega_d(t)) e_3,$$

при $\bar{k}_2 > 0$ та додатному значення неперервності функції підсилення k_1 та k_3 .

Нехай v_d та ω_d обмежені обмежувальними похідними та $v_d(t) \neq 0$ чи $\omega_d(t) \neq 0$ при $t \rightarrow \infty$, закон управління (2.69) глобально асимптотично стабілізує початок $e = 0$.

Поєднання рівнянь (2.62), (2.63) та (2.69) маємо закон управління:

$$\begin{aligned} v &= v_d \cos(\theta_d - \theta) + k_1(v_d, \omega_d) [\cos \theta (x_d - x) + \sin \theta (y_d - y)]; \\ \omega &= \omega_d + k_2 \frac{\sin(\theta_d - \theta)}{\theta_d - \theta} [\cos \theta (x_d - x) + \sin \theta (y_d - y)] \\ &\quad + k_3(v_d, \omega_d)(\theta_d - \theta). \end{aligned} \quad (2.70)$$

Таким чином, скориставшись попереднім лінійним аналізом, можна вибрати функції підсилення k_1 та k_2 та сталі підсилення \bar{k}_2 як:

$$k_1(v_d(t), \omega_d(t)) = k_3(v_d(t), \omega_d(t)) = 2\zeta \sqrt{\omega_d^2(t) + b v_d^2(t)}; \quad (2.71)$$

$$\bar{k}_2 = b,$$

при $b > 0$ та $\zeta \in (0; 1)$.

Лінійність динамічного оберненого зв'язку.

Нелінійний контролер для відстеження траєкторії, що оснований на точній лінеаризації із динамічним оберненим зв'язком на сьогодні вже існує.

Відносно загального класу дрейфуючих систем (2.38), задача лінеаризації динамічного оберненого зв'язку полягає у тому, щоб знайти компенсатор оберненого динамічного стану форм:

$$\dot{\xi} = a(q, \xi) + b(q, \xi)u; \quad (2.72)$$

$$\omega = c(q, \xi) + d(q, \xi)u,$$

із ν -мірним станом ξ та m -мірним зовнішнім виходом u , так що замкнута система (2.38)–(2.72) еквівалентна при перетворенні стану $z = T(q, \xi)$ в лінійну керовану систему.

Для розв'язання завдання лінеаризації динамічного зворотного зв'язку наявні лише необхідні або достатні (але не одночасно необхідні і достатні) умови.

Вихідною точкою служить визначення відповідного m -мірного виходу системи $\eta = h(q)$, який дозволяє призначити бажану поведінку. Потім проводиться послідовне диференціювання виводу до тих пір, поки вхід з'явиться в не вироджений спосіб. На певному етапі може знадобитися додавання інтеграторів до підмножини вхідних каналів для уникнення подальшого диференціювання вихідних сигналів. Цей динамічний алгоритм розширення формує стан ξ динамічного компенсатора (2.72). Алгоритм завершується після фінальної кількості диференціювань кожного разу, коли система стає оберненою відносно оберненого виходу. Якщо сума порядків диференціювання виходів дорівнює розмірності $n + v$ розширеного простору станів, тоді отримуємо повну лінеаризацію вхідного стану. В такому випадку замкнута система еквівалентна безлічі розв'язаних ланцюгів введення-виведення інтеграторів від u_i до η_i ($i = 1, \dots, m$).

Давайте проілюструємо цю точну процедуру лінеаризації на прикладі моделі одноколісного робота (2.39). Нам потрібно визначити лінеаризований вихідний вектор для $\eta = (x, y)$:

$$\dot{\eta} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (2.73)$$

У рівнянні (2.73) відображається, що тільки v впливає на $\dot{\eta}$, а кутова швидкість ω не може бути відновлена з цієї диференціальної інформацією першого порядку. Для продовження потрібно додати інтегратор (стан якого позначається ξ) на лінійному вході швидкості:

$$v = \xi; \quad (2.74)$$

$$\dot{\xi} = a \Rightarrow \dot{\eta} = \xi \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}.$$

Наступний крок – це диференціювання:

$$\ddot{\eta} = \dot{\xi} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} + \xi \dot{\theta} \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\xi \sin \theta \\ \sin \theta & \xi \cos \theta \end{bmatrix} \begin{bmatrix} a \\ \omega \end{bmatrix}. \quad (2.75)$$

За умови вхідних (a, ω) й при $\xi \neq 0$:

$$\begin{bmatrix} a \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & -\xi \sin \theta \\ \sin \theta & \xi \cos \theta \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (2.76)$$

щоб отримати наступне:

$$\ddot{\eta} = \begin{bmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = u, \quad (2.77)$$

Як результат ми маємо динамічний компенсатор:

$$\begin{aligned} \dot{\xi} &= u_1 \cos \theta + u_2 \sin \theta; \\ v &= \xi; \\ \omega &= \frac{u_2 \sin \theta - u_1 \cos \theta}{\xi}. \end{aligned} \quad (2.78)$$

Оскільки динамічний компенсатор є саме одновимірним, ми маємо $n + \nu = 3 + 1 = 4$, що дорівнює загальному числу вихідних диференціювань у рівнянні (2.77). Тому на нових координатах:

$$\begin{aligned} z_1 &= x; \\ z_2 &= y; \\ z_3 &= \dot{x} = \xi \cos \theta; \\ z_4 &= \dot{y} = \xi \sin \theta. \end{aligned} \quad (2.79)$$

Розширена система повністю лінеаризується до керованої форми й описується двома ланцюгами інтеграторів введення-виведення другого ступеня, заданими рівнянням (2.77):

$$\begin{aligned} \ddot{z}_1 &= u_1; \\ \ddot{z}_2 &= u_2. \end{aligned} \quad (2.80)$$

Зауважимо, що контролер лінеаризації із динамічним зворотним зв'язком (2.78) має потенційну особливість при $\xi = v = 0$. Коли одноколісний робот не котиться. Показано, що саме виникнення такої особливості в динамічному процесі розширення є структурним для систем керування, що розглядаються. Ця проблема повинна бути врахована при розробці законів управління еквівалентної лінійної моделі.

Припустимо, робот повинен слідувати гладкою траєкторією виведення $(x_d(t), y_d(t))$. Такий номінальний керуючий вхід $v_d = \sqrt{(\dot{x}_d^2 + \dot{y}_d^2)}$ уздовж

траєкторії ніколи не наближається до нуля. На еквівалентній лінійній й розв'язуваній системі (2.79) неважко розробити глобальний експоненціальний стабілізуючий зворотний зв'язок для шуканої траєкторії (з лінійними декартовими перехідними процесами), так що:

$$\begin{aligned} u_1 &= \ddot{x}_d(t) + k_{p1}(x_d(t) - x) + k_{d1}(\dot{x}_d(t) - \dot{x}); \\ u_2 &= \ddot{y}_d(t) + k_{p2}(y_d(t) - y) + k_{d2}(\dot{y}_d(t) - \dot{y}), \end{aligned} \quad (2.81)$$

із коефіцієнтами підсилення, які обираються як $k_{pi} > 0, k_{di} > 0$, для $i = 1, 2$.

У реалізації швидкості \dot{x} та \dot{y} можуть бути визначені через два останні вирази в рівнянні (2.79), залежно від стану робота та стану компенсатора ξ . В якості альтернативи можна використовувати оцінки \dot{x} та \dot{y} , отримані з одометричних вимірювань. Це рішення є більш стійким відносно динаміки.

2.10 Математична модель колісного робота

У цьому дослідженні обрано мобільний робот з диференціальним приводом як об'єкт управління. Маючи два сервомотора, робот з диференціальним приводом може змінювати напрямок руху за рахунок різних швидкостей. Для прямолінійного руху необхідно, щоб колеса оберталися з однаковими швидкостями, а для повороту робота на місці - з однаковими за модулем, але протилежними за напрямком. Інші комбінації швидкостей ведуть до руху вздовж кривої.

Система представляє собою два абсолютно твердих диска, розташованих на осях, які лежать на одній прямій. На місцях кріплення коліс до осей розташовані точкові циліндричні шарніри, а колеса управляються сервомоторами. Корпус, що жорстко прикріплений до осей, є абсолютно твердим тілом, яке може рухатися в плоскопаралельному напрямку. Корпус рухається по абсолютно шорсткій площині, причому колеса не знаходяться в контакті з площиною в точках дотику. Модель робота показана на рисунку 2.16. При умові плоскопаралельного руху корпусу положення системи описується п'ятьма координатами $(x, y, \Theta, \varphi_1, \varphi_2)$, де кути (φ_1, φ_2) представляють собою кути повороту ведучих коліс робота відносно осей. Центр мас корпусу системи розташований в точці C . Положення центру мас b пов'язаної системи координат задається вектором $b = (b_1, b_2)$. Середина відстані між колесами позначена $D(x, y)$. Вплив пасивних коліс на рух системи розглядається як незначний.

Нехай початковий стан системи задано набором, кінцеве становище визначається набором $\varphi(x_0, y_0, \Theta_0, ((\varphi_1)))$. Припустимо, що під час руху робота координати (x, y, Θ) разом із кутовими швидкостями обертання коліс є

неперервними, і $|\dot{\varphi}_2| \leq \max$. Розглядається задача переходу робота з заданого початкового стану в заданий кінцевий стан, з обов'язковим переходом робота в задану кінцеву точку із визначеною орієнтацією поздовжньої осі та визначеною швидкістю уздовж цієї поздовжньої осі.

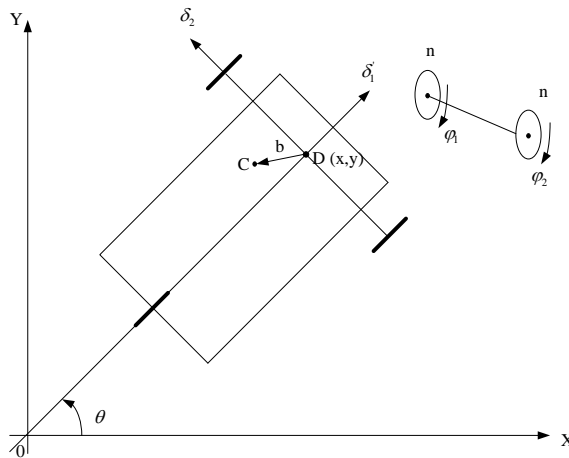


Рисунок 2.16 – Модель мобільного робота з диференційним приводом

Із умови не проковзування коліс на площині виводяться рівняння зв'язків:

$$\begin{cases} \dot{x} = (a\dot{\theta} + r\dot{\varphi}_1)\cos(\theta); \\ \dot{y} = (a\dot{\theta} + r\dot{\varphi}_1)\sin(\theta); \\ \dot{\varphi}_2 = \dot{\varphi}_1 + \frac{2a}{r}\dot{\theta}. \end{cases} \quad (2.82)$$

Вже згадана система має 2 ступені свободи, в якості незалежних координат виберемо (φ_1, θ) . Рівняння руху системи записуються у вигляді рівнянь Чаплигіна.

$$\begin{cases} r^2(m + m_1)\ddot{\varphi}_1 + ((m + m_1)ar - m_0b_2r)\ddot{\theta} - m_0b_1r\dot{\theta}^2 = Q_1^* \\ ((m + m_1)ar - m_0b_2r)\ddot{\varphi}_1 + ((m + 2m_1)a^2 - 2m_0b_2a + J)\ddot{\theta} + m_0b_1r\dot{\varphi}_1 = Q_2^* \end{cases}; \quad (2.83)$$

$$\begin{cases} Q_1^* = v_1u_1 + v_2u_2 - (\mu_1 + \mu_2)\dot{\varphi}_1 - \frac{2a}{r}\mu_2\dot{\theta} \\ Q_2^* = \frac{2a}{r}\left(v_2u_2 - \mu_2\dot{\varphi}_1 - \frac{2a}{r}\mu_2\dot{\theta}\right) \end{cases}; \quad (2.84)$$

де m_0 – маса корпусу робота, m_1 – маса колесу робота, m – повна маса робота, J – приведений момент інерції робота, залежить від моментів інерції корпусу і коліс робота і інших динамічних параметрів робота.

У правій частині (2.83) включені загальні сили (2.84), що характеризують вплив моментів, породжених двигунами постійного струму, на активні колеса. У (2.84) коефіцієнти $\nu_1, \nu_2, \mu_1, \mu_2$ – це коефіцієнти лінійних моделей електродвигунів робота. Нехай базовими траєкторіями руху робота є дуги кола і відрізки прямих. Підґрунтям для введення додаткової кривої є той факт, що при відсутності такого "зчеплення" двох траєкторій руху в точці їх з'єднання відбувається розрив у швидкостях. Це призводить до невизначеності у керуючих напругах, як впливає з (2.83), і можливості динамічних ударів у системі. Узагальнено кажучи, розрив швидкостей виникає при переході з кривої на криву, на якому відбувається стрибок функції орієнтованої кривизни траєкторії:

$$k_0(s) = \frac{\left\| \left[\dot{\vec{r}}(s), \ddot{\vec{r}}(s) \right] \right\|}{\left(\dot{\vec{r}}(s), \dot{\vec{r}}(s) \right)^{3/2}};$$

звідки

$$k_0 = \frac{\dot{\theta}}{v} = \frac{r(\dot{\phi}_2 - \dot{\phi}_1)}{2av};$$

де $v = r \frac{\dot{\phi}_1 + \dot{\phi}_2}{2}$ – лінійна швидкість корпусу робота.

Для того, щоб виключити по $\dot{\phi}_1, \dot{\phi}_2$ вводиться додатковий режим руху системи по інтервалу "склеювання" $[T_1, T_2]$, при якому кутові швидкості коліс змінюються лінійно. Відповідна схема приведена в рис. 2.16.

Відповідна інтервалу $[T_1, T_2]$ крива на координатах (x, y) виходить підстановкою лінійного закону зміни $\dot{\phi}_1, \dot{\phi}_2$ у перше й друге рівняння зв'язків (2.82):

$$\begin{aligned} x(t) &= x_0 + a \frac{F^+}{F^-} (\sin(\theta_0) - \sin(\theta(t))) + \alpha_1 \cdot (C(\beta_1) - C(\beta_2)) - \alpha_2 \cdot (S(\beta_1) - S(\beta_2)); \\ y(t) &= y_0 + \alpha \frac{F^+}{F^-} (\cos(\theta(t)) - \cos(\theta_0)) + \operatorname{sgn}(F^-) \cdot \alpha_1 \cdot (S(\beta_1) - S(\beta_2)) + \\ &\quad \operatorname{sgn}(F^-) \cdot \alpha_2 \cdot (C(\beta_1) - C(\beta_2)); \\ \theta(t) &= \frac{rF^-}{4aT} t^2 + \frac{r}{2a} (\dot{\phi}_2^0 - \dot{\phi}_1^0)t + \theta_0; \end{aligned} \quad (2.85)$$

де $a, F^-, F^+, \theta_0, \alpha_1, \alpha_2, \beta_1, \beta_2$ – константи або відомі функції часу, $C(x)$ та $S(x)$ – інтеграли Френеля. Що виходять крива - узагальнена спіраль Корню.

Приклади синтезу траєкторій руху робота, які складаються із двох кіл, з'єднаних спіральною кривою (рис. 2.17).

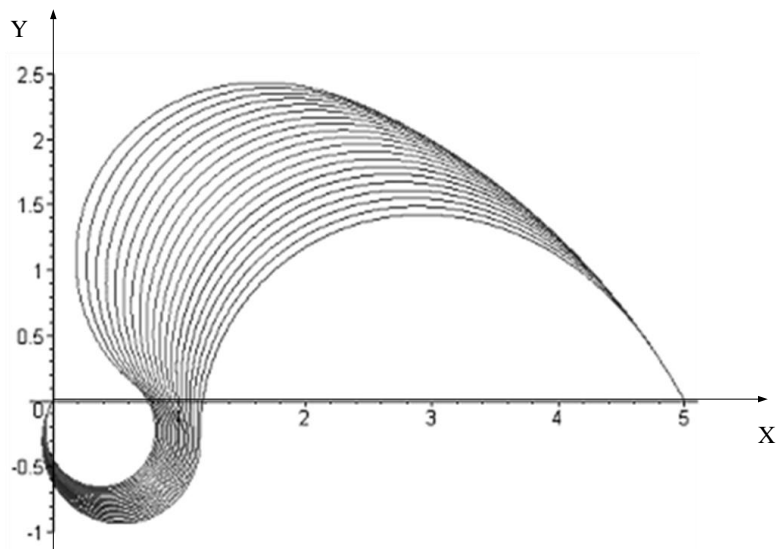


Рисунок 2.17 – Приклад синтезованих траєкторії роботів

Якщо припустити, що роботу відоме місце розташування, то планування маршруту на основі його розташування дозволить йому рухатися по точному шляху вздовж послідовності передбачених точок. Різні криволінійні траєкторії можуть бути розроблені, з оптимізацією таких критеріїв, як час руху по маршруту або енергоспоживання. Припустимо, що рух робота складається з прямих відрізків окремо від поворотів на місці; робот прагне мінімізувати загальну відстань, так що він завжди відразу повертається обличчям до наступної точки і їде прямо до неї.

Як інструмент для оцінки стану колісного робота використовується одометр. Звичайною схемою одометра робота є використання енкодерів, що вимірюють кут повороту коліс (рис. 2.18).

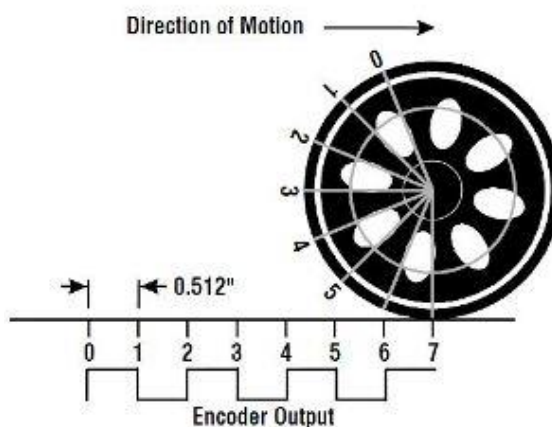


Рисунок 2.18 – Стандартна схема одометрії робота

Довжина кола колеса обраховується по формулі:

$$C = 2\pi R = \pi \cdot D \quad (2.86)$$

де R – радіус колеса, D – діаметр колеса.

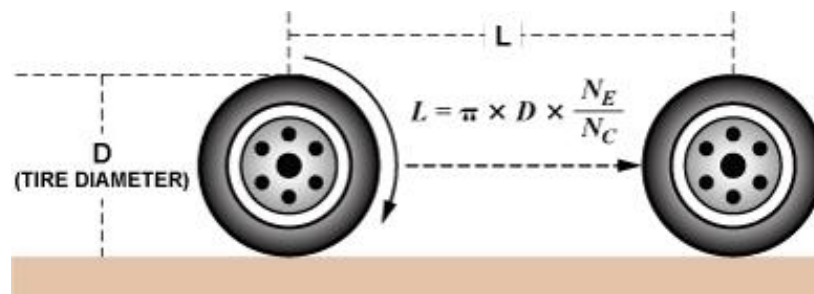


Рисунок 2.18 – Переміщення робота

За один оборот колеса, робот зсувається на відстань C . Пройдену відстань можна обчислити за формулою:

$$L = \frac{n}{N} \cdot C = \frac{n}{N} \cdot \pi D, \quad (2.87)$$

де: R – радіус колеса (D – діаметр колеса); L – підсумкове пройдену відстань (за заданий проміжок часу); N – кількість відліків енкодера за один оборот колеса; n – сумарна кількість відліків енкодера (за заданий проміжок часу).

Положення центру O робота в момент часу $t + 1$ буде становити (в радіанах):

$$O(t+1) = O(t) + (D_r - D_l)/W; \quad (2.88)$$

де: $O(t)$ – положення робота в момент часу t ; D_r – відстань, яка пройдена колесом робота; D_l – відстань, пройдене лівим колесом робота; W – ширина роботу.

Відстань, яка пройдена за цей проміжок часу можна оцінити так:

$$D(t,t+1) = (D_r + D_l)/2. \quad (2.89)$$

Для відображення координат робота на мапі можна розрахувати його Декартові координати:

$$\begin{aligned} X(t+1) &= X(t) + D(t,t+1) \cdot \cos(O(t+1)); \\ Y(t+1) &= Y(t) + D(t,t+1) \cdot \sin(O(t+1)). \end{aligned} \quad (2.90)$$

Але важливо враховувати, що при розгляді більших проміжків часу, накопичення похибок призводить до збільшення відхилення в орієнтації, що зростає пропорційно пройденому роботом шляху.

Джерела невірності включають:

- відхилення у вимірах радіуса коліс;
- помилки в обчисленні імпульсів від енкoderів;
- обмежена частота обробки одометрії.

Недостатня частота обробки даних від одометрії може призвести до такої помилки (рис. 2.20). Ліворуч представлений випадок, коли робот рухається по прямій лінії, і значення енкoderів рівні; справа - коли робот об'їжджає перешкоду. Лівий енкoder показує 80, а правий - 40, що вказує на правий поворот. Але після цього робот повертається вліво, і в кінцевому положенні енкодери вказують однакові значення (вище, ніж на лівому зображенні, оскільки робот пройшов більший шлях, слідуючи S-подібної кривої).

Якщо програмне забезпечення робота фіксує значення датчиків лише в кінцевому положенні, то природно припускати, що робот рухався по прямій лінії. Таким чином, недостатня частота обробки даних від енкoderів може призвести до похибок.

Для визначення напрямку руху колісного робота при оминанні перешкод необхідно аналізувати інформацію, отриману від датчиків, у цьому випадку - відеокамери. Для обробки зображень найбільш універсальним підходом є метод оптичних потоків.

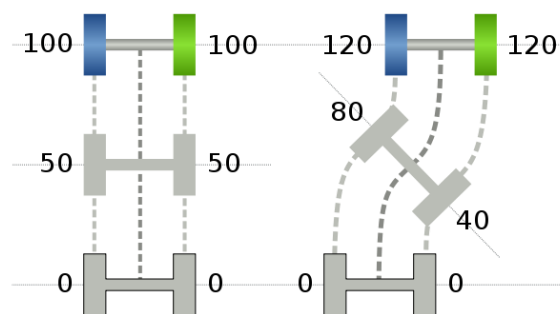


Рисунок 2.19 – Модель переміщення методів оптичних потоків

Спочатку важливо відзначити, що в даний час дуже розповсюджений метод аналізу оптичних потоків, який забезпечує ефективний та гнучкий засіб для вивчення руху об'єктів на цифрових відео-послідовностях.

Однією з ключових проблем, пов'язаних із використанням методу аналізу оптичних потоків у практичних застосуваннях, є велика чутливість цього методу до перешкод. Це впливає з того, що, на відміну від інтегральних алгоритмів аналізу зображення, які базуються на накопиченні інформації про сигнал протягом певного часу, метод оптичних потоків є, по суті, диференціальним методом. Він ґрунтується на порівнянні лише двох послідовних кадрів відео-потoku та обчисленні відповідних просторових і часових похідних в точках зображення. Відомо, що процедури диференціювання та фільтрації завжди більш чутливі до шумів та різних видів перешкод, в порівнянні з процедурами

накопичення даних, які використовуються для приглушення таких шумів та перешкод.

Мета полягає в розробці алгоритму виявлення та відстеження рухомих об'єктів, заснованого на модифікованому методі аналізу оптичних потоків, який забезпечує стійкість на високому рівні.

У методі оптичних потоків для виявлення перешкод виконуються такі операції на кожному кадрі відео-послідовності:

1. Породження різниці між накопиченими зображеннями.
2. Обчислення оптичного потоку на основі різниці між накопиченими зображеннями.
3. Визначення перешкод на зображенні за допомогою оцінки накопиченого оптичного потоку.

Побудова оптичного потоку традиційно вважається процедурою оцінки яскравіших та геометричних змін між поточним та попереднім кадрами. Рух об'єктів перед нерухомою камерою, так само як і рух камери в навколишньому середовищі, призводять до відповідних змін на зображенні. Рух видимого поля, спостережуваного при русі камери щодо зображуваних об'єктів або об'єктів щодо камери, називається оптичним потоком. Необхідно визначити поле руху, призначивши кожній точці зображення вектор швидкості. В певний момент часу точка P_i на зображенні рисунку 2.20 відповідає певній точці P_0 на поверхні об'єкту. Ці дві точки пов'язані рівняннями проєкції. Точка об'єкта P_0 переміщується відносно камери із швидкістю v_0 . Це призводить до руху v_i відповідної точки зображення P_i . За час δt точка P_0 зміщується на відстань $v_0 \delta t$, а її зображення P_i - на відстань $v_i \delta t$.

Яскравості руху відбуваються разом із об'єктами. Оптичний потік, як вже зазначалося, представляє рух яскравішого зображення. У ідеалі оптичний потік відповідає певному полю руху, але на практиці це не завжди так.

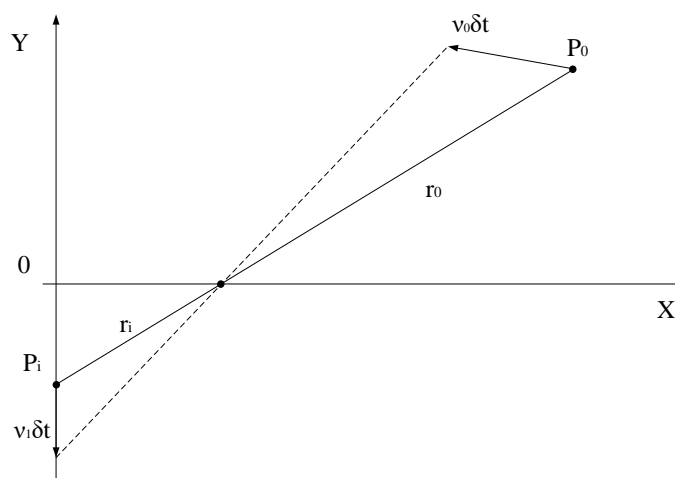


Рисунок 2.20 – Переміщення точки

Нехай тепер $I(x, y, t)$ – це яскравість пікселя в точці зображення (x, y) на момент часу t . Тоді, якщо $u(x, y)$ й $v(x, y)$ – x та y компоненти вектору оптичного потоку в цій точці, то можна очікувати, що для малого інтервалу часу δt :

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t). \quad (2.91)$$

Припустивши, що яскравість змінюється плавно по x, y, t , можна розкласти ліву частину рівняння в ряд Тейлора і звідси отримати наступне:

$$I(x, y, t) + \delta x \cdot \frac{\partial I}{\partial x} + \delta y \cdot \frac{\partial I}{\partial y} + \delta t \cdot \frac{\partial I}{\partial t} + o(\delta x, \delta y, \delta t) = I(x, y, t); \quad (2.92)$$

$$\frac{\partial I}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial I}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0. \quad (2.93)$$

Співвідношення можна записати у наступному вигляді:

$$\frac{dI}{dt} = 0;$$

де в лівій частині стоїть повна похідна I по часу. Ввівши скорочення $I_x = \delta I / \delta x$, $I_y = \delta I / \delta y$, $I_t = \delta I / \delta t$, також враховуючи, що $u = dx/dt$ и $v = dy/dt$:

$$I_x u + I_y v + I_t = 0. \quad (2.94)$$

Похідні I_x, I_y, I_t отримуємо із зображення за допомогою числових апроксимацій похідних кінцевими різницями.

Формула набуває вигляду:

$$\nabla I(x, y, t) \cdot (u, v)^T + I_t(x, y, t) = 0. \quad (2.95)$$

$$\iint_D (\nabla I \cdot (u, v)^T + I_t)^2 + (\lambda^2 (\|\nabla u\|^2 + \|\nabla v\|^2)) dx dy = 0. \quad (2.96)$$

Тут область D – область, в котрій відбувається пошук оптичного потоку. Значення коефіцієнта λ визначає саме рівень значимості згладжування частини функціоналу.

Послідовність швидкостей, що мінімізує функціонал (2.96) (u^m, v^m) має вигляд:

$$\begin{aligned}
 u^{m+1} &= \bar{u}^m - \frac{I_x \cdot [I_x \bar{u}^m + I_y \bar{v}^m + I_t]}{\alpha^2 + I_x^2 + I_y^2}; \\
 v^{m+1} &= \bar{v}^m - \frac{I_x \cdot [I_x \bar{u}^m + I_y \bar{v}^m + I_t]}{\alpha^2 + I_x^2 + I_y^2}.
 \end{aligned}
 \tag{2.97}$$

Якщо за область D приймати увесь кадр аналізу та крок сітки вибрати рівним 1 пікселю по горизонталі й вертикалі, то u^m й v^m будуть матрицями розміру $W \times H$. В даних ітераційних рівняннях \bar{u}^m та \bar{v}^m – швидкості, які усереднені по сусіднім точкам:

$$\begin{aligned}
 \bar{u}_{i,j}^m &= \frac{1}{4} (u_{i+\delta x,j}^m + u_{i,j+\delta y}^m + u_{i-\delta x,j}^m + u_{i,j-\delta y}^m); \\
 \bar{v}_{i,j}^m &= \frac{1}{4} (v_{i+\delta x,j}^m + v_{i,j+\delta y}^m + v_{i-\delta x,j}^m + v_{i,j-\delta y}^m).
 \end{aligned}
 \tag{2.98}$$

Тут індекс m показує номер поточної ітерації, i та j – індекси поточного вузла сітки.

Ітераційний процес закінчується, коли невязка (2.99) між двома послідовними ітераціями буде менше наперед заданого числа μ :

$$\sum_{i=1}^W \sum_{j=1}^H \left\{ \|u_{i,j}^{m+1} - u_{i,j}^m\|^2 + \|v_{i,j}^{m+1} - v_{i,j}^m\|^2 \right\} \leq \mu.
 \tag{2.99}$$

Однак це умову досить не зручно використовувати явно через значні обчислювальні витрати на її обчислення та перевірку на кожній ітерації. На практиці для зображень з високою контрастністю об'єкта вистачає 10-15 ітерацій. Застосування значно більшої кількості ітерацій може викликати появу неправильних ненульових швидкостей в областях, де поле швидкостей фактично рівне нулю. Це може статися, наприклад, коли два різних об'єкти рухаються на малих відстанях один від одного. Поле швидкостей між ними насправді є нульовим, але оптичний потік, обчислений при великій кількості ітерацій, може бути ненульовим через припущення про неперервність руху.

Оптичний потік також може дорівнювати нулю там, де поле швидкостей не є нульовим. Такий випадок, наприклад, може виникнути при руху об'єкта, який має постійну яскравість на всій площі зображення. У цьому випадку оптичний потік, обчислений на межі об'єкта, буде ненульовим, тоді як обчислений в центрі об'єкта буде близьким до нуля, хоча справжнє поле швидкостей повинно бути однаковим на всій поверхні об'єкта. Ця проблема відома як "проблема апертури".

Для оцінки похідних яскравості зображення пропонується розглядати наступним чином:

$$\begin{aligned}
 I_x &\approx \frac{1}{4} \delta x \left[\begin{aligned} &(I_{i+\delta x, j}^k + I_{i+\delta x, j+\delta y}^k + I_{i+\delta x, j}^{k+1} + I_{i+\delta x, j+\delta y}^{k+1}) - \\ &-(I_{i, j}^k + I_{i, j+\delta y}^k + I_{i, j}^{k+1} + I_{i, j+\delta y}^{k+1}) \end{aligned} \right]; \\
 I_y &\approx \frac{1}{4} \delta y \left[\begin{aligned} &(I_{i, j+\delta y}^k + I_{i+\delta x, j+\delta y}^k + I_{i, j+\delta y}^{k+1} + I_{i+\delta x, j+\delta y}^{k+1}) - \\ &-(I_{i, j}^k + I_{i+\delta x, j}^k + I_{i, j}^{k+1} + I_{i+\delta x, j}^{k+1}) \end{aligned} \right]; \\
 I_t &\approx \frac{1}{4} \left[\begin{aligned} &(I_{i, j}^{k+1} + I_{i, j+\delta y}^{k+1} + I_{i+\delta x, j}^{k+1} + I_{i+\delta x, j+\delta y}^{k+1}) - \\ &-(I_{i, j}^k + I_{i, j+\delta y}^k + I_{i+\delta x, j}^k + I_{i+\delta x, j+\delta y}^k) \end{aligned} \right].
 \end{aligned} \tag{2.100}$$

Тут використовується мережева апроксимація похідних. Індекс k вказує на номер поточного кадру, а i та j - індекси поточного вузла мережі. Варіації δx та δy при обчисленні приватних похідних можна вибрати за бажанням. Зазвичай використовується мережа з параметрами $\delta x = \delta y = h$.

Тепер, вставивши приватні похідні та середні швидкості в ітераційний процес (2.98) з початковими умовами $u_{i, j}^0 = 0, v_{i, j}^0 = 0$ для всіх i та j в області D , можна легко знайти швидкості всіх точок мережі на кожному спостережуваному кадрі відео-послідовності.

Описана обчислювальна схема слід за традиційними методами оцінки оптичного потоку. Однак експерименти на великому обсязі реальних відеозаписів показали, що якість вихідних даних подібних алгоритмів, які працюють безпосередньо з вихідними цифровими напів-тоновими зображеннями, недостатньо висока через значний вплив шуму та інших перешкод на якість детектування об'єктів. У зв'язку з цим пропонується використовувати спеціальну процедуру перед обробкою відеоданих, таку як різницево-накопичувальна процедура. Суть цієї процедури полягає в попередньому виділенні контурів рухомих об'єктів, за якими потім обчислюється оцінка оптичних потоків, яка використовується на етапі формування гіпотез і простеження рухомих об'єктів.

Для застосування на етапі попередньої обробки зображення в модифікованому методі оптичних потоків пропонується використовувати процедуру "тимчасової нормалізації фону", засновану на порівнянні двох накопичених в часі зображень (що ковзають зважених сум) з різними періодами накопичення/забування даних, що охоплюються акумулятором 1 і акумулятором 2 відповідно.

При появі першого кадру відео-послідовності він записується в обидва акумулятора. При появі кожного нового кадру викликається обробник подій, який запускає процедуру оновлення значень акумулятора 1 і акумулятора 2. Акумулятор 1 і акумулятор 2 мають ті ж розміри, що і вхідне зображення.

Позначимо ширину зображення буквою W , а висоту - буквою H . Порівняємо акумулятор 1 матрицю M_1 , акумулятор 2 – матрицю M_2 , вхідне зображення – матрицю Q . Кожен елемент даних матриць логічно відповідає яскравості пікселя і тому приймає значення від 0 до 255. У k -тий момент часу накопичені матриці приймають значення $M_1(k)$ і $M_2(k)$.

Алгоритм оновлення значень акумулятора 1 і акумулятора 2 на кожному $k + 1$ -му кадрі працює за наступною схемою:

$$\begin{aligned} M_1(k+1) &= p_1 \cdot M_1(k) + q_1 \cdot Q(k); \\ M_2(k+1) &= p_2 \cdot M_2(k) + q_2 \cdot Q(k). \end{aligned} \quad (2.101)$$

Тут коефіцієнти p_1, q_1, p_2, q_2 – це параметри тимчасового усереднення, що визначаються, виходячи з того, скільки кадрів необхідно «пам'ятати» даного акумулятора. При цьому повинні виконуватися для параметрів усереднення природні умови нормування $p_1 + q_1 = 1, p_2 + q_2 = 1$.

Перетворивши вираз (2.101), щоб обчислити $M_1(k+1)$ і $M_2(k+1)$ через суму вхідних даних. Для цього необхідно ввести початкові умови $M_1(0) = M_2(0) = Q(0)$. Складаючи всі вхідні кадри до k -го включно, отримуємо:

$$\begin{aligned} M_1(k+1) &= q_1 \cdot \sum_{i=0}^k p_1^{k-i} G(i); \\ M_2(k+1) &= q_2 \cdot \sum_{i=0}^k p_2^{k-i} G(i). \end{aligned} \quad (2.102)$$

Тепер припустимо, що необхідно пам'ятати n_1 кадрів в першому акумуляторі і n_2 – у другому. Це означає, що внесок $(n_1 + 1)$ -го кадру в першу суму, а $(n_2 + 1)$ -го – в другу повинен бути менше деякого заздалегідь заданого ε . Виходячи з цього, умови «забування» $(n_1 + 1)$ і $(n_2 + 1)$ кадрів будуть виглядати наступним чином:

$$\begin{aligned} q_1 p_1^{n_1} G(n_1) &< \varepsilon; \\ q_2 p_2^{n_2} G(n_2) &< \varepsilon. \end{aligned} \quad (2.103)$$

Оскільки відносно можливих значень $G(n_1)$ та $G(n_2)$ відомий тільки допустимий діапазон, що використовується наступною нерівністю:

$$\begin{aligned} q_1 p_1^{n_1} G(n_1) &< 255 \cdot q_1 p_1^{n_1} < \varepsilon; \\ q_2 p_2^{n_2} G(n_2) &< 255 \cdot q_2 p_2^{n_2} < \varepsilon. \end{aligned} \quad (2.104)$$

Звідси:

$$\begin{aligned} q_1 p_1^{n_1} &< \frac{\varepsilon}{255} = \tilde{\varepsilon}; \\ q_2 p_2^{n_2} &< \frac{\varepsilon}{255} = \tilde{\varepsilon}. \end{aligned} \quad (2.105)$$

Враховуючи умови нормування $q_1 = 1 - p_1$ та $q_2 = 1 - p_2$, остаточно маємо:

$$\begin{aligned} p_1^{n_1} - p_1^{n_1+1} &= \tilde{\varepsilon}; \\ p_2^{n_2} - p_2^{n_2+1} &= \tilde{\varepsilon}. \end{aligned} \quad (2.106)$$

Тепер можна вирішити рівняння (2.106) щодо p_1 і p_2 будь-яким зручним чисельним методом, доберемо метод дихотомії або метод дотичних (так як цільова функція неперервно диференційована).

При деяких n_i , $i = 1, 2$ виявляється, що для будь-яких p_i вираз $q_i p_i^{n_i}$ буде завжди менше фіксованого $\tilde{\varepsilon}$. Це означає, що для кожного ε існує межа n_i кадрів, наявних в акумуляторі, таких, що для будь-яких p_i і q_i кожен наступний кадр буде «забуватися», для будь-яких p_i , q_i і $G(n)$ завжди буде виконуватися $q_i p_i^{n_i} G(n_i) < \varepsilon$.

Отже, в результаті накопичення отримали дві матриці $M_1(k)$, $M_2(k)$ та формуємо їх різницю $\bar{M}(k)$. При використанні різноманітних комбінацій n_1 та n_2 дана різниця має різний зміст. При великих $n_1 \in [300, 450]$ та $n_2 \approx n_1/2$ $\bar{M}(k)$ може бути використана для виділення новопосталих або, навпаки, зниклих об'єктів сцени спостереження (завдання «детектування залишених предметів», часто виникає в області відеоспостереження). При менших значеннях $n_1 \in [3, 7]$ та $n_2 \in [1, 3]$ різниця $\bar{M}(k)$ містить злегка розмиті (усереднені по декільких кадрах) контури рухомих об'єктів, які зручно використовувати для оцінки руху і межкадрового простеження об'єктів. Таким чином, в модифікованому методі оптичних потоків замість яскравості і поточного зображення в якості вхідного двовимірного розподілу сигналу для оцінки оптичного потоку пропонується використовувати різницю накопичених зображень $\bar{M}(k)$.

Без суттєвої втрати загальності можна розглядати, наприклад, квадратні піксельні "блоки" розміру $w \times w$ як первинні рухомі елементи зображення. Далі, піксельний квадрат розміру $w \times w$ вважається "рухомим", якщо сума модулів швидкостей точок сітки, що входять в квадрат, перевищує деяке наперед задане число. Це число можна підібрати емпірично, наприклад, розрахувавши медіану модулів швидкостей точок протягом кількох початкових кадрів. Лінійні усереднені оцінки в цьому випадку не можна використовувати через те, що при спостереженні на відеопослідовності лише невеликих ділянок ненульового

оптичного потоку обчислений лінійно усереднений модуль швидкості значно відрізняється в меншу сторону від фактичної середньої швидкості рухомих об'єктів на даній відео послідовності.

При розбитті зображення на елементарні осередки (квадрати) поверх сітки оптичного потоку утворюється більш розріджена підсітка "елементарних квадратів". Розмір елементарного квадрата в пікселях дорівнює добутку розміру сітки на стороні квадрата. Сторону квадрата підбирають емпірично в залежності від очікуваного розміру рухомих об'єктів: необхідно, щоб очікуваний об'єкт, який рухається мінімального розміру, містив не менше 2-4 елементарних квадратів.

Позначимо матрицю ідентифікаторів руху квадратів як R . Кожен елемент цієї матриці приймає значення 0 або 1 в залежності від того, чи було визначено квадрат як рухливий чи ні. Для первинного формування областей, які можуть бути вважати єдиним об'єктом, що рухається, використовується наступний алгоритм.

Перший прохід: Оптичний потік добре визначається на кордонах об'єкта, але для однотонного об'єкта може дорівнювати нулю в центрі об'єкта. Тобто можливі випадки, коли деякі сусідні квадрати $R_{i,j}$ вважаються рухливими, а сам квадрат $R_{i,j}$ вважається нерухомим. Рішення цієї проблеми полягає в присвоєнні значення 1 нерухомому квадрату $R_{i,j}$ при проходженні матриці R за стовпцями, якщо K квадратів-сусідів вважаються рухливими. Значення K обирається емпірично, і оптимальні значення для K виявилися $K = 4$ і $K = 5$ при роботі з різними відеопослідовностями.

Другий підхід: Переглядаючи матрицю R стовпець за стовпцем, присвоюється нульове значення $Q_{i,j} = k$ кожному рухому квадрату $R_{i,j}$. Тут Q - це матриця номерів рухливих квадратів, така ж розмірність як і матриця ідентифікаторів руху R . Величина k вибирається відповідно до номерів сусідніх пікселів за маскою. Маска включає сусідів для порівняння, і в разі 4-х зв'язності це сусіди $R_{i-1,j}$ і $R_{i,j-1}$, а в разі 8-ми зв'язності - $R_{i,j-1}$, $R_{i-1,j-1}$, R_{i-1} та $R_{i-1,j+1}$. Можливі дії в різних випадках:

- якщо всі сусіди - нерухомі квадрати, то рухаючомуся квадрату $R_{i,j}$ присвоюється новий, на даний момент, невикористаний номер;
- якщо є точно один сусідній рухливий квадрат з ненульовим номером, присвоюємо цей номер рухаючомуся квадрату $R_{i,j}$;
- якщо є більше одного рухомого квадрата серед сусідів, присвоюється номер будь-якого з вже пронумерованих рухомих квадратів. Якщо номери сусідів різняться, відбувається зіткнення номерів - зберігаються номери пари як еквівалентні. Отримані пари зберігаються в окремій структурі даних - таблиці еквівалентності.

Третій прохід: Всі рухомі квадрати вже пронумеровані під час першого проходу, але деякі області мають квадрати з різними номерами через зіткнення. Матрицю R проглядають знову, і рухомі квадрати перенумеровуються з

використанням інформації про еквівалентні номери з таблиці еквівалентності. На виході отримується індексна матриця Q номерів рухливих квадратів.

2.11 Висновки за розділом

Для розв'язання завдання навігації роботизованою платформою необхідно провести ідентифікацію об'єктів орієнтації, включаючи QR-мітки. Для визначення мітки на зображенні в першу чергу потрібно підготувати його до використання автоматизованими алгоритмами СКЗ. Під час процесу розпізнавання використовуються алгоритми фільтрації та трансформації зображення, а також різноманітні преобразування для отримання шуканого об'єкта. Знаходження об'єкта на кадрі з камери, що обладнана роботизованою платформою, відкриває можливість визначення відстані до об'єкта і, таким чином, сприяє уникненню зіткнень з будь-якими перешкодами. Застосовуючи алгоритм пошуку SURF, можна передбачати та фіксувати місце фінішу. Знаючи точки старту та фінішу траєкторії, можна за допомогою алгоритму Дейкстри опрацювати всі маневри та знайти мінімально короткий шлях від початку до кінця. Це сприяє оптимізації переміщень автоматизованих роботизованих платформ у різних середовищах.

Під час виконання розділу формування пристрою автоматичного управління було визначено основні завдання руху для колісного мобільного робота. Розглянуті властивості моделювання та управління колісним мобільним роботом, зокрема сформовано принципи управління по точках і по заданій траєкторії. Проведено лінеаризацію для принципів зворотного зв'язку у системі колісного робота та розглянуті ланцюгові форми канонічної структури моделей роботів.

Запропоновано та спроектовано модель колісного робота, наведено опис його фізичних характеристик. Розглянуто архітектуру системи управління мобільним роботом, а також кінематику мобільного робота.

Для подальших досліджень конкретної моделі робота розраховано контрольні обмеження та синтезовано принципи управління для виконання руху по заданій траєкторії. Синтезовано математичну модель колісного робота.

3 КОНСТРУКТОРСЬКА ЧАСТИНА

3.1 Вибір об'єкта досліджень - робота версії AlphaBot

Для виконання досліджень обрано роботизовану платформу AlphaBot, зовнішній вигляд якого наведено на рис. 3.1.



Рисунок 3.1 – Зовнішній вигляд КМР

Роботизована конструкція AlphaBot – це універсальний стартовий набір для збирання роботизованої платформи. Унікальність цього комплекту порівняно з іншими наявними прототипами полягає в тому, що його можливості включають модульну структуру та широкі можливості використання в якості управляючого контролера як Arduino, так і з мікроконтролером Raspberry Pi або обома одночасно. Склад комплекту включає не лише саму платформу, але й двигуни і датчики.

Роботизовані конструкції AlphaBot на сьогодні доступні в трьох різних версіях. Ця комплексна механічна система включає 22 елементи, які зображено на рисунку 3.2.

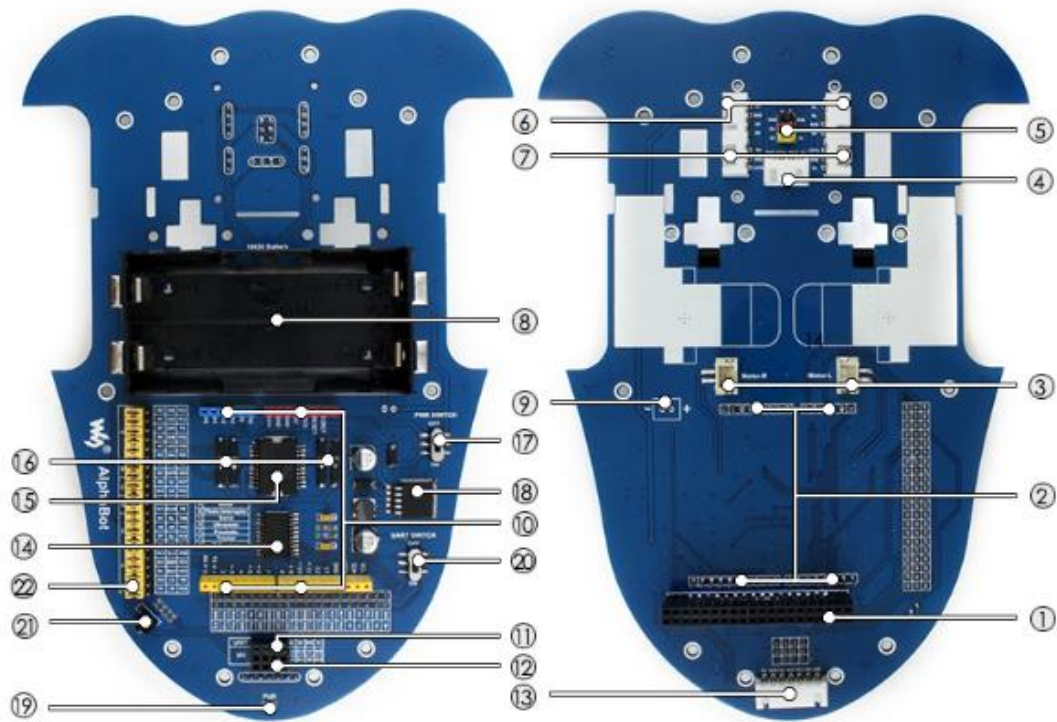


Рисунок 3.2 – Склад КМР типу AlphaBot

1. Arduino інтерфейс – для підключення контролера Arduino.
2. Raspberry Pi інтерфейс слугує для підключення плати Raspberry Pi.
3. Інтерфейс двигунів.
4. Інтерфейс серво-модуля.
5. Інтерфейс ультразвукового датчика відстані.
6. Інтерфейс вимірювача швидкості.
7. Інтерфейс модуля виявлення завад.
8. Резервний вхід живлення
9. Батарейний відсік для елементів 18650.
10. UART інтерфейс – для підключення Bluetooth модуля.
11. Роз'єм розширення для шилдів Arduino.
12. Модуль інтерфейсу line tracking.
13. SPI інтерфейс – для підключення модуля NRF24L01.
14. LM298P: драйвер двигунів із максимальним струмом 2А.
15. TLC1543: 10-біт АЦП для аналогових датчиків.
16. Захисний діод.
17. LM2596: 5В стабілізатор.
18. Вимикач живлення.
19. Індикатор живлення.
20. IR приймач – для віддаленого (дистанційного) управління роботом.
21. UART перемикач: перемикач послідовного інтерфейсу між Arduino та Raspberry Pi.

22. Arduino/Raspberry Pi перемикач: для перемикання периферії між Arduino та Raspberry Pi.

Номери переліку вище відповідають позиціям на рисунку 3.2.

Технічні характеристики цієї моделі колісного робота представлені в таблиці 3.1.

Таблиця 3.1 – Технічні характеристики AlphaBot

ПАРАМЕТР	ЗНАЧЕННЯ
Напруга живлення	7,2 В
Стабілізатор напруги	LM2596
Елементи живлення	2 x 18650
Управляючий контролер	Arduino UNO / Raspberry Pi
Драйвер двигунів	L298
Напруга логічних рівнів	5В / 3В
Інтерфейси АЦП для аналогових датчиків	TLC1543
Маса	577 грам

AlphaBot-Ar-Basic Версія "базова" відрізняється від "мінімальної" наявністю таких елементів: контролер UNO PLUS, ультразвуковий датчик відстані, сервопривід SG90, адаптер ультразвукового датчика, монтажна пластина для сервоприводу, ХН2.54 20 см 4Pin і USB-кабель.

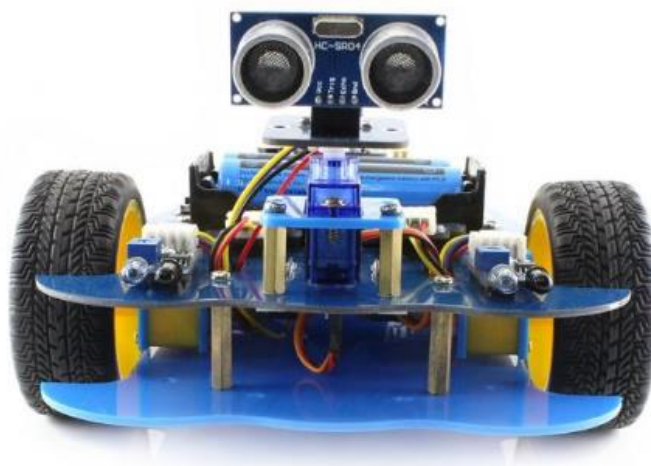


Рисунок 3.3 – Зовнішній вигляд КМР в комплектації AlphaBot-Ar-Basic

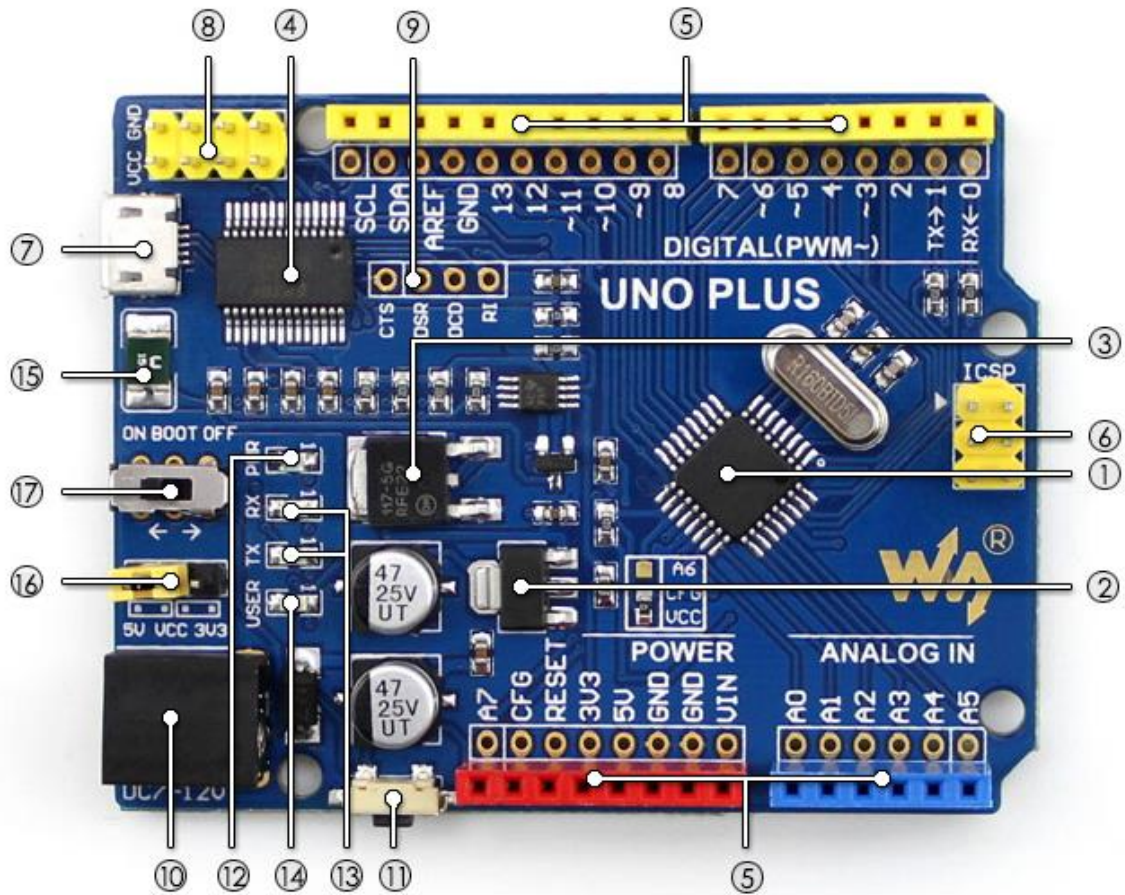


Рисунок 3.4 – Комплектація UNO PLUS

На рисунку 3.4 прийнято наступні позначення:

1. Мікроконтролер ATMEGA328P-AU.
2. Стабілізатор напруги AMS1117 -3.3 : 3.3В.
3. Стабілізатор напруги NCP1117ST50T3G.
4. USB-UART конвертер FT232RL.
5. Arduino інтерфейс.
6. ICSP інтерфейс.
7. MICRO USB роз'єм – для загрузки та налаштування програм.
8. Додатковий вихід для живлення.
9. FT232 виводи – для оновлення загрузчика мікроконтролера.
10. Вхід живлення 7..12В.
11. Кнопка перезавантаження.
12. Індикатор живлення.
13. Індикатор стану послідовного порту Rx/Tx.
14. Користувацький світлодіод (D13).
15. Захисний запобіжник.
16. Перемикач напруги живлення: 5В / 3.3В.

17. Перемикач роботи загрузчика (ON/OFF).

Представлена версія AlphaBot (рис. 3.5) може відслідковувати напрямок руху, вимірювати швидкість та обходити перешкоди. Управління рухом здійснюється за допомогою зручного інфрачервоного пульта. Вся система управління платформою реалізована через програмне забезпечення з відкритим вихідним кодом Arduino.



Рисунок 3.5 – Робо платформа AlphaBot Bluetooth

Для даного проектування було обрано комплектацію робота AlphaBot.

3.2 Вибір плати управління

Для створення системи керування рухомим апаратом необхідно вибрати обчислювальний блок на основі мікроконтролера. Нижче приведено ряд обчислювачів, які відрізняються за технічними параметрами, габаритами та ціною стратегією.



Рисунок 3.6 – Зовнішній вигляд мікроконтролер ного обчислювача «Raspberry Pi»

Технічні характеристики контролера представлені у таблиці 3.2.

Таблиця 3.2 – Характеристика «Rasberry Pi»

ПАРАМЕТР	ЗНАЧЕННЯ
Комплектація	SoC Broadcom BCM2835 (CPU, GPU, DSP и SDRAM)
Процесор	CPU: 700 МГц ARM1176JZF-S core (смейство ARM11)
Графічний проц.	GPU: Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30
Графічний кодек.	h.264/MPEG-4 AVChigh-profile decoder
ОЗП	(SDRAM): 512 Мб
Роз'єми	microSD, 10/100 Ethernet RJ45, HDMI, 3.5 мм

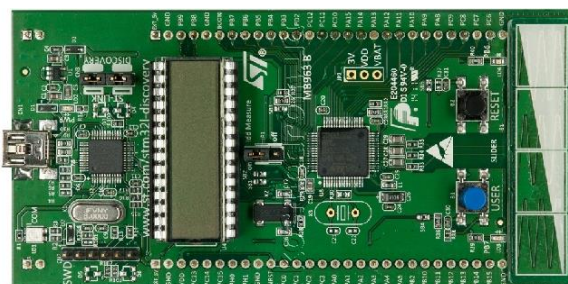


Рисунок 3.7 – Зовнішній вигляд мікроконтролерного обчислювача «Discovery STM 32l»

Технічні характеристики контролера представлені у таблиці 3.3.

Таблиця 3.3 – Характеристика «Discovery STM 32l»

ПАРАМЕТР	ЗНАЧЕННЯ
Мікроконтролер	STM32L152RBT6
Робоча частота	16 МГц
Робоча напруга	5 В
Напруга живлення	7-9 В
Кількість цифрових вх / вих	32 (16 ШІМ)

Продовження таблиці 3.3

Кількість аналогових входів	16
Сила струму на входах / виходах	40 мА
Об'єм flash пам'яті	32 кБ
SRAM	2 кБ
EEPROM	1 кБ

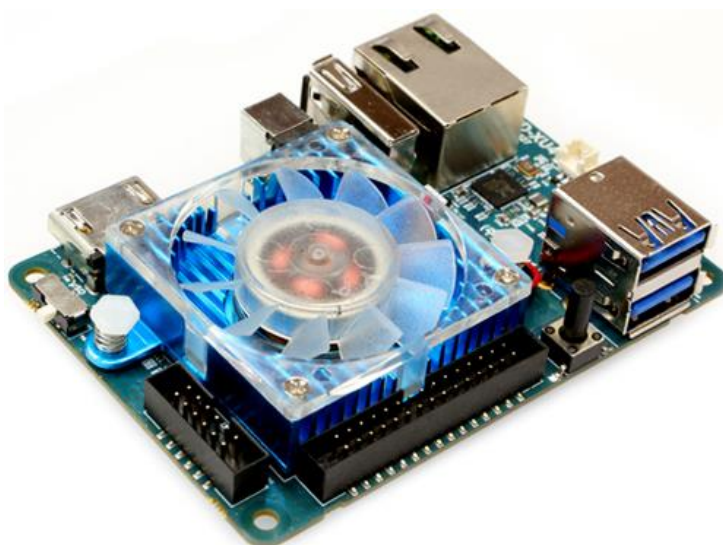


Рисунок 3.8 – Зовнішній вигляд обчислювача «ORDROID xu4 »

Технічні характеристики контролера представлені в таблиці 3.4.

Таблиця 3.4 – Характеристика «ORDROID xu4 »

ПАРАМЕТР	ЗНАЧЕННЯ
Центральний проц.	Samsung Exynos5422 Cortex™-A15 2Ghz та 8-ядерний процесор Cortex™-A7
Графічний проц.	Mali-T628 MP6(підтримка OpenGL ES 3.0/2.0/1.1 та OpenCL 1.1 повний профіль)
ОЗП	2ГБайта LPDDR3 RAM PoP
Flash пам'ять	eMMC5.0 HS400, 2 x USB 3.0 Host, 1 x USB 2.0 Host
Інтернет з'єднання	Гігабітний Ethernet порт
Інтерфейси	HDMI 1.4a
Габарити	82x58x22 мм (включаючи кулер процесора)

Існує множина аналогів вищезазначених контролерів, всіх їх можна розділити на дві категорії: з операційною системою і без операційної системи. Перша категорія надає різноманітні можливості для спрощення процесу розробки, такі як віддалений доступ через SSH протокол, можливість налаштування програми на стаціонарному комп'ютері і т. д.

Також важливо відзначити, що продуктивність процесора на кілька порядків вища у контролерів, що відносяться до першої категорії, що є вкрай актуальним при виконанні завдань обробки відеопотоку в режимі реального часу. Для досягнення поставленої мети необхідна наявність графічного процесора для ефективної обробки відеозображення.

Важливою перевагою є здатність реалізації багатозадачності при впровадженні подібних систем, що також вказує на перевагу Raspberry Pi або Ordroid хі4 для керування. Проте плата Ordroid з урахуванням її характеристик та архітектури призначена для проектів і систем, пов'язаних із наявністю сервера та обробкою мережевої інформації, що вказує на те, що найбільш підходящим пристроєм є Raspberry Pi.

3.3 Опис плати управління мобільним роботом

У ролі бортового комп'ютера взяли одноплатний пристрій Raspberry Pi B, який під час експерименту виконує завдання обчислювального ресурсу для обробки зображень в режимі реального часу та функції управління механічною частиною (керування двигунами).

Цей пристрій є універсальним для такого завдання, оскільки має високу продуктивність (порівняно з іншими мікроконтролерами, оснащений операційною системою Raspbian, що базується на Debian Linux, та має достатньо ресурсів для обробки зображень малих розмірів). Крім того, він забезпечує можливість апаратного підключення зовнішніх пристроїв і датчиків за допомогою цифрових виходів GPIO, що є критичним для цього проекту.

Для забезпечення комутації між обчислювачем і елементами управління необхідно використовувати GPIO виходи одноплатного комп'ютера Raspberry Pi B. Ці виходи дозволяють виконувати перемикання рівнів, формувати сигнали ШІМ, а також використовувати різні інтерфейси для передачі і зчитування даних (I2C, SPI, UART).

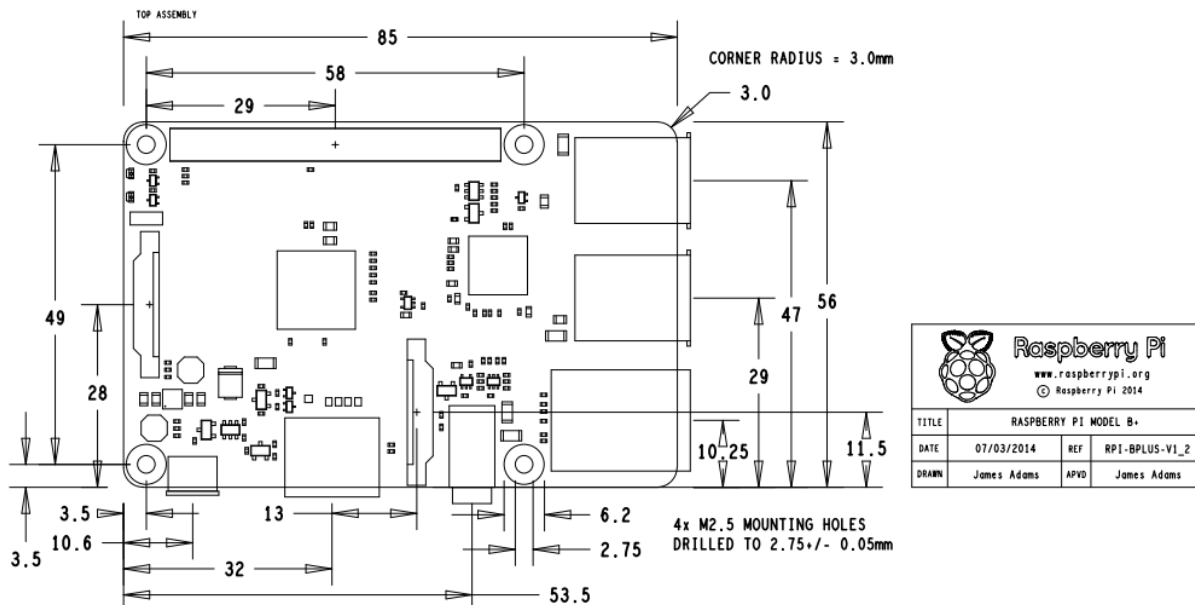


Рисунок 3.9 – Схематичне зображення Raspberry Pi B

3.4 Опис плати L298n

Для розв'язку задачі управління силовою частиною мобільного робота використовується плата L298n (рис. 3.10).

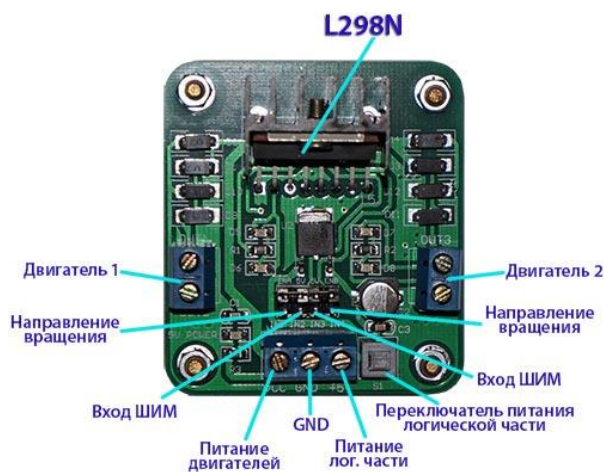


Рисунок 3.10 – Підсилююча плата керування двигунами

Принципова схема такого модуля має такий вигляд:

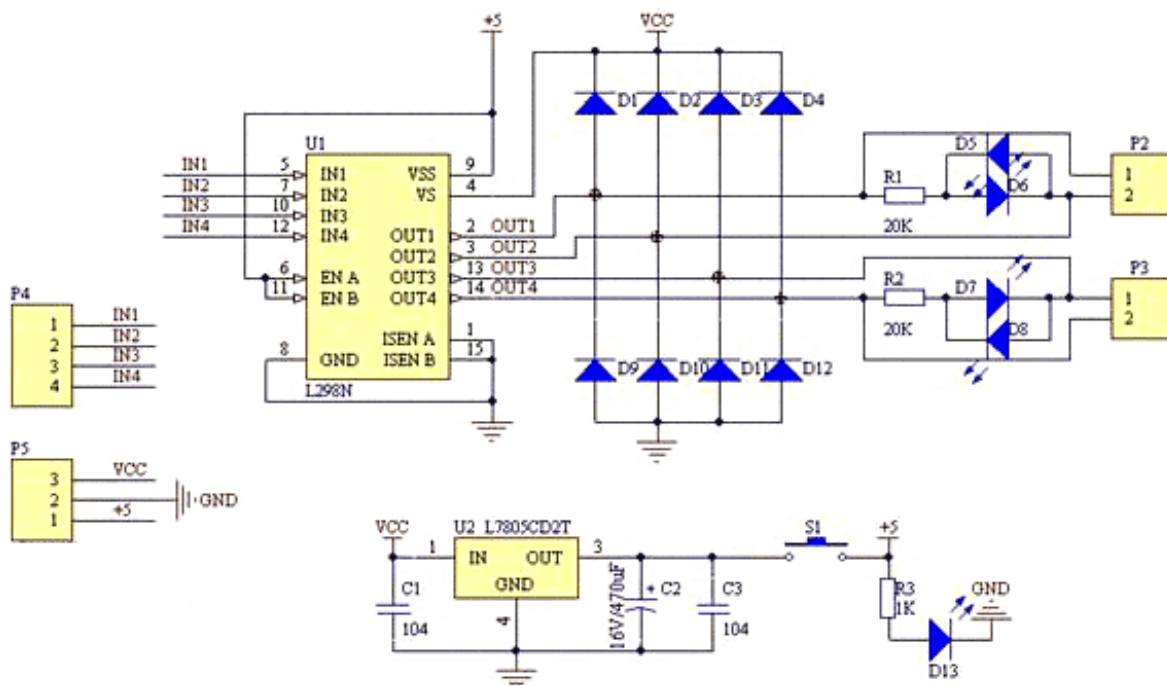


Рисунок 3.11. – Зображення принципової схеми модуля управління двигунами

3.5 Висновки до конструкторської частини

Під час розробки конструкторської частини дипломного проекту було обрано модифікацію робота AlphaBot. Вибір проводився шляхом порівняння характеристик серед аналогічних роботизованих платформ, до яких відносяться AlphaBot-Ar-Basic та AlphaBot-Ar-Bluetooth.

Також здійснено відбір плати управління.

Визначено, що для управління мобільним роботом буде використовуватися одноплатний комп'ютер Raspberry Pi через його переваги. Порівняння проводилося з наступними обчислювачами: Raspberry Pi, Discovery STM321, ORDROID xu4.

Всі порівняльні характеристики відображено в пунктах 3.1 та 3.2 у пояснювальній записці до дипломної роботи.

Крім того, у роботі розглянуто плату L298n, яка використовується для управління силовою частиною мобільного робота, а також детально описано обраний мікроконтролер для керування мобільним роботом.

4 ДОСЛІДНИЦЬКА ЧАСТИНА

4.1 Модельні випробування

Нижче наведено результати моделювання руху робота по заданій траєкторії (рис. 4.1).

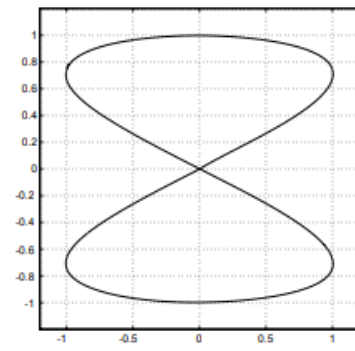


Рисунок 4.1 – Вісімка-подібна траєкторія руху

Даний рух описується наступним законом:

$$x_d(t) = \sin \frac{t}{10}, y_d(t) = \sin \frac{t}{20}, \text{ при } t \in [0; T]. \quad (4.1)$$

Траєкторія починається з початку координат із $T = 2\pi \cdot 20 \approx 125$ с. При цьому стартова швидкість становить:

$$v(0) \approx 0,1118 \text{ м/с}; \quad (4.2)$$

$$\omega(0) = 0 \text{ рад/с.}$$

У першій фазі моделювання, конфігурація робота узгоджена із початковою опорною траєкторією. Таким чином, команди форсування дозволяють точно слідувати траєкторії в ідеальних умовах. Але на практиці існує певний перехідний процес.

На зображеннях 4.2 – 4.4 представлені результати, отримані за допомогою лінійно спроектованого контролера, з використанням коефіцієнтів підсилення із $\zeta = 0,7$ та $b = 10$.

Відстеження еталонної траєкторії на малюнку 4.1 дійсно досить точне. Залишкові похибки в основному пов'язані із квантуванням та дискретизацією команд швидкості. Зокрема, існує велика помилка в часі через динаміку власне

транспортного засобу, тобто приводу із початковим ненульовим значенням $v_d(0)$. Це демонструється на рисунку 4.5.

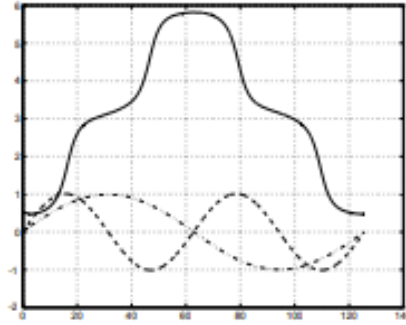


Рисунок 4.2 – Траєкторне відстеження з лінійної конструкцією зв'язку

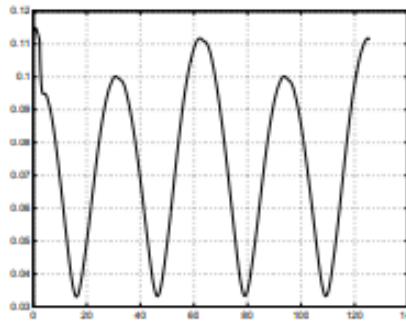


Рисунок 4.3 – Траєкторне відстеження з лінійної конструкцією зворотнього зв'язку для швидкості

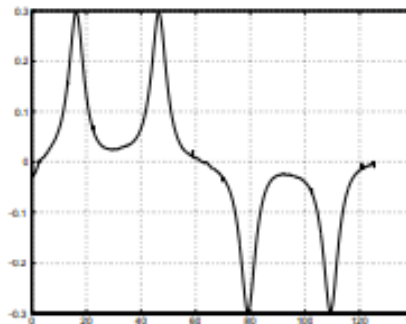


Рисунок 4.4 – Траєкторне відстеження з лінійної конструкцією зворотнього зв'язку для кутового прискорення

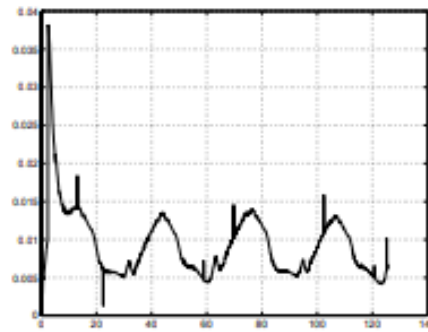


Рисунок 4.5 – Траєкторне відстеження з лінійною конструкцією зворотного зв'язку для норми декартової похибки швидкості

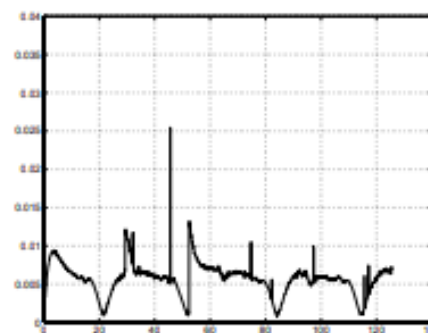


Рисунок 4.6 – Траєкторне відстеження з лінійною конструкцією зворотного зв'язку для норми декартової похибки прискорення

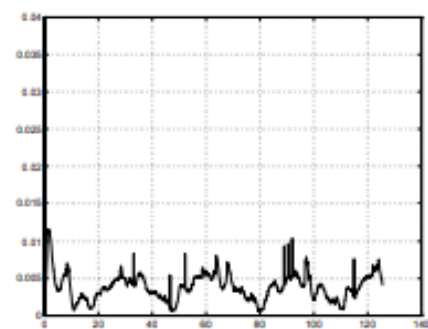


Рисунок 4.7 – Траєкторне відстеження при динамічній лінеаризації зворотного зв'язку: норма декартової погрешності

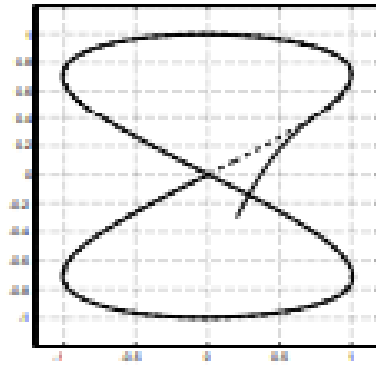


Рисунок 4.8 – Відстеження асимптотичних траєкторій із проектуванням лінійних зворотних зв'язків: декартовий рух (x, y)

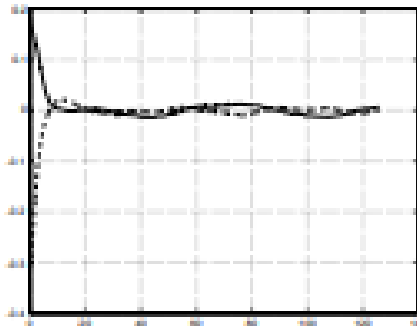


Рисунок 4.9 – Відстеження асимптотичних траєкторій з проектуванням лінійних зворотних зв'язків: декартова похибка

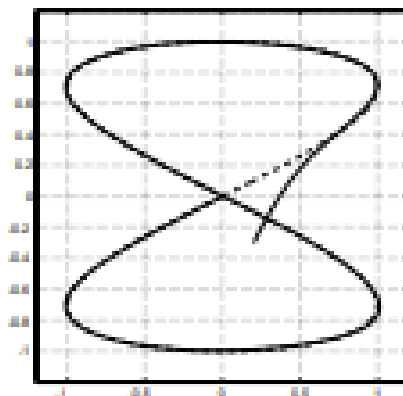


Рисунок 4.10 – Відстеження асимптотичної траєкторії через динамічну лінеаризацію зворотного зв'язку: декартовий рух (x, y)

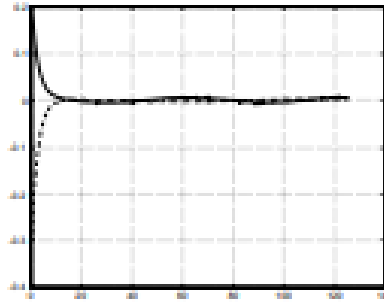


Рисунок 4.11 – Відстеження асимптотичної траєкторії через динамічну лінеаризацію зворотного зв'язку: декартова похибка

Подібно лінійному проводиться моделювання і для нелінійного контролера, використовуючи лінеаризацію із динамічним оберненим зв'язком.

Для оцінки покращення продуктивності, порівнюються декартові відхилення на малюнках 4.6 та 4.7 із попереднім результатом 4.5.

Виявлено, що середня відхилення зменшиться від значення 1 см (лінійна конструкція) до 0,5 см (нелінійна конструкція) і, нарешті, до 0,38 см (конструкція лінеаризації динамічного оберненого зв'язку).

4.2 Стабілізація стану

Як зазначалося у другому розділі, положення КМР не може бути досягнуте шляхом гладкого статичного оберненого зв'язку. Тому пропонується досліджувати кілька видів регулювання: плавне регулювання, що змінюється з часом, та ті, що базуються на перервному оберненому зв'язку.

Плавне змінне з часом регулювання. За допомогою припущень, виконаних раніше, можна стверджувати, що розв'язок проблеми стабілізації положення може бути отримано шляхом проектування бажаного руху. А отже, структура плавного стабілізуючого регулятора, така ж сама, як і у контролера нелінійної траєкторії:

$$\begin{aligned}
 u_1 &= -k_1(v_d(t), \omega_d(t))e_1; \\
 u_2 &= -\bar{k}_2 v_d(t) \frac{\sin e_3}{e_3} e_2 - k_3(v_d(t), \omega_d(t))e_3.
 \end{aligned}
 \tag{4.4}$$

У такому випадку траєкторія сама собою являється додатковим ступенем свободи, із урахуванням обмеження, яке характеризується у вигляді похибки відстеження стану.

$$v_d(t) = \dot{x}_d(t) = -k_4 x_d(t) + g(e, t), \text{ при } k_4 > 0,
 \tag{4.5}$$

де, $g(e, t)$ – C^2 -функція рівномірно обмежена за часом разом з її частковою похідною.

Використовуючи ту саму функцію Ляпунова, похибка являється обмеженою.

Функція $g(e, t)$ відіграє ключову роль по забезпеченню асимптотичної стійкості, підтримуючи рух, доти, поки похибка не буде дорівнювати нулю.

Проводиться тестування роботи перемінного в часі закону управління із бажаним рухом, що ініціалізований у точці $x_d(0) = 0$, та функція $g(e, t)$. Коефіцієнти вибираються із ряду $k_1 = 0,5$; $k_2 = 2$; $k_3 = 1$; $k_4 = 1$ та $k_5 = 50$.

Початок моделювання відбувається на початку координат. Ця задача буде використовуватися в якості базової лінії для порівняння всіх метрів стабілізації положення. На рисунках 4.12 – 4.15 показано, що після відносно швидкого наближення згладжування плавного контролеру, що перемінний в часі.

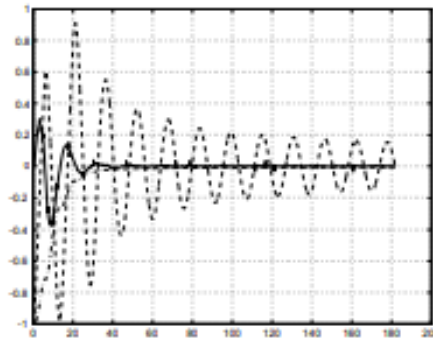


Рисунок 4.12 – Статична стабілізація положення із гладким змінним в часі зворотним зв'язком

Наприклад, на малюнку 6.2 зображено рух робота, який фіксується через кожні 10 секунд. Одним з обмежень цього роду управління є більша кількість додаткових маневрів, узгоджених із кінцевою бажаною орієнтацією. Діапазон коливань x знаходиться в межах $(-1;1)$ за рахунок великого значення п'ятого коефіцієнта.

Точність регулювання на вихідних координатах визначається відповідністю наступним термічним обмеженням, які використовуються у всіх моделях для стабілізації положення:

$$|x| < 0,2 \text{ cm} \quad |y| < 0,2 \text{ cm} \quad |\theta| < 0,02 \text{ rad}. \quad (4.6)$$

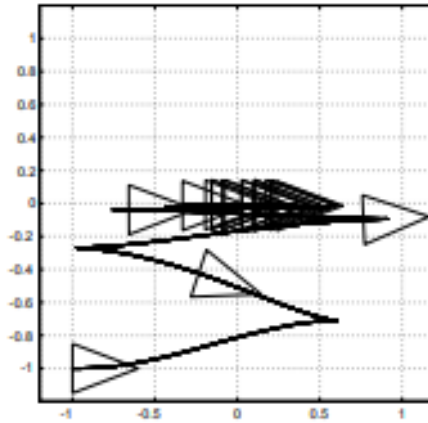


Рисунок 4.13 – Статична стабілізація положення зі зворотним зв'язком, що плавно змінюється в часі

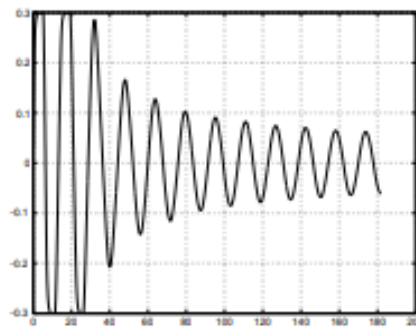


Рисунок 4.14 – Стабілізація положення із зворотним зв'язком, що плавно змінюється в часі: по швидкості руху

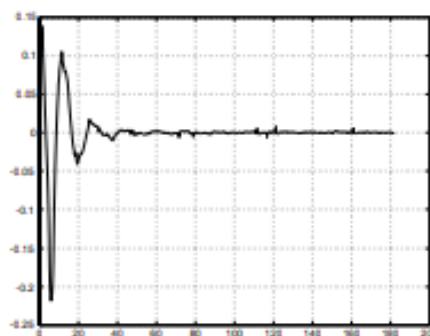


Рисунок 4.15 – Стабілізація положення із зворотним зв'язком, що плавно змінюється в часі: по кутовій швидкості

Неоднорізна зміна з часом. Відмовившись від необхідності плавності, було запропоновано кілька контролерів для стабілізації положення із покращеними перехідними характеристиками. Закон управління є неперервним в часі, однак

нестійким по відношенню до стану, що повертаються лише при рівномірно вибраних моментах.

На основі математичної моделі, описаної в розділі 2, проводиться моделювання для регулювання із неплавною зміною в часі.

На малюнках 4.16 – 4.19 показано результати застосування закону управління, де $T=6$ та $k=0,8$; $\lambda_2=\lambda_3=0,4$ для виконання базової задачі перед початковою виставкою. Швидкість конвертації неплавного контролера, що змінюється з часом, дещо покращена, однак все ще доволі мала. Вид руху колісного робота, який фіксується через кожні 5 секунд, представлений на малюнку 4.17.

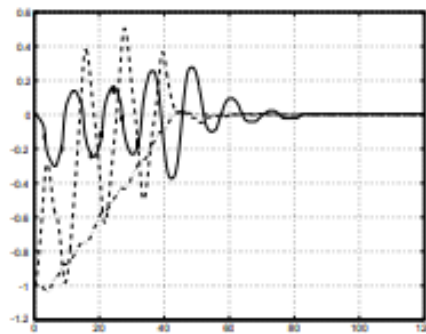


Рисунок 4.16 – Стабілізація положення із зворотним зв'язком, що випадково змінюється в часі

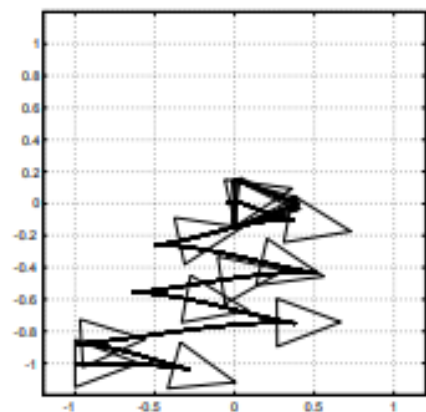


Рисунок 4.17 – Стабілізація положення із зворотним зв'язком, що не плавно змінюється в часі: декартовий рух

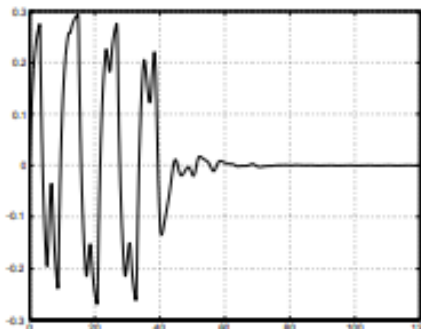


Рисунок 4.18 – Стабілізація положення із зворотним зв'язком, що не плавно змінюється в часі: швидкість руху

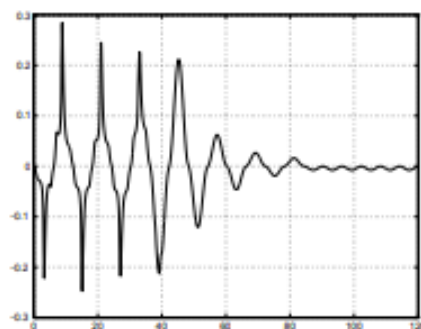


Рисунок 4.19 – Стабілізація положення із зворотним зв'язком, що не плавно змінюється в часі: кутова швидкість

Необхідно відзначити, що підхід у напрямку y є дуже однорідним, а маневри y безпосередній близькості від ланцюга спрямовані на налаштування θ , а не x . Це є невід'ємною частиною структури ланцюгової форми для розробки системи управління.

Розглянемо управління на основі полярних координат. Інший підхід дозволяє застосувати зміну координат так, щоб вхідні векторні поля були сингулярні в початкових координатах.

На малюнках 4.20 – 4.23 представлені результати використання контролера із коефіцієнтами підсилення $k_1=1$, $k_2=3$ та $k_3=2$ для виконання базової задачі.

На малюнку 4.21 показано вид руху одноколісного робота, що відраховується кожну секунду.

Поведінка системи управління не є неперервною відносно початкового стану. Наприклад, початкова конфігурація дорівнює:

$$x(0) = a > 0, y(0) = \varepsilon \text{ та } \theta(0) = 0. \quad (4.7)$$

Додатні й від'ємні найменш малі значення ε приведуть до різних перехідних процесів до ланцюга.

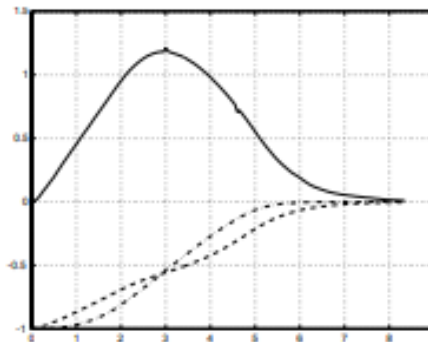


Рисунок 4.20 – Стабілізація положення із зворотним зв'язком в полярних координатах

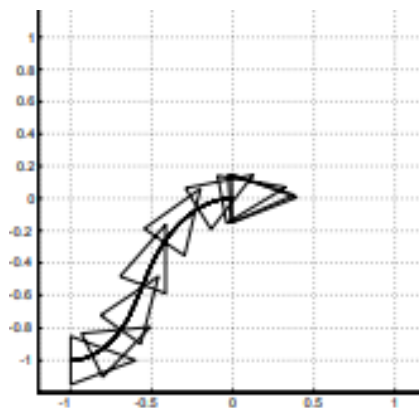


Рисунок 4.21 – Стабілізація положення із зворотним зв'язком у полярних координатах: декартовий рух

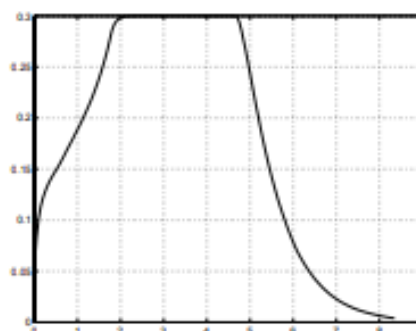


Рисунок 4.22 – Стабілізація положення із зворотнім зв'язком в полярних координатах: швидкість руху

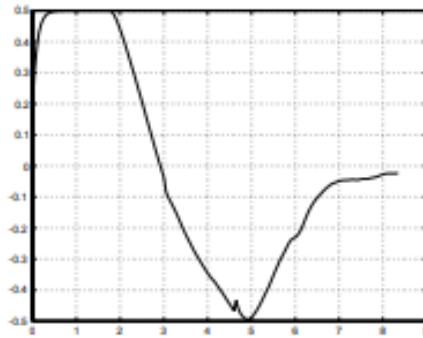


Рисунок 4.23 – Стабілізація положення із зворотнім зв'язком в полярних координатах: кутова швидкість

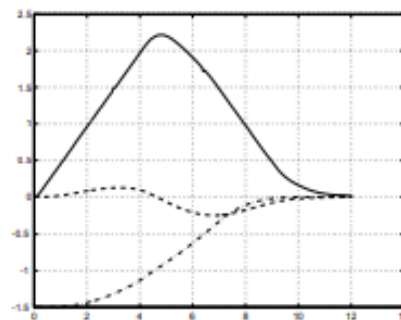


Рисунок 4.24 – Стабілізація положення із зворотнім зв'язком в полярних координатах (паралельний рух)

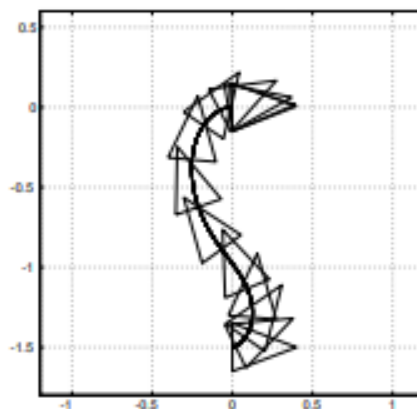


Рисунок 4.25 – Стабілізація положення із зворотнім зв'язком у полярних координатах: декартовий рух (паралельний рух)

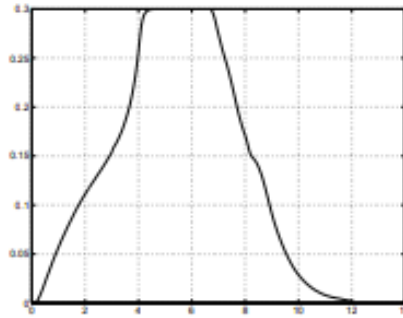


Рисунок 4.26 – Стабілізація положення із зворотнім зв'язком в полярних координатах: швидкість руху (паралельний рух)

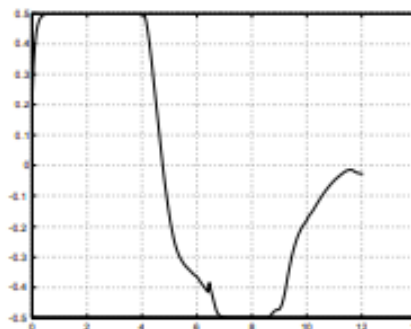


Рисунок 4.27 – Стабілізація положення із зворотнім зв'язком в полярних координатах: кутова швидкість (паралельний рух)

Розширити можливості контролера, досліджуючи траєкторії на основі лінеаризації динамічного зворотного зв'язку для вирішення проблеми стабілізації положення, уникаючи внутрішніх особливостей, що виникають при зупинці робота. Це просто потребує відповідного вибору коефіцієнтів підсилення.

Оскільки асимптотична швидкість збіжності дорівнює $2|\lambda_{22}|$, то ω прямує до нуля при $t \rightarrow \infty$.

Орієнтація θ збігається до нуля. Це легко визначити за допомогою таких факторів:

- 1) одноколісний робот досягає начала координат із горизонтальною дотичною, оскільки у прискорюється до нуля швидше, ніж x , із урахуванням впорядкування на власні значення;
- 2) інверсії руху не виникають, оскільки швидкість ніколи не перетинає значення нуля;
- 3) траєкторія обмежена областю, з якої починається рух робота; фактично координати робота ніколи не змінюють свій знак.

Рисунки 4.28 – 4.31 відносяться до завдання, використовуючи лінеаризацію із динамічним зворотнім зв'язком із коефіцієнтами $[[k_p]]_1=2$, $[[k_d]]_1=3$ та $[[k_p]]_2=12$, $[[k_d]]_2=7$. На малюнку 4.29 показано рух робота кожні 1,5 секунди. Насичення відбувається на обох входах під час перехідної фази.

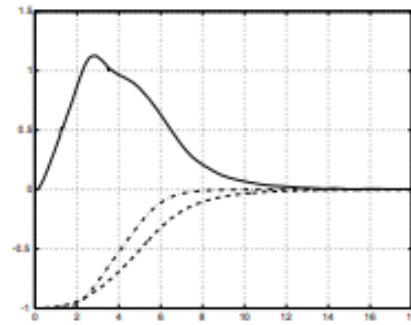


Рисунок 4.28 – Стабілізація положення шляхом динамічної оберненої лінеаризації

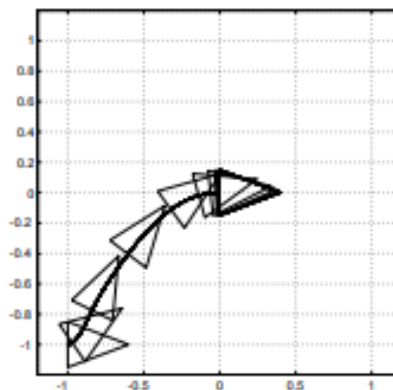


Рисунок 4.29 – Стабілізація положення шляхом динамічної оберненої лінеаризації: декартовий рух

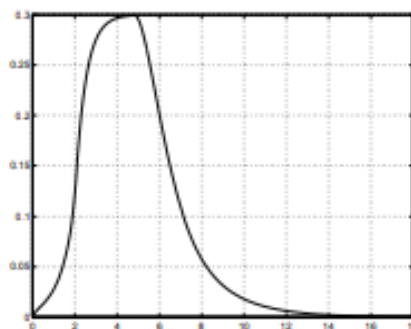


Рисунок 4.30 – Стабілізація положення шляхом динамічної оберненої лінеаризації: швидкість руху

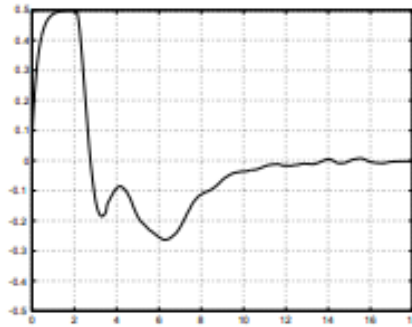


Рисунок 4.31 – Стабілізація положення шляхом динамічної оберненої лінеаризації: кутова швидкість

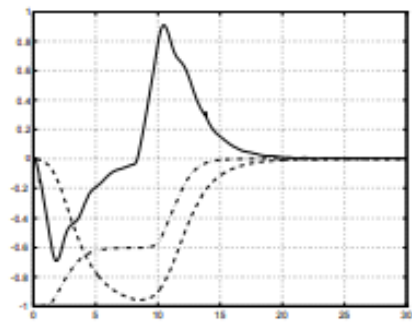


Рисунок 4.32 – Стабілізація положення шляхом динамічної оберненої лінеаризації (паралельний рух)

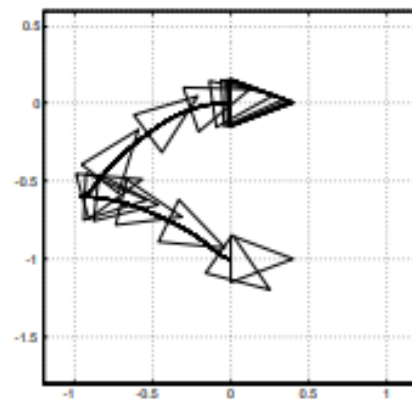


Рисунок 4.33 – Стабілізація положення шляхом динамічної оберненої лінеаризації: декартовий рух (паралельний рух)

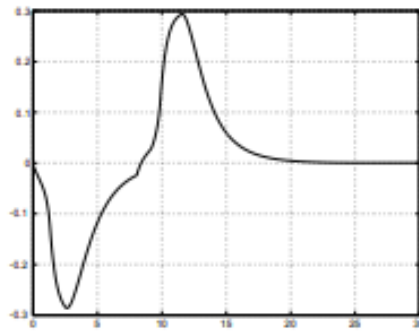


Рисунок 4.34 – Стабілізація положення шляхом динамічної оберненої лінеаризації: швидкість руху (паралельний рух)

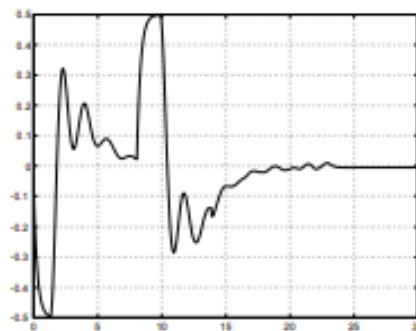


Рисунок 4.35 – Стабілізація положення шляхом динамічної оберненої лінеаризації: кутова швидкість (паралельний рух)

Результати, що показані на рисунках 4.32 – 4.35 – це простий лінійний фільтр першого порядку, що введено для урахування динаміки привода, а також ефективний для згладжування розриву швидкості руху, що створюється процедурою. З іншого боку, наявність того ж фільтру для кутової швидкості в поєднанні із насиченням програмного забезпечення нейтралізує ефект сингулярності.

4.3 Висновки

Для вирішення завдання розробки системи об'єктивного контролю руху транспортного робота під час розгляду розділу 4, у науковій частині дипломного проекту, було проведено моделювання роботи системи та моделювання стабілізації стану, враховуючи плавні та несплавні зміни часу регулювання, а також для полярної системи координат. Крім того, була виконана модель системи в умовах лінійного оберненого зв'язку..

5 ЕКСПЕРИМЕНТАЛЬНО-ПРАКТИЧНА ЧАСТИНА

5.1 Завдання і цілі виконання експериментальної розробки

Під час проведення експериментальної частини потрібно виконати наступний набір завдань:

- налаштувати управління роботом;
- підготувати майданчик для випробувань транспортного засобу;
- встановити маркери на точці фінішу;
- здійснити експеримент щодо переміщення робота від точки старту до точки фінішу з урахуванням обходу перешкод.

5.2 Опис лабораторної установки

У експериментальній частині необхідно підтвердити правильність розробленої системи об'єктивного контролю руху транспортного робота. Для перевірки правильності функціонування системи керування необхідно провести тестування отриманої системи, розділене на декілька основних частин: пошук цілі, переміщення до цілі, гальмування, повернення до місця старту.

В якості моделі колісного робота для тестування була обрана система з чотирма колесами та повним приводом.

Спочатку для роботи з даною обчислювальною системою необхідно здійснити ряд налаштувань. Операційна система контролера - Raspbian, що є дистрибутивом Debian. Система, що розробляється, вимагає використання алгоритмів комп'ютерного зору та взаємодії із зовнішніми пристроями (управління двигунами). Для цього потрібно встановити бібліотеки SimpleCV (з алгоритмами комп'ютерного зору) та RPIO (бібліотека для роботи із зовнішніми пристроями).

З використанням віддаленого доступу через протокол SSH будемо проводити встановлення бібліотек SimpleCV і RPIO через термінал.

```

pi@raspberrypi: ~
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Apr  8 13:08:36 2017 from 192.168.0.103
pi@raspberrypi:~$ idle
pi@raspberrypi:~$ sudo apt-get install python-dev python-rpi.gpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package python-rpi
pi@raspberrypi:~$ sudo apt-get install python-dev python-rpi.gpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-dev is already the newest version.
python-dev set to manually installed.
python-rpi.gpio is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 156 not upgraded.
pi@raspberrypi:~$

pi@raspberrypi: ~/Code/SimpleCV
requirements.txt (line 1)
Requirement already satisfied (use --upgrade to upgrade): scipy in /usr/lib/pyth
requirements.txt (line 2)
Downloading/unpacking PIL (from --requirements.txt (line 3))
Could not find any downloads that satisfy the requirement PIL (from --require
Cleaning up...
No distributions at all found for PIL (from --requirements.txt (line 3))
Storing debug log for failure in /root/.pip/pip.log
pi@raspberrypi:~/Code/SimpleCV $ sudo python setup.py develop
running develop
running egg_info
creating SimpleCV.egg-info
writing SimpleCV.egg-info/PKG-INFO
writing top-level names to SimpleCV.egg-info/top_level.txt
writing dependency links to SimpleCV.egg-info/dependency_links.txt
writing entry points to SimpleCV.egg-info/entry_points.txt
writing manifest file 'SimpleCV.egg-info/SOURCES.txt'
reading manifest file 'SimpleCV.egg-info/SOURCES.txt'
writing manifest file 'SimpleCV.egg-info/SOURCES.txt'
running build_ext
Creating /usr/local/lib/python2.7/dist-packages/SimpleCV.egg-link (link to .)
Adding SimpleCV 1.3 to easy-install.pth file
Installing simplecv script to /usr/local/bin

Installed /home/pi/Code/SimpleCV
Processing dependencies for SimpleCV==1.3
Finished processing dependencies for SimpleCV==1.3
pi@raspberrypi:~/Code/SimpleCV $

```

Рисунок 5.1 – Зображення установки утиліт через термінал

Важливо врахувати, що верхній рівень логічного сигналу на виходах GPIO становить 3.3 В, а пристрої не мають захисту від перевантаження.

Для здійснення дистанційного доступу між мобільним колісним роботом і комп'ютером використовується додаток для віддаленого доступу putty, який працює з протоколом ssh.

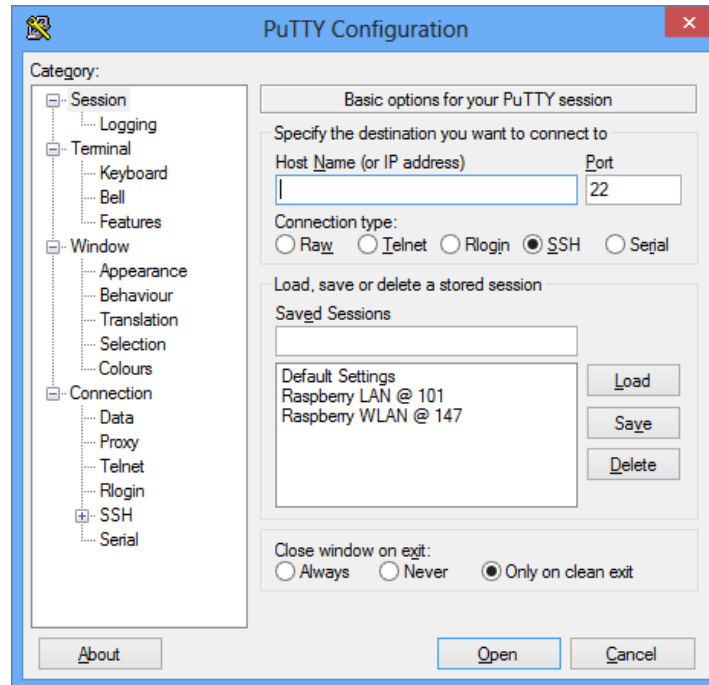


Рисунок 5.2 – Зображення додатки по реалізації віддаленого управління по протоколу ssh

Для реалізації з'єднання робота з комп'ютером за допомогою putty, необхідно об'єднати їх в одну локальну мережу через модем і підключитися до IP-адреси пристрою.

5.3 Аналіз результатів проведених експериментів та впровадження розробки

Як було вказано в розділі 2, вся навігація робота ґрунтується на використанні об'єктивної інформації; отже, всі навігаційні параметри необхідно синтезувати зображення. Згідно з розділом 1, об'єктом орієнтації буде м'ячик однорідного кольору. Для реалізації пошуку кута напрямку до об'єкта орієнтування, колісному роботу слід виконати круговий оберт навколо власної вертикальної осі до зупинки, коли об'єкт орієнтування (кулька) потрапляє в центр зображення камери.

Відповідно до ланцюгового коду Фрімана (зазначеного в розділі 2), виконується кадровий пошук замкнутого контуру і подальших морфологічних перетворень.

Алгоритм роботи пошуку об'єкта орієнтації показано на рис. 5.3.



Рисунок 5.3 – Зображення алгоритму пошуку предмета орієнтування

Так само далі наведений приклад реалізації роботи алгоритму. Видно з рис. 5.4, на однорідному фоні добре визначається предмет орієнтування.

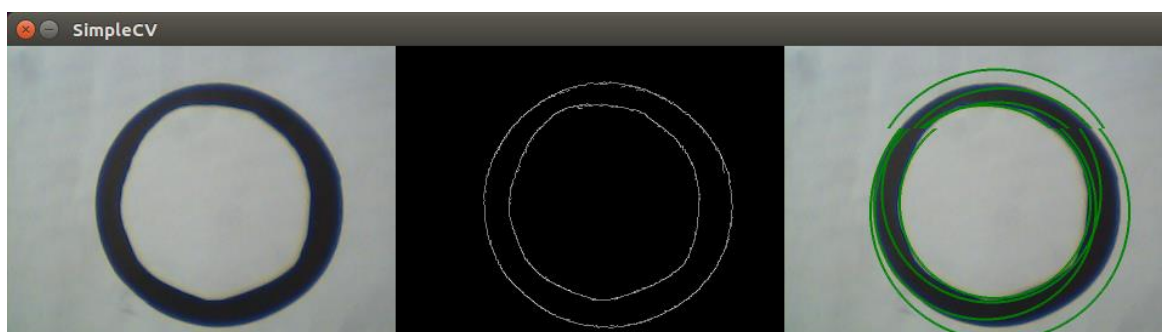


Рисунок 5.4 – Алгоритм пошуку об'єкта орієнтування

Необхідно додатково відзначити, що алгоритм не передбачає методів захисту від перешкод, тому для коректної роботи програми бажано усунути з поля виконання місії все круглі об'єкти із метою уникнення подвійного детектування.

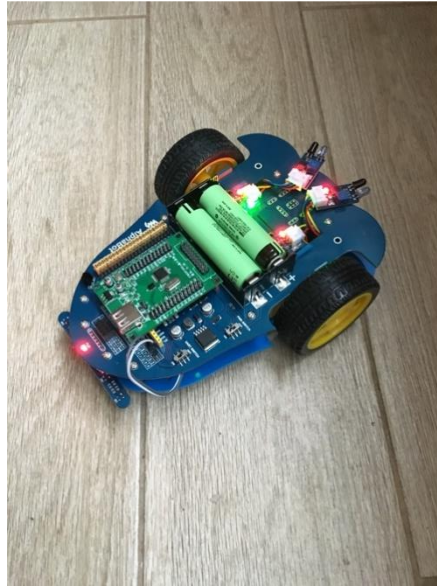


Рисунок 5.5 – Зображення експериментальної моделі колісного робота

Під час переміщення робота через турбуленції або різкі рухи може відбутися тимчасова втрата об'єкта (цілі). Алгоритм передбачає зупинку і очікування до відновлення виявлення об'єкта.

Оскільки в досліджуваному роботі відсутні ультразвукові або інфрачервоні датчики, інформацію про відстань до перешкоди (до об'єкта орієнтування) також потрібно визначати за допомогою об'єктної (візуальної) інформації.

Для спрощення процедури визначення відстані (а також для вказівки, що ця інформація використовується виключно для захисту робота від зіткнень), можна визначати відстань не числово, а якісно, тобто виявляти «малу» або «небезпечну» відстань. Для цього достатньою умовою буде діаметр м'яча в порівнянні з шириною зображення. Якщо він перевищує 60% ширини зображення, буде вислатися сигнал про досягнення об'єкта і переходження в режим гальмування і повернення.

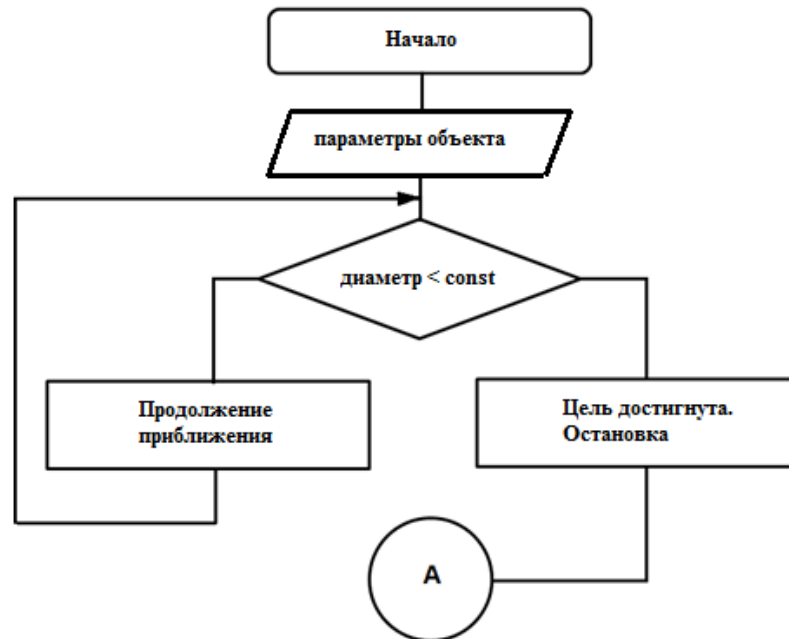


Рисунок 5.6 – Зображення алгоритму руху до мети

Алгоритм переміщення від мети розроблено на зворотньому русі при встановленому куті курсу. Слід відзначити, що дана система не призначена для випадків, коли робот механічно висувається з траєкторії заданого руху, буксує в зоні обмеженого руху чи штучно перериває свій рух. Рух вважається рівномірним, а відстань визначається за часом руху в напрямку до об'єкта.

Для проведення перевірки валідності, в кінці руху назад здійснюється порівняння радіуса об'єкта з початковим. При збігу довжин (з допуском в 5%) маршрут вважається правильним, місце зупинки вважається вірним, а завдання виконаним.

5.4 Висновки

У розділі формування системи автоматичного управління були розроблені методи і алгоритми для навігації мобільного робота з використанням об'єктивної інформації. Алгоритми базуються на даних, отриманих від кадрової обробки відеокамери.

6 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ

6.1 Мета економічної частини

Метою цього розділу в випускній роботі є економічне обґрунтування розробки системи об'єктивного контролю руху транспортного робота. Для досягнення поставленої мети необхідно вирішити такі завдання:

- 1) здійснити сегментацію ринку для систем управління роботами,
- 2) розрахувати трудомісткість виконання завдань,
- 3) скласти кошториси на розробку системи об'єктивного контролю руху транспортного робота,
- 4) провести розрахунок заробітної плати, визначити вартість об'єкта і розрахувати очікуваний прибуток від реалізації.

6.2 Короткий опис апарату

Розроблена система – це апаратно-програмний комплекс, який може виконувати управління колісного робота у закритому просторі без наявності сигналу від супутникової навігаційної системи на підставі візуальної інформації. Прототип робота може здійснювати пошук місця із спеціальною позначкою у закритому просторі із можливістю обходу перешкод та запам'ятовувати шлях від точки старту до точки фінішу з подальшим обчисленням найшвидшої можливої траєкторії на підставі теорії графів. Вся навігаційна робота реалізується на підставі обробки оптичного потоку камери за методом Лукаса – Канаде.[26]

6.3 Компанія виробника

В якості компанії - виробника була обрана компанія «AlphaBot», що наразі є лідером по розробці компактних мобільних роботів. Продукція цієї компанії дозволяє швидко та легко створювати прототипи систем управління роботами за допомогою завдяки гнучкості апаратної частини (є можливість під'єднувати різні модулі) та opensource проєкти.

6.4 Сегментування ринку

Сегментування ринку – це процес поділу споживачів на групи з урахуванням різних принципів і факторів сегментації. У процесі сегментування визначаються сегменти ринку. Сегменти - це групи споживачів, що реагують на товар однаково. Сегмент ринку оцінюється за рядом характеристик (описує профіль сегмента): кількість потенційних споживачів, ємність ринку, темпи

зростання ємності по роках, споживчі ціни, прибутковість продажів і т.д. Загальний порядок сегментування ринку можна представити так:

- 1) визначення основних принципів та факторів сегментації для даного виробу (спочатку розглядаються географічні чинники, а потім демографічні й інші);
- 2) сегментування ринку: визначення складу груп споживачів, розподіл ринку на сегменти, опис профілю кожного сегмента, розрахунки щорічної ємності сегментів та всього ринку;
- 3) оформлення результатів сегментування у вигляді таблиць;
- 4) аналіз інформації про сегменти, вибір сегментів для подальшого аналізу (сегменти з незначною ємністю можна не розглядати);
- 5) позиціонування товару в обраних сегментах, визначення цільової ємності. Споживачами розроблюваного комплексу є різні підприємства України та інших країн. Основною характеристикою сегмента є його ємність - кількість виробів, які можуть бути продані за рік. Розрахунок повної ємності ринку буде проводитися за формулою:

$$S_{\text{полн.1}} = N_1 * Q_1 * m_1 ,$$

де N_1 – кількість підприємств споживачів виробу в і-тому сегменті;

Q_1 – середня річна програма виробів у і-тому сегменті, для яких буде поставлятися розглянутий товар;

m_1 – кількість комплектуючих виробів, що йдуть в один виріб-споживач (1 шт.).

Сегментування та розрахунок ємності ринку представлені в таблиці 6.1.

Таблиця 6.1 – Сегментування та розрахунок повної ємності ринку.

Сегменти ринку	Код сегмента	Кількість підприємств-споживачів даного товару	Середня річна програма виробів підприємств-споживачів	Повна ємність шт . / рік
Підприємства споживачі України	А	630	1,63	1026
Підприємства споживачі ближнього закордону	Б	229	1,52	348

Підприємства споживачі далекого закордону	В	199	1,21	240
Разом		1058	4,36	1614

Як видно з таблиці 6.1, повна місткість ринку становить 1614 шт. / рік. Це обґрунтовується тим, що в Україні багата кількістю технічних ВУЗів та високий розвиток в галузі робототехніки.

6.5 Розрахунок собівартості та ціни розробки мобільного колісного робота

Собівартість продукції складається з ряду позицій витрат. Сюди входять: розробники і їх денні оклади; трудомісткість робіт; основна заробітна плата; матеріали й покупні вироби.

Для розробки виробів необхідна участь наступних працівників, таких як: керівник, розробник, складальник, монтажник. Тривалість робочого місяця в середньому вважається 22 дня. Склад виконавців наведено в таблиці 6.2.

Оклад складу виконавців робіт становить: керівник - 690 грн., конструктор - 600 грн., монтажник - 509 грн., складальник - 458 грн.

Розрахунок трудомісткості робіт для розробки МРК представлений в таблиці 6.3.

Таблиця 6.2 – Склад виконавців роботи

Посада	Посадові оклади, грн	
	Денні	Місячні
Керівник	690	25180
Конструктор	600	19200
Монтажник	509	14198

Складальник	458	11076
Разом	2257	49654

Таблиця 6.3 – Розрахунок трудомісткості робіт.

Види робіт	Тривалість, дн	Трудомісткість, люд / дні	Виконавець			
			Керівник	Конструктор	Складальник	Монтажник
Постановка задач	1	2	+	+		
Розробка ТЗ	2	4	+	+		
Погодження та затвердження ТЗ	1	2	+	+		
Розробка ескізного проекту	5	10	+	+		
Комплектування (покупка \ доставка необхідних деталей)	7	7			+	
Збірка і монтаж комплектації	2	4			+	+
Збори продукту	3	9		+	+	+
Налагодження продукту	1	1				+
Перевірка відповідності продукту	1	2	+	+		
Випробування і здача продукту в експлуатацію	4	12		+	+	+
Разом	27	53	10	17	16	9

Після обчислимо основну заробітну плату безпосередньо виконавців розробки мобільного робота, з урахуванням трудовитрат, кількості виконавців та середньоденної заробітної плати:

$$\text{ОЗП} = \sum Ni * \text{ЗПср} \quad , \quad (6.1)$$

де N_i – кількість днів, відпрацьовані i -ми виконавцями за стадіями, ЗПср – денні оклади i -х виконавців.

Таким чином, основна заробітна плата за місяць згідно формули (6.1) становить:

$$\text{ОЗП} = 690*10+600*17+509*16+458*9 = 29875 \text{ грн.}$$

Таким чином, основна заробітна плата за місяць становить 24611 грн.

Зробимо розрахунок додаткової заробітної плати, яка дорівнює 15% від основної заробітної плати:

$$\text{ДЗП} = 15\% * \text{ОЗП} \quad (6.2)$$

Отримаємо по формулі (6.2) ДЗП яка дорівнює:

$$\text{ДЗП} = 0,15*29875 = 4481,25 \text{ грн.}$$

Після цього розрахуємо вартість покупки виробів і матеріалів, необхідних для виготовлення безпілотного літального апарату. Перелік покупних виробів складається з урахуванням переліку всіх комплектуючих елементів апарату таблиця 6.4. Ціни наведені у гривнях.

Таблиця 6.4 – Перелік покупних виробів і матеріалів.

№	Назва матеріалів	Кількість виробів, шт.	Ціна, грн.
1	Механізована платформа	1	1810
2	Одноплатний комп'ютер	1	1530
3	Датчик наближення	1	146
4	Комутаційні дроти	1	190
5	Акумулятор	1	350
6	Камера	1	120
Ціна покупних виробів			4146

Таким чином, вартість покупних виробів для мобільного робота становить 4046 грн.

Для виготовлення робота використовувалося певне обладнання, вартість якого наведено в таблицю 6.5.

Таблиця 6.5 - Вартість основних засобів.

Основні засоби	К-сть,шт	Цена за од., грн.	Цена, грн.
Робочий стіл з оснащенням	4	510	2040
Лампа	4	155	620
Робочий стілець	5	187	935
ПК	3	13800	41400
Паяльна установка	2	590	1180
Всього, $C_{п}$	18		47175

Також зробимо розрахунок амортизації (A_m), яка дорівнює 20% від $C_{п}$.

$$A_m = 25\% * \frac{C_{п}}{N_{дней}} * P_r$$

де $C_{п}$ – вартість обладнання,
 $N_{днів}$ – кількість робочих днів в році,
 P_r – тривалість виконання роботи.

Отже, амортизація дорівнює:

$$A_m = 0,20 * \frac{47175}{264} * 53 = 1954 \text{ грн.}$$

Після чого, складемо таблицю собівартості і ціни робота за статтями, що наведена в таблиці 6.6.

Таким чином, ціна робота без урахування ПДВ становить - 71717,56 грн., а з урахуванням ПДВ - 87061,07 грн.

Рентабельність продукції (або норма прибутку) - це відношення загальної суми прибутку до витрат виробництва і реалізації продукції, дорівнює:

$$P_n = \frac{Ц - С}{С} * 100\% = \frac{86061,07 - 59764,63}{59764,63} * 100\% = 44\%$$

де $Ц$ – ціна одиниці виробу; $С$ – собівартість одиниці виробу.

Таблиця 6.6 – Собівартість і ціна за статтями.

№	Статті	Сума, грн.	Примітки
1	Основна заробітна плата (ОЗП)	29875	
2	Додаткова заробітна плата (ДЗП)	4481,25	15% від ОЗП
3	Єдиний соціальний податок	7558,38	22%*(ОЗП+ДЗП)

4	Матеріали й куплені вироби	4146	Із табл. 7.4
5	Амортизація	1954	25% от Сп/ Нднів* Пр
6	Поза виробничі витрати	12950	40% від ОЗП
7	Собівартість (С)	59964,63	п.1+п.2+п.3...п.6
8	Прибуток (П)	12952,93	20% від С
9	Ціна без ПДВ	71717,56	П + С
10	ПДВ	15343,51	20% від ціни без ПДВ
11	Ціна з ПДВ	87061,07	п.9 + п.10

Точка беззбитковості – це фінансовий показник, розмір якого визначає необхідний обсяг продажів для стабільної та ефективної роботи підприємства без отримання збитків і прибутку.

По суті, точка беззбитковості є так званим критичним рівнем виробництва. При досягненні цього показника, прибуток, а також збитки, стають рівні нулю.

Точка беззбитковості має важливе значення при оцінці фінансового стану компанії. Перевищення обсягів виробництва і продажів над точкою беззбитковості вказує на фінансову стійкість компанії.

Показник точки беззбитковості розраховується за допомогою формули:

$$ТБ = \frac{Ц_{бн}}{Ц - С} = \frac{603285,36}{86061,07 - 59764,63} = 96,0356$$

Точка беззбитковості зображена на рис. 6.1.

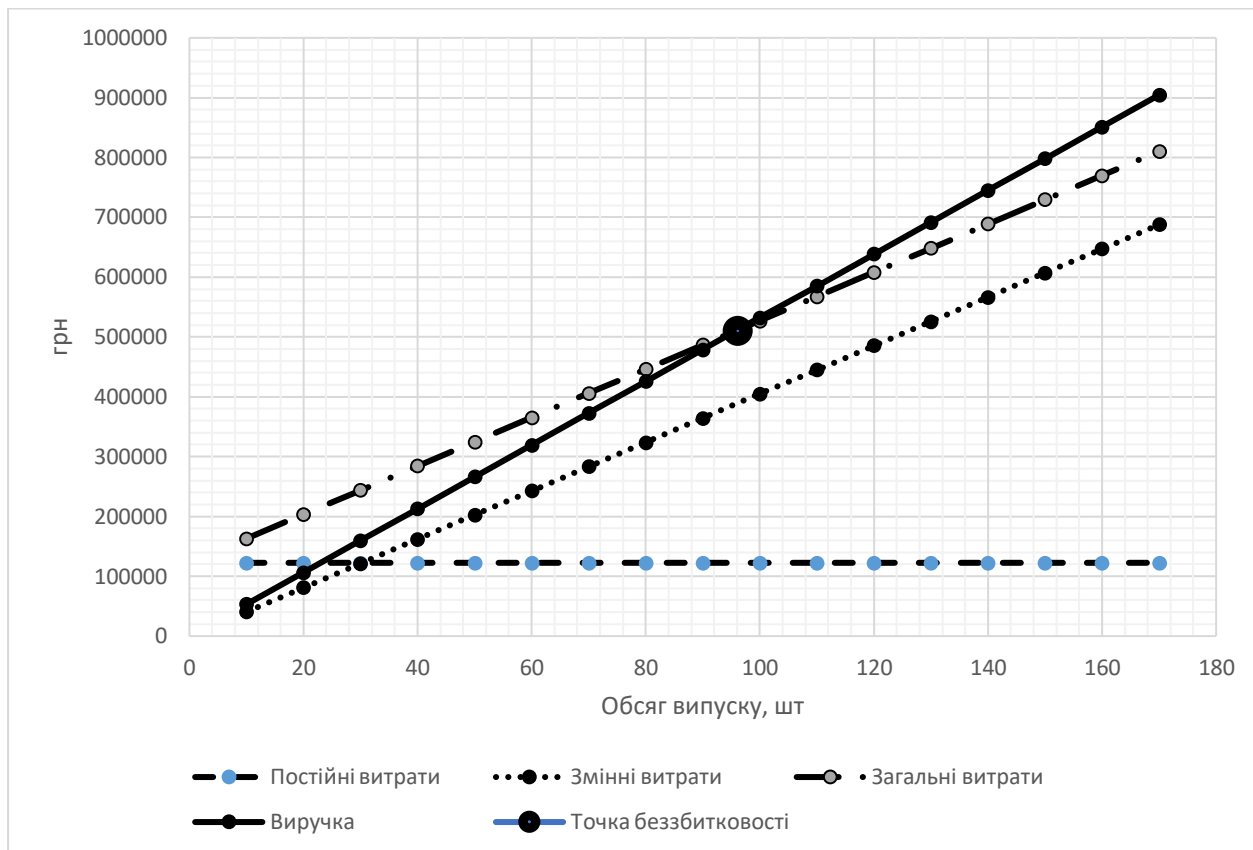


Рисунок 6.1 – Графік точки безбитковості

6.6 Висновок

Внаслідок економічних розрахунків можна здійснити висновки: розроблений робот є конкурентоздатним як у сфері спеціальної техніки для дослідження закритих просторів, так і на ринку радіомоделей для розваг, оскільки рівень якості розробленої системи порівняно з продуктами конкурентів перевищує 1,35. Продукт має достатню кількість потенційних покупців, щоб покрити всі витрати, пов'язані з розробкою, збиранням, впровадженням й подальшим ремонтом й заміною. Розрахована собівартість продукту складає 5321 грн. з урахуванням ПДВ. Проект стане прибутковим після продажу 96 екземплярів, а з урахуванням того, що щомісячне замовлення становить 139 одиниць, безбитковість компанії наступить менше ніж через місяць.

ВИСНОВКИ

Під час виконання дипломного проекту було проведено аналітичний аналіз, а також оцінка та синтез системи контролю руху транспортного робота; виконаний відбір елементної бази для проектування та створення моделі мобільного робота, проведено напівнатурні та натурні випробування для розробленої системи, розраховано собівартість запропонованої схеми.

Під час виконання розділу формування пристрою автоматичного управління були визначені базові задачі руху для колісного мобільного робота. Розглянуто властивості при отриманні моделей та управлінні колісним мобільним роботом, зокрема сформовано закони управління по точкам і по заданій траєкторії. Проведено лінеаризацію для законів зворотного зв'язку в системі колісного робота. Розглянуті ланцюгові форми структури канонічних моделей роботів. Запропоновано й спроектовано модель колісного робота, наведено її опис фізичних характеристик. Описано будову системи управління мобільним роботом й кінематику мобільного робота. Для подальших досліджень вже для конкретної моделі робота було розраховано контрольні обмеження та синтезовано закони управління для виконання руху по заданій траєкторії з перешкодами. Проведено синтез математичної моделі колісного робота.

Під час виконання конструкторської частини дипломного проектування було обрано версію робота AlphaBot. Вибір здійснювався шляхом порівняння характеристик між подібних роботизованих платформ, до яких відносяться AlphaBot-Ar-Basic й AlphaBot-Ar-Bluetooth. Також виконано відбір плати управління.

Для вирішення завдань розробки системи об'єктивного контролю руху транспортного робота під час роботи над дослідницькою частиною дипломної роботи було виконано моделювання роботи системи й моделювання стабілізації стану для плавної і несправної зміни часу регулювання також для полярної системи координат. Крім того, була промодельована система у стані лінійності оберненого зв'язку.

У частині формування пристрою автоматичного управління були розроблені методи і алгоритми для навігації мобільного робота з використанням об'єктивної інформації. Алгоритми ґрунтуються на інформації, отриманій від покадрової обробки відеокамери.

Проведено економічне обґрунтування розробки запропонованої системи..

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Букін, А. Г. A computer vision system for navigation of ground vehicles: Hardware and software [Text] / А. С. Лучагов, Р. Н. Садеков, О. А. Славін // Gyroscopy and Navigation. – 2016. – №7. – Р. 66 – 71.
2. Beauregard, S. Omnidirectional pedestrian navigation for first responders [Text] / S. Beauregard // In Proc. 4th Workshop on Positioning, Navigation and Communication WPNC '07. – 2007. – Р. 33 – 36.
3. Luinge, H. J. Inclination measurement of human movement using a 3D accelerometer with autocalibration [Text] / H. J. Luinge, P. H. Veltink // – 2004. – №12(1). – Р. 112–121.
4. David, J. Calibration and data fusion solution for the miniature attitude and heading reference system [Text] / J. David, J. Marko, K. Roman, T. Marko // Sensors and Actuators A: Physical. – 2007. – № 138(2). – Р. 411 – 420.
5. Roetenberg, D. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation [Text] / H. J. Luinge, C. T. M. Baten, P. H. Veltink // – 2005. – № 13(3). – Р. 395 – 405.
6. John, L. Crassidis Unscented filtering for spacecraft attitude estimation [Text] / John L. Crassidis, Landis F. Markley // Journal of guidance, control, and dynamics. – 2003. – № 26(4). – Р. 536 – 542.
7. Gebre-Egziabher, D. Design of multi-sensor attitude determination systems [Text] / D. Gebre-Egziabher, R. C. Hayward, J. D. Powell // Aerospace and Electronic Systems, IEEE Transactions. – 2004. – № 40(2). – Р. 627 – 649.
8. Phuong, N. H. Q. A DCM based orientation estimation algorithm with an inertial measurement unit and a magnetic compass [Text] / N. H. Q. Phuong, H. J. Kang, Y. S. Suh, Y. S. Ro // Journal of Universal Computer Science. – 2009. – № 15(4): – Р. 859 – 876.
9. Daniel, Ch. Novel methods for attitude determination using vector observations. PhD thesis. Institute of Technology. Israel. – 2003.
10. Baten, C. T. M. Magnetic distortional motion labs, implications for validating inertial magnetic sensors [Text] / C. T. M. Baten, F. C. T. Van Der Helm, W.

H. K. de Vries, H. E. J. Veeger // *Gait & Posture*. – 2009. – № 29(4). – P. 535 – 541.

11. Hyde, R. A. Estimation of upper-limb orientation based on accelerometer and gyroscope measurements / R. A. Hyde, L. P. Ketteringham, S. A. Neild, R. J. S. Jones. – 2008. – № 55(2). P. 746 – 754.

12. Mahony, R. Nonlinear complementary filters on the special orthogonal group [Text] / R. Mahony, T. Hamel, J. M. Pflimlin // *Automatic Control, IEEE Transactions on*. – 2008. – № 53(5). P. 1203 – 1218.

13. Bachmann, E. R. An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors [Text] / E. R. Bachmann, Xiaoping Yun, and C. W. Peterson // *IEEE International Conference on Robotics and Automation ICRA '04*. – 2004. – № 2. – P. 1115 – 1122.

14. Itzhack, Y. Bar-Itzhack New method for extracting the quaternion from a rotation matrix [Text] / Itzhack Y Bar-Itzhack // *AIAA Journal of guidance, control and dynamics*. – № 23(6). – 2000.

15. Rong Zhu A. Real-Time Articulated Human Motion Tracking Using Tri-Axis Inertial/Magnetic Sensors Package [Text] / Rong Zhu, Zhaoying Zhou // *IEEE Transaction on neural systems and rehabilitation engineering*. – 2004. – № 2.

16. Seong, Y. Calibration Technique for a Two-Axis Magnetic Compass in Telematics Devices [Text] / Y. Seong, A. Chan Gook Park // *ETRI Journal*. – 2005. – № 3.

17. Adam, N. Bingaman, Tilt-Compensated Magnetic Field Sensor [Text] / N. Adam Bingaman // *Virginia Polytechnic Institut and State University*. – 2010.

18. P.S. Geidarov, Algorithm for determining the location and dimensions of objects based on analysis of image objects [Текст] / Geidarov P.S. // *Компьютерная оптика*, – 2010. – № 2(35). – P. 276 - 280.

19. Grewal, M. Global positioning systems, inertial navigation, and integration [Text] / M. Grewal, L. Weill, A. Andrews // *John Wiley & Sons*. – 2001.

20. Kallapur, A. A robust gyroless attitude estimation scheme for a small fixed-wing unmanned aerial vehicle [Text] / I. Petersen, S. Anavatti. – 2009. – P. 666 – 671.

21. Watanabe, Y. Simultaneous Visual Target Tracking and Navigation in a

GPS-Denied Environment [Text] / Y. Watanabe, P. Fabiani, G. Le Besnerais // In International Conference on Advanced Robotics. Munich. Germany. – 2009.

22. Garratt, M. A. Visual Control of an Autonomous Helicopter [Text] / M. A. Garratt, J. S. Chahl // 41st Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics. Reno. Nevada. – 2003.

23. Morel, J. M. ASIFT: A New framework for fully affine invariant image comparison [Text] / J. M. Morel, G. Yu // SIAM Journal on Imaging Sciences. – 2009. – P. 438 – 469.

24. Канканхалі, А. Сегментація потокового відео [Текст] / А. Канканхалі, В. Смоляр // Мультимедійні системи. – 2003. – С. 10 – 28.

25. Yu, G. A Fully affine invariant image comparison method [Text] / G. A Yu, J. M. Morel // Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). – 2009

26. Liu, G. Content-based image retrieval using color difference histogram [Text] / G. Liu, J. Yang. – 2013. – №1.