

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 системи управління мобільної колісної платформи та дослідження показників якості управління траєкторним рухом

ХАІ.301.362.24О.151.183018 ПЗ

Виконав: студент 6 курсу, групи 362

Галузь знань 15 «Автоматизація та приладобудування»

Спеціальність

151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітня програма

“Інженерія мобільних додатків”

Ринченко В.О.

(прізвище та ініціали студента)

Керівник Гавриленко О. В.

(прізвище та ініціали)

Рецензент Трубчанінова К.А.

(прізвище та ініціали)

Харків – 2024 рік

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 (№301)
(повне найменування)
Рівень вищої освіти другий (магістерський)
Галузь знань 15 Автоматизація та приладобудування
(код та найменування)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код та найменування)
Освітня програма Інженерія мобільних додатків
(найменування)

ЗАТВЕРДЖУЮ
Завідувач кафедри

к.т.н., доц. _____ Костянтин ДЕРГАЧОВ

“ _____ ” _____ 2024 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ринченко Владислава Олександровича

(прізвище, ім'я та по батькові)

1. Тема кваліфікаційної роботи Розробка системи управління мобільної колісної платформи та дослідження показників якості управління траєкторним рухом

керівник кваліфікаційної роботи Гавриленко Олена Володимирівна, доц., к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом Університету 6.11.2023 року № 1968-уч.

2. Термін подання здобувачем кваліфікаційної роботи 9 січня 2024 року

3. Вихідні дані до роботи Розробка системи управління мобільної колісної платформи та дослідження показників якості управління траєкторним рухом

4. Зміст пояснювальної записки (перелік завдань, які потрібно розв'язати) Оцінка стану проблеми, вербальний і математичний опис об'єкту автоматичного управління, синтез системи траєкторним рухом, напівнатурне моделювання системи моніторингу траєкторним рухом, розрахунок собівартості виготовлення системи.

5. Перелік графічного матеріалу 10 слайдів для презентації та захисту роботи

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стан проблеми	доц. Гавриленко О. В.	12.09.23	17.01.24
Аналіз і синтез СУ	доц. Гавриленко О. В.	12.09.23	17.01.24
Конструктор. частина	доц. Гавриленко О. В.	12.09.23	17.01.24
Технологічна частина	доц. Гавриленко О. В.	12.09.23	17.01.24
Експ.-практ. частина	доц. Гавриленко О. В.	12.09.23	17.01.24
Економічне обґрунтування	доц. Гавриленко О. В.	12.09.23	17.01.24

Нормоконтроль _____ доц. Гавриленко О. В. «22» січня 2024 р.
 (підпис) (ініціали та прізвище)

7. Дата видачі завдання 12.09.2023 р

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1.	Формулювання теми проекту.	01.09.2023	
2.	Розробка технічного завдання	12.09.2023	
3.	Огляд стану проблеми і патентний пошук. Математичний опис системи управління.	20.10.2023	
4.	Аналіз і синтез системи управління	15.12.2023	
5.	Конструкторська частина роботи. Технологічна частина роботи. Економічне обґрунтування розробки	08.01.2024	
6.	Експериментально-практична частина	08.01.2024	
7.	Оформлення розрахунково-пояснювальної записки і графічного матеріалу. Попередній захист роботи	12.01.2024	
	Рецензування проекту. Захист проекту в ЕК	18.09.2024	

Здобувач _____ Владислав РИНЧЕНКО

Керівник роботи _____ Олена ГАВРИЛЕНКО

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

Кафедра систем управління літальних апаратів (№301)

«ЗАТВЕРДЖУЮ»

Завідувач кафедри 301

к.т.н., с.н.с., доцент

_____ Костянтин ДЕРГАЧОВ

«__» _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на дипломне проектування

Ринченко Владислава Олександровича

- 1 Тема роботи: Розробка системи управління мобільної колісної платформи та дослідження показників якості управління траєкторним рухом
затверджена наказом по університету від «6» листопада 2023 р. № 1968-уч.
- 2 Строк здачі студентом закінченої роботи « 9 » _____ січня _____ 2024 р.
- 3 Область застосування розробки: всі області промисловості, що вимагають застосування рухомих транспортних роботів, а також військова сфера (роботи-розвідники для отримання інформації у важких кліматичних і бойових умовах, роботи для розмінування і завдання прямих вогневих ударів по стратегічно важливих об'єктах супротивника), прикладні робототехнічні системи наведення і пошуку цілі, що реалізують заданий алгоритм наведення (пошук і ліквідація снайперів в бойових операціях, пошук людей після землетрусів і інших техногенних і природних катастроф).

4 Початкові дані для розроблювальної системи

4.1 Призначення і мета створення системи: дослідження траєкторного руху

4.2 Загальні відомості: система управління мобільної колісної платформи для дослідження траєкторного руху

5 Технічні вимоги до каналів системи управління

5.1 Питання, що підлягають розробці: формування лінеаризованої математичної моделі об'єкта управління, вибір виконавчого та вимірювального пристроїв, розробка функціональної і структурної схем, вибір і узгодження вимірювальних пристроїв, градування вимірювальних каналів, розрахунок перетворювального і підсилювального блоку, вибір мікропроцесорного пристрою, розробка алгоритмів управління системи

5.2 Режим роботи системи (безперервний, циклічний, одноразової дії): безперервний

5.3 Показники якості системи управління: час перехідного процесу для підсистеми управління приводом платформи – 1 с; перерегулювання не більше 5%; смуга пропускання системи не менше 0,01 рад/сек; показник коливальності не більше 1,1; стала статична похибка управління системи не більше 0,5 %; час реагування системи на збурюючі і вимірювальні сигнали з датчиків не більше 10 мс.

5.4 Вимоги до приладового складу системи: для підсистеми управління приводом платформи – датчики кутової швидкості на привідних осях; навігаційні датчики – ультразвуковий дальномір, GPS. Система припускає розширення вимірювального блоку – має вільні роз'єми через які можливе підключення різних датчиків до мікропроцесорної системи, з попереднім завантаженням програми, що реалізовує закон (алгоритм) управління.

5.5 Вимоги до взаємозамінності блоків: у даній системі взаємозамінні блок датчиків, мікроконтролер серії ESP32, 4 двигуни постійного струму (привід системи), підсилювальний і перетворювальний блоки датчиків.

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 $T = (-10..+60)$ С, вологість повітря $F = (0..60)\%$, ступінь запиленості $G = 0,2,0,6$; склад хімічно активних компонентів в повітрі відсутній

6.2 Механічні вимоги (вібрація, тряска, можливі перекоси, удари, нахили і т.ін.): можливі перекоси, можлива тряска, нахили.

6.3 Наявність перешкод (електричні наведення радіоперешкоди, магнітні впливи): можна знехтувати

6.4 Електричні параметри системи (напруга джерел живлення, потужність, стабільність, частота): напруга живлення 5 - 8.6В

7 Додаткові функції, реалізовані системою (сигналізація про несправності, реєстрація необхідної інформації, самоконтроль самої системи і т.ін.): самоконтроль самої системи управління реалізується програмою управління шляхом перевірки наявності вимірювального блоку в мікроконтролері системи, реєстрація інформації про поточну роботу системи в пам'яті мікроконтролера.

8 Обсяг виконуваних розроблювачем робіт

8.1 Етапи проведення роботи:

I етап: Оцінка стану проблеми

II етап: Вербальний і математичний опис об'єкту автоматичного управління.

III етап: Синтез системи управління траєкторним рухом

IV етап: Моделювання системи моніторингу траєкторним рухом.

8.2 Обсяг розробки по кожному етапу: I етап: 8 стор. II етап: 17 стор. III етап: 10 стор. IV етап: 18 стор.

9. Параметри обладнання системи

9.1 Габарити: розмір системи (довжина, ширина, висота) – не більше $0,24 * 0,14 * 0,12$ м

9.2. Маса: не більше 2 кг

9.3. Вимоги до конструктивного виконання і розміщенню: всі вказані вище блоки кріпляться на одній рухомій платформі; платформа приводиться до руху чотирма двигунами постійного струму, платформа кріпиться на 4 колеса(по 1 на кожній осі), колеса мають 1-у міру свободи.

9.4. Інші вимоги бічний рух платформи здійснюється за рахунок зміни різниці кутових швидкостей обертання 4 двигунів. Кутові швидкості обертання регулює мікроконтролерний блок системи (реалізація управління двигуном ШІМ-імпульсами). Підсилювальний, перетворювальний і мікропроцесорний блоки повинні розміщуватися на одному печатному вузлі _____

10. Вимоги безпеки бічний рух платформи здійснюється за рахунок зміни різниці кутових швидкостей обертання 4 двигунів. Кутові швидкості обертання регулює мікроконтролерний блок системи (реалізація управління двигуном ШІМ-імпульсами). Підсилювальний, перетворювальний і мікропроцесорний блоки повинні розміщуватися на одному печатному вузлі _____

11. Технологічна частина

11.1. Розробка технологічного процесу включає розробку печатного вузла, на якому розміщуються всі елементи і пристрої системи, окрім двигунів і вимірників _____

11.2. Умови і вимоги середній час виробництва печатного вузла – не більше 6 годин з попередньою підготовкою виробництва _____

11.3. Очікуваний результат технологічність виробництва печатного вузла роботів даного класу, дрібносерійний тип виробництва _____

12 Експериментально-практична частина: розробити алгоритм управління для його впровадження на контролері, а також розробити мобільний додаток для зчитування та передачі даних на мобільну платформу.

13 Економічна частина

13.1 Розробити (розрахувати, одержати): розрахувати собівартість роботи і одержати ціну розробки конструкції.

13.2 Умови і вимоги: Виконати проект в зазначені терміни

13.3 Очікуваний результат: робоча система керування мобільної платформи з передаванням результатів за допомогою мобільного додатку

14 Перелік графічних матеріалів із зазначенням форматів: 10 плакатів у вигляді слайдів

15 Мова підготовки пояснювальної записки (захисту): українська

Керівник роботи

доц. Гавриленко О. В.
(П. І. Б.)

Прийняв до виконання

Ринченко В. С.
(П. І. Б. студента)

« » _____ 2023 р.

« » _____ 2023 г.

Узгоджено за питаннями:

конструкції

доц. Гавриленко О. В.
(П. І. Б.)

дослідницької частини

доц. Гавриленко О. В.
(П. І. Б.)

« » _____ 2023 г.

« » _____ 2023 г.

економіки

доц. Гавриленко О. В.
(П. І. Б.)

« » _____ 2023 г.

РЕФЕРАТ

Пояснювальна записка містить 95 сторінок, 32 рисунків, 2 додатку, 11 джерел.

Тема роботи: Розробка системи управління мобільної колісної платформи та дослідження показників якості управління траєкторним рухом.

Мета роботи: Розробити і впровадити в дію оптимальну систему управління мобільної колісної платформи, які дозволять збільшити точність та стабільність траєкторного руху мобільної платформи, зменшити час реакції в системі.

Предмет роботи: система управління мобільною платформою, що виконує завдання по переміщення, по певній траєкторії.

У дипломній роботі було розроблено систему управління для переміщення мобільної колісної платформи, та дослідження якості управління її руху по заданій траєкторії, управління якої здійснювалося за допомогою мобільного додатку.

Результатом роботи є розроблена система управління мобільною колісною платформою, з можливістю керування та отримання даних за допомогою мобільного додатку.

КЛЮЧОВІ СЛОВА: КЕРУВАННЯ, МОБІЛЬНА КОЛІСНА ПЛАТФОРМА, МОБІЛЬНИЙ ДОДАТОК, ОПИС ФУНКЦІОНУВАННЯ МОБІЛЬНОЇ ПЛАТФОРМИ, АВТОМАТИЧНЕ УПРАВЛІННЯ МОБІЛЬНОЇ ПЛАТФОРМИ, МАТЕМАТИЧНИЙ ОПИС МОБІЛЬНОЇ ПЛАТФОРМИ.

СПИСОК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

МКП – мобільна колісна платформа;

ОАУ – об'єкт автоматичного управління;

AS - Android Studio;

САУ – система автоматичного управління;

МД – мобільний додаток.

ЗМІСТ

ВСТУП	13
1 ОЦІНКА СТАНУ ПРОБЛЕМИ УПРАВЛІННЯ МОБІЛЬНИХ КОЛІСНИХ ПЛАТФОРМ ТА ДОСЛІДЖЕННЯ ЯКОСТІ ТРАЄКТОНОГО РУХУ	14
1.1 Огляд літератури щодо існуючих рішень	14
1.2 Постановка задач	17
1.3. Висновки	17
2 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ І СИНТЕЗ СИСТЕМИ УПРАВЛІННЯ	18
2.1 Опис загального функціоналу системи	18
2.2 Отримання та опис нелінійних математичних моделей	19
2.3 Розробка функціональної та структурної схем	25
2.4 Статичний розрахунок	27
2.5 Можливість зміни швидкості робота	29
2.6 Висновки	39
3 КОНСТРУКТОРСЬКА ЧАСТИНА	40
3.1 Підбір комплектуючих	40
3.2 Основні характеристики мобільної платформи	44
3.3 Аналіз впливу параметрів на рух по заданій траєкторії	45
3.4 Розробка рухомої платформи.	45
3.5 Висновки	50
4 ДОСЛІДНИЦЬКА ЧАСТИНА	51
4.1 Розробка моделей об'єкта	51
4.2 Методи проведення досліджень	53
4.3 Програма проведення досліджень	53
4.4 Результати досліджень	59
4.5 Аналіз результатів досліджень	60
4.6 Висновки	60
5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА	62
5.1 Формування алгоритму роботи системи керування	62
5.2 Розробка мобільного додатку	62
5.3 Розробка системи зв'язку	65
5.4 Опис роботи додатку	66
5.5 Висновки	66
6 ЕКОНОМІЧНА ЧАСТИНА	67
6.1 Мета економічної частини	67

	12
6.2 Трудомісткість виконання роботи	67
6.3 Розрахунок витрат на виконання роботи	67
6.4 Визначення можливої (договірної) ціни роботи	71
6.5 Висновки	72
ЗАКЛЮЧЕННЯ	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	74
ДОДАТОК А	75
ДОДАТОК Б	79

ВСТУП

Сучасний світ вимагає постійного розвитку та впровадження інновацій у різні галузі, такі як сфера мобільної робототехніки. Одна з важливих областей досліджень являється розробка систем управління мобільними колісними платформами та вдосконалення їх траєкторного руху. Питання автоматизації управління та оптимізації руху є досить актуальним, а навіть з плином часу лише загострюється. Якщо розгляду чим займають технологічні гіганти такі як Tesla чи Amazon. Вони розвивають автопілоти 5 покоління, які не потребують участі людини.

Актуальність теми зумовлена потребою у удосконалення автоматизованих логістичних систем, розширення та пришвидшення виробництва у галузях медицини, промисловості та інших галузях. Дослідження у цій галузі має великий потенціал для майбутніх систем.

У цьому контексті, дана робота може дослідити можливості даних систем, але зі зменшеним масштабом, а також бюджеті. Що дозволяє оцінити трудомісткість та вартісність даного напрямку та проекту в цілому.

Отже, робота спрямована на вирішення актуальних завдань в галузі автоматизованих систем управління та робототехніки, що може сприяти подальшому розвитку та впровадженню сучасних технологій у відповідні галузі промисловості та науки.

На наступних сторінках даної роботи буде описані тонкощі розробки мобільної платформи, розробки системи управління, та розробки програмного забезпечення для коректного працездатності та виконання поставленого завдання з отримання даних, та оцінки якості керування.

1 ОЦІНКА СТАНУ ПРОБЛЕМИ УПРАВЛІННЯ МОБІЛЬНИХ КОЛІСНИХ ПЛАТФОРМ ТА ДОСЛІДЖЕННІ ЯКОСТІ ТРАЄКТОНОГО РУХУ

1.1 Огляд літератури щодо існуючих рішень

У світі відбувається стрімке зростання темпу застосування роботів у промисловості та побуті. За останній рік у світі було продано понад 630 тисяч промислових роботів, і це число зростає [1]. Більшість впроваджених роботів – це роботи, що застосовуються у приміщенні, тобто Indoor типу. Обсяги світового ринку Indoor роботів також стрімко зростають і до 2026 досягнуть 50 мільярдів доларів [2]. У існуючу статистику включені всі промислові роботи indoor типу, проте більша частина з них – це мобільні робототехнічні платформи, які впроваджуються у складські приміщення та туди, де особливо важлива швидкість та точність перевезення вантажів.

Повсюдно відбувається заміщення людини роботами та залучення мобільних роботів для вирішення завдань, які раніше вирішувалися без їх використання. Процес застосування мобільних роботів виробництва дуже трудомісткий. Необхідно розробити систему управління цим роботом, а також підготувати інфраструктуру приміщення, в якому експлуатуватиметься мобільний робот. Усі ці завдання входять у поняття побудови програмно-апаратного забезпечення управління мобільним роботом.

Основною перевагою мобільних робототехнічних платформ є їхня універсальність. Одні й самі мобільні платформи можуть бути застосовані для вирішення різних завдань, наприклад, транспортування вантажів і побудови карти місцевості. Залежно від власних характеристик для кожної мобільної платформи може бути створена система управління для вирішення конкретного завдання.

Для того, щоб познайомитися з існуючими рішеннями щодо впровадження роботів на виробництво та склади, було розглянуто такі системи: складські роботи Kiva Amazon Robotics, Toru Cube, SBR robot. Всі з перелічених роботів застосовуються для роботи в складських приміщеннях, проте базуються на різних алгоритмах та методах навігації у просторі, потребують різноманітних інфраструктурних та програмних впроваджень. Їх особливості, що цікавлять, описані нижче.

Мобільні роботи Kiva, що використовуються на складах Amazon, є лідерами за кількістю застосовуваних зараз. Лише у 2017 році їх налічувалося близько 45 тисяч. Роботи Kiva здатні переміщати вантажі вагою до 317 кг, оснащені неповоротним колесами і платформою, що обертається, і орієнтуються в просторі за спеціальною розміткою на підлозі і їх бортовий комп'ютер оснащений операційною системою для роботів ROS [3]. На рис.1.1 представлено зображення мобільних роботів Kiva.



Рисунок 1.1 – Складські роботи Kiva

Мобільний робот TORU Cube так само, як і попередні роботи, має вбудовану ROS, але в іншому значно відрізняється від розглянутих вище. Його система управління не потребує додаткового оснащення приміщення спеціальними мітками, робот переміщається за збудованою заздалегідь картою та за командами оператора, що не може забезпечити повну автономність. Система безпеки робота уникає зіткнення з перешкодами чи людиною у робочій зоні за рахунок лідерів на борту робота. Особливістю даної системи є можливість постійного оновлення карти для навігації роботів за рахунок їх з'єднання в локальну бездротову мережу. Таким чином, якщо один із роботів виявив зміну на складі або нову перешкоду, дані про це вносяться на загальну карту [4]. Мобільні роботи TORU представлені рис.1.2.



Рисунок 1.2 – Мобільний робот TORU Cube

Робот SBR від Української компанії SB Robotics зібрано на базі складського робота компанії "Стандарт-ПАК" [5]. Ці роботи призначені для перенесення вантажів вагою до 23 кілограм. Роботи SBR позиціонуються як функціональні роботи-перевізники. За рахунок малої вантажопідйомності їм складно знайти застосування на складах, проте ці роботи хороші для автоматизованої кур'єрської доставки, як робота офіціанта і помічників у перенесенні негабаритних вантажів. Для переміщення всередині приміщення роботу потрібна розмітка у вигляді спеціальних напрямних ліній. Однак, оскільки робот оснащений мікроконтролером Arduino, його навігація може здійснюватися за рахунок приєднання додаткових датчиків та настройки їх обробки за допомогою Arduino. Зображення робота SBR наведено рис. 1.3.



Рисунок 1.3 – Робот SBR

Основною перевагою більшості розглянутих систем є наявність попередньо встановленої спеціалізованої конкретної моделі операційної системи для робота. ROS дозволяє легко переналаштувати роботи під виконання конкретних завдань, що значно розширює сферу застосування кожної мобільної платформи, а її поширеність дозволяє розробникам систем керування не підлаштовуватися під конкретну модель мобільного робота.

З огляду існуючих рішень з управління мобільними робототехнічними платформами на складах, промисловості та інших цілей було виділено недолік, властивий як розглядуваним системам, а й більшості існуючих над ринком рішень. Основним та найпоширенішим недоліком є необхідність підстроювання інфраструктури приміщення для впровадження в нього мобільної платформи. До таких методів підстроювання відносяться QR-мітки на підлозі, напрямні лінії, штрих-коди та напрямні рейки. Все це значно ускладнює первинне впровадження роботів, а також їх подальшу експлуатацію, оскільки подібні

об'єкти в промислових приміщеннях вимагають постійного оновлення, контролю та не дають можливості миттєвих змін усередині приміщення.

1.2 Постановка задач

– Виходячи з виявленого в огляді аналогів недоліку, у рамках роботи ставиться за мету розроблення і дослідження програмно-апаратного комплексу управління мобільною колісною платформою. Для цього мета була розбита на такі завдання:

- розробити математичні моделі і структурну схему планованої системи управління;
- розробити функціональну схему системи керування;
- здійснити вибір програмних та апаратних компонентів;
- побудувати систему прийому-передачі даних між вибраними програмними та апаратними компонентами;
- дослідити математичну модель руху платформи по заданій траєкторії;
- розробити мобільний додаток для моніторингу параметрів руху системи.

1.3. Висновки

Пошук та аналіз інформації про існуючі рішення були ключовими кроками у розвитку мобільної платформи. Цей аналіз допоміг зрозуміти сучасний стан цієї галузі та визначити основні задачі роботи.

2 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ І СИНТЕЗ СИСТЕМИ УПРАВЛІННЯ

2.1 Опис загального функціоналу системи

Для грамотного вибору програмних та апаратних компонентів системи необхідно насамперед визначити вимоги до проектованої системи управління та скласти її структурну схему.

Під час проектування структурної схеми було проаналізовано такі вимоги:

- забезпечити рух мобільної платформи без використання додаткової інфраструктури у приміщенні;
- створити можливість введення в систему додаткових датчиків, сенсорів та виконавчих елементів;
- мінімізувати обчислювальні ресурси бортового комп'ютера мобільної платформи.

Для побудови системи управління мобільним роботом необхідно насамперед визначитися з вимогами до мобільного роботу. Робот повинен пересуватися в обмеженому просторі без перешкод, а його траєкторія може змінюватися тільки в межах даного простору. Таким чином, немає необхідності оснащувати робота датчиками навігації в приміщенні, для запобігання його зіткненню з можливими перешкодами. Однак, оскільки регулювання швидкості мобільної платформи носить основний характер, робот, крім двигунів, повинен бути оснащений енкодерами на кожному приводі. Для зчитування та обробки даних з енкодерів, а також перетворення управляючих команд на обертання двигунів, мобільна платформа повинна бути оснащена бортовим комп'ютером з вбудованою системою управління. Таким чином, блок мобільної платформи на структурній схемі повинен містити блок вбудованої системи управління, блок двигунів і блок енкодерів.

Щоб забезпечити рух платформи без використання додаткової інфраструктури, мобільний робот повинен бути оснащений рядом датчиків та здійснювати рух згідно з командами, що подаються системою управління. Оскільки система управління, здатна одночасно аналізувати дані з датчиків мобільної платформи і будувати траєкторію руху робота, що динамічно змінюється, вимагає великих обчислювальних ресурсів, а серед поставлених завдань потрібно мінімізувати обчислювальні ресурси вбудованого бортового комп'ютера, було прийнято рішення про впровадження в систему управління

мобільною платформою персонального комп'ютера, який аналізуватиме дані з датчиків мобільної платформи, а також із зовнішніх пристроїв і передаватиме на робота управляючі команди. Фізично зв'язок між персональним комп'ютером та мобільною платформою може здійснюватися провідним або бездротовим шляхом. У лабораторних умовах і при тестуванні системи управління зв'язок здійснюватиметься провідним шляхом, каналом, який передбачає обрана мобільна платформа, найчастіше це Ethernet. При експлуатації системи на виробництві для мінімізації додаткових пристроїв та проводів з'єднання та передача даних здійснюватиметься бездротовим шляхом за протоколом, який передбачає обрана мобільна платформа, в більшості випадків мобільні платформи оснащені Wi-Fi модулем.

Для встановлення зв'язку персонального комп'ютера та мобільної платформи, для формування сигналів керування для двигунів, а також для побудови програмної частини системи керування необхідно вибрати середовище розробки та методи чи моделі програмування. Робоче середовище може бути об'єднане в деякий програмний пакет, робота з яким значно спростить процес створення системи управління.

Однією з вимог до системи є можливість впровадження в систему зовнішніх датчиків та пристроїв. Для спрощення обробки сигналів зовнішніх пристроїв та можливості безперешкодного додавання в систему нових датчиків необхідно встановити в систему мікроконтролер, які виконуватиме ЦАП і АЦП функції, а також фільтрувати шуми. На стадії проектування та тестування мікроконтролер повинен передавати дані із зовнішніх датчиків на персональний комп'ютер, а також отримувати з нього команди для управління зовнішніми пристроями. Зв'язок між персональним комп'ютером і мікроконтролером здійснюється за протоколом, що підтримується мікроконтролером, в більшості випадків таким протоколом виступає USB. Надалі для спрощення експлуатації системи мікроконтролер необхідно буде з'єднати з бортовим комп'ютером мобільної платформи.

2.2 Отримання та опис нелінійних математичних моделей

Отримання лінеаризованої математичної моделі об'єкта для подальшого опису всієї системи стабілізації лінійної швидкості платформи. На рис. 2.1 представлено схему приводу рухомої платформи.

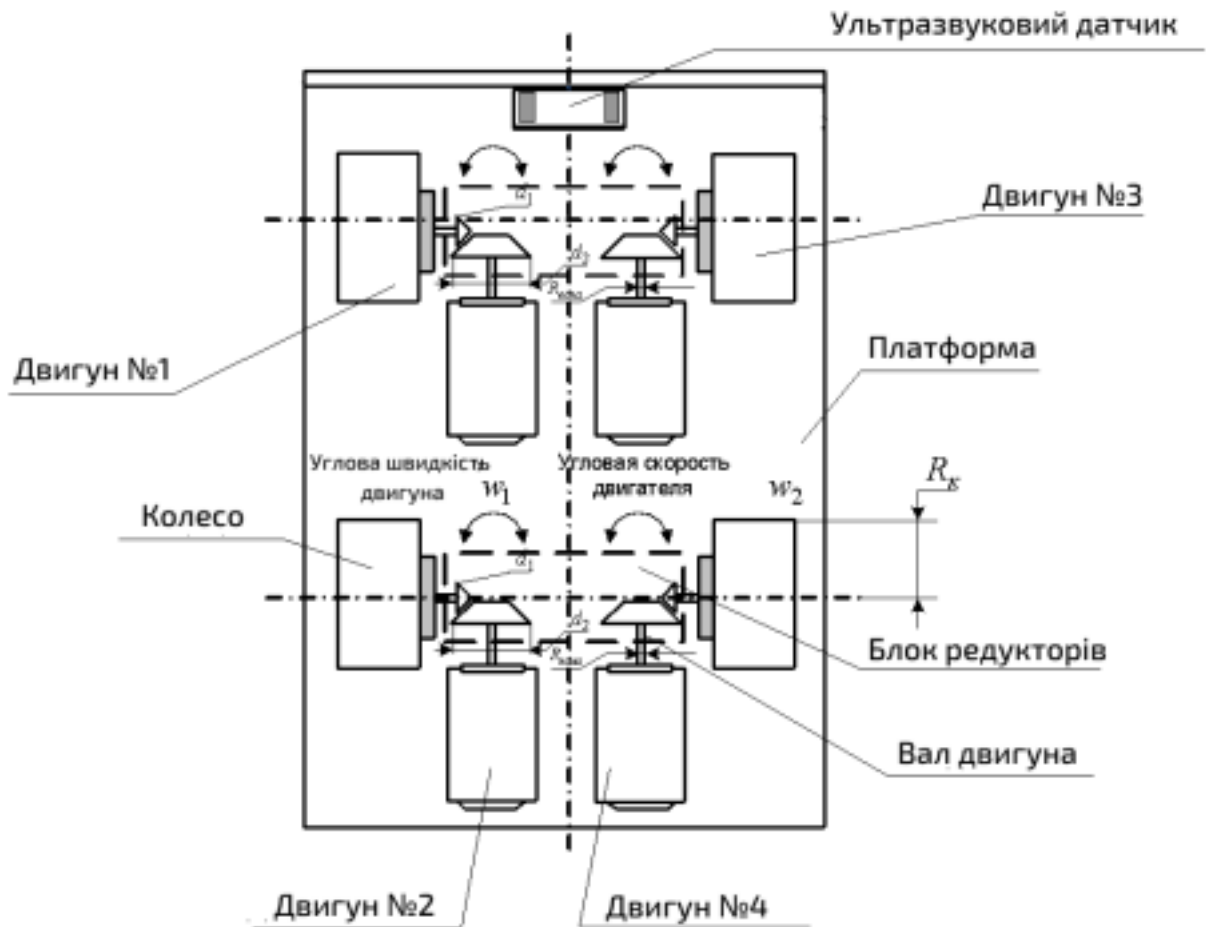


Рисунок 2.1 – Схема приводу рухомої платформи

На даній схемі позначено: розташування та кількість коліс, 4 блоків редукторів, кількість та тип датчиків.

$$W_{\partial в}(s) = \frac{w(s)}{U_{я}(s)} = \frac{k_{\partial}}{T_{ем}s + 1} = \frac{2.22}{0.2 + 1} \quad (2.1)$$

Де коефіцієнт двигуна:

$$k_{\partial} = \frac{\omega_{xx}}{U_{xx}} = \frac{6.66}{3} = 2.22. \quad (2.2)$$

А електромеханічна постійна часу:

$$t_{\text{пп}} = 3.4 T_{\text{ем}}; t_{\text{пп}} = 0.6 \text{ с}; T_{\text{ем}} = 0.2 \text{ с}, \quad (2.3)$$

Подальший етап передбачає побудову математичної моделі МКП, як об'єкту автоматичного управління. Для того щоб визначати просторове положення МКП скористаємося кінетичним рівнянням другого закону Ньютона, а саме:

$$F_{\text{дин}} = F_{\text{упр}} - F_{\text{супр}}, \quad (2.4)$$

Створимо математичну модель ОАУ - мобільної колісної платформи. Для визначення просторового положення слід скористатися кінематичними рівняннями.

$$m_{\text{пл}} \ddot{x} = F_{\text{упр}} \cdot Z - \frac{M_{\text{н}}}{R_{\text{к}}} \quad (2.5)$$

$$m_{\text{пл}} \dot{V} = F_{\text{упр}} \cdot Z - \frac{M_{\text{н}}}{R_{\text{к}}} \quad (2.6)$$

де V - вихід об'єкта управління, швидкість однієї з поздовжніх осей платформи.

Сила опору, що діє на колеса платформи:

$$F_{\text{супр}} = \frac{M_{\text{н}}}{R_{\text{к}}} \quad (2.7)$$

Керуюча сила:

$$F_{\text{упр}} = \frac{M_{\text{упр}}}{R_{\text{вала}}}. \quad (2.8)$$

Коефіцієнт передачі крутного моменту, від якоря двигуна на колеса рухомої платформи:

$$Z = \frac{r_{\text{я}}}{R_{\text{к}}} \cdot i_{12} = \frac{0.033 \cdot 1.3}{0.01} = 4.29. \quad (2.9)$$

Таблиця 2.1 - Фізичні параметри об'єкта управління та виконавчого механізму

Фізичний параметр	Чисельна величина
Кутова швидкість холостого ходу двигуна, $\frac{1}{c}$	6.66
Напруга холостого ходу двигуна, B	3
Маса платформи, кг	0.7
Маса якоря двигуна, кг	0.04
Радіус якоря двигуна, м	0.01
Радіус валу двигуна, м	0.001
Радіус колеса платформи, м	0.033
Внутрішній радіус колеса, м	0.022
Діаметр зубчастої передачі від валу двигуна, м	0.007
Діаметр зубчастої передачі на вал колеса платформи, м	0.009
Момент тертя кочення (тертя гуми про тверду поверхню) $M_n, H \cdot m$	0.32

Передавальне число редуктора:

$$k_p = i_{12} = \frac{\omega_1}{\omega_2} = \frac{d_2}{d_1} = \frac{0.009}{0.007} = 1.3. \quad (2.10)$$

Крутячий момент якоря двигуна:

$$M = J \cdot \frac{d\omega(t)}{dt} \quad (2.11)$$

Момент інерції якоря (якір приймаємо за цілісне циліндричне тіло):

$$J = \frac{m_{\text{я}} \cdot r_{\text{я}}^2}{2} \quad (2.12)$$

Динамічна сила руху платформи має вигляд:

$$F_{\text{упр}} = \frac{I}{R_{\text{валу}}} \cdot \frac{d\omega(t)}{dt} = \frac{m_{\text{я}} \cdot r_{\text{я}}^2}{2 \cdot R_{\text{валу}}} \cdot \dot{\omega}(t) \quad (2.13)$$

Знаходимо передатну функцію об'єкта управління:

$$m_{\text{нл}} \ddot{x}(t) + \frac{M_{\text{н}}(t)}{R_{\text{к}}} = \frac{m_{\text{я}} \cdot r_{\text{я}}^2}{2 \cdot R_{\text{вала}}} \cdot Z \cdot \dot{\omega}(t); \quad (2.14)$$

$$K = \frac{m_{\text{я}} \cdot r_{\text{я}}^2}{2 \cdot R_{\text{вала}}} \cdot Z = \frac{0.04 \cdot (0.01)^2 \cdot 4.29}{2 \cdot 0.001} = 0.0086; \quad (2.13)$$

$$\dot{x}(t) + \frac{M_{\text{н}}(t)}{R_{\text{к}} \cdot m_{\text{нл}}} = \frac{K}{m_{\text{нл}}} \cdot \dot{\omega}(t). \quad (2.14)$$

Знаходимо передатну функцію об'єкта управління:

$$m_{\text{нл}} \ddot{x}(t) = K \cdot \dot{\omega}(t); \quad (2.15)$$

$$m_{\text{нл}} \cdot V(s) \cdot s = K \cdot \omega(s) \cdot s; \quad (2.16)$$

$$W_{\text{oy}}^u(s) = \frac{V(s)}{\omega(s)} = \frac{K}{m_{\text{нл}}} = \frac{0.0086}{0.7} = 0.012. \quad (2.17)$$

А також передавальну функцію двигуна з обурення:

$$m_{\text{нл}} \ddot{x}(t) = -\frac{M_{\text{н}}(t)}{R_{\text{к}}}; \quad (2.18)$$

$$m_{\text{нл}} \cdot V(s) \cdot s = -\frac{M_{\text{н}}(s)}{R_{\text{к}}}; \quad (2.19)$$

$$W_{\text{oy}}^f(s) = \frac{V(s)}{M(s)} = -\frac{1}{R_{\text{к}} \cdot m_{\text{нл}} \cdot s} = -\frac{43.29}{s}; \quad (2.20)$$

$$W_{\partial \theta}^f(s) = \frac{\omega(s)}{M(s)} = \frac{W_{oy}^f(s)}{W_{oy}^u(s)} = \frac{-43.29}{s \cdot 0.012} = \frac{-3607}{s}. \quad (2.21)$$

Для управління системою вибираємо інтегральний закон управління, оскільки вплив, що обурює, має у складі інтегруючу ланку [6, 7].

Враховуючи наявність у об'єкту достатньої маси, можна сказати, що вона негативно впливатиме на постійну часу двигуна. Час розгону двигуна з блоком редуктора та колесом становить - $t_{пнк} = 1,2$ з. Звідси постійна часу двигуна з блоком редуктора та колесом, але без урахування маси об'єкта, дорівнює:

$$T_{ем1} = \frac{t_{пнк}}{3..4} = 0,36; \quad (2.22)$$

Виходячи з формули знаходження електромеханічного постійного часу двигуна постійного струму:

$$T = I \cdot R \cdot C_e \cdot C_m \cdot \Phi^2; \quad (2.23)$$

Можна сказати, що для одного і того ж двигуна частина буде постійною, щоб знайти вплив навантаження, достатньо врахувати його момент інерції:

$$T_{ем1} = I \cdot R \cdot C_e \cdot C_m \cdot \Phi^2; \quad (2.24)$$

$$T_{ем2} = I \cdot R \cdot C_e \cdot C_m \cdot \Phi^2; \quad (2.25)$$

$$\frac{T_{ем1}}{I_1} = \frac{T_{ем2}}{I_2} \quad (2.26)$$

$$I_2 = I_1 + I_H; \quad (2.27)$$

$$T_{эм2} = \frac{T_{эм1} \cdot (I_1 + I_H)}{I_1}; \quad (2.28)$$

$$T_{эм2} = \frac{T_{эм1} \cdot \left(\frac{r_{я}^2 \cdot m_{я}}{2} + \frac{(R_{к1}^2 + R_{к2}^2) \cdot m_{к}}{2} + m_{пл} \cdot i^2 \right)}{\left(\frac{r_{я}^2 \cdot m_{я}}{2} + \frac{(R_{к1}^2 + R_{к2}^2) \cdot m_{к}}{2} \right)}; \quad (2.29)$$

$$T_{эм2} = \frac{0.36 \cdot \left(\frac{0.01^2 \cdot 0.04}{2} + \frac{(0.033^2 + 0.022^2) \cdot 0.05}{2} + 0.7 \cdot 0.012^2 \right)}{\left(\frac{0.01^2 \cdot 0.04}{2} + \frac{(0.033^2 + 0.022^2) \cdot 0.05}{2} \right)} = 0.58. \quad (2.30)$$

2.3 Розробка функціональної та структурної схем

У системі управління рухомою платформою об'єктом управління є сама платформа. Основним датчиком системи є датчик потенціометра, який вимірює положення платформи. Для розширення функціональності системи передбачено можливість підключення інших датчиків до мікропроцесорного блоку.

Основним компонентом мікропроцесорного блоку є мікроконтролер серії ESP32. Він містить вбудований 8-канальний мультиплексор і 10-розрядний АЦП. Система реалізує автоматичний режим роботи згідно з заданим алгоритмом управління, який завантажується в мікроконтролер перед початком роботи.

Для посилення потужності та перетворення сигналів необхідні блоки підсилення потужності та елементи перетворення. Виконавчими механізмами в системі є 4 двигуни постійного струму (приводи) на двох умовних осях.

Замкнений цикл управління платформою по вертикальній осі аналогічний для інших осей. Структурна схема системи управління показана на рис. 2.2.

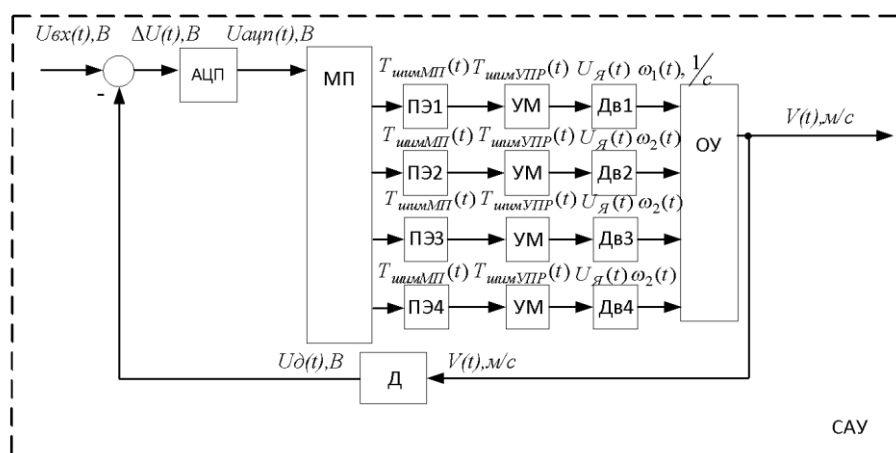


Рисунок 2.2 – Функціональна схема системи автоматичного керування

Розробимо систему автоматичного управління рухом однієї осі роботизованої платформи без використання цифрового регулятора. Система

буде працювати за принципом порівняння двох сигналів. Якщо сигнали відрізняються, платформа переміщається в сторону, яка зближує їх.

Система управління рухом однієї осі роботизованої платформи буде працювати так:

1. Датчик вимірює положення платформи.
2. Мікропроцесор обробляє сигнал з датчика і генерує сигнал управління.
3. Виконавчі механізми рухають платформу в напрямку, вказаному сигналом управління.

Для інших осей платформи система управління буде аналогічною. Структурна схема системи управління контуром однієї поздовжньої осі руху платформи представлена на рис. 2.3.

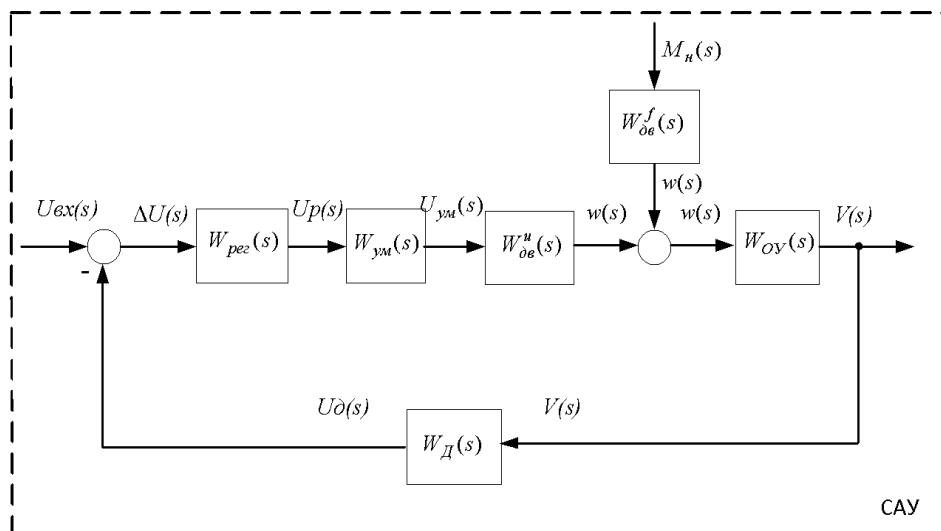


Рисунок 2.3 - Структурна схема системи автоматичного управління за контуром однієї поздовжньої осі руху платформи

Елементи структурної схеми:

$W_{\text{рег}}$ – регулятор;

$W_{\text{УМ}}$ - підсилювач потужності;

$W_{\text{ДВ}}$ – двигун постійного струму (виконавчий механізм, привід);

$W_{\text{Д}}$ – потенціометричний датчик;

$W_{\text{ОУ}}$ – об'єкт управління (рухома платформа).

Коефіцієнт передачі датчика:

$$k_{\text{Д}} = 2,5. \quad (2.31)$$

Передатна функція регулятора:

$$W_{pez}(s) = \frac{\Delta U(s)}{U_p(s)} = \frac{k_{pez}}{s} = \frac{0.835}{s}. \quad (2.32)$$

Коефіцієнт передачі підсилювача потужності

$$k_{ym} = 1.8. \quad (2.33)$$

2.4 Статичний розрахунок

Відповідно до передавальної функції кожного блоку системи впливає на передатну функцію розімкнутої системи:

$$\begin{aligned} W^u(s) &= \frac{V(s)}{U_{6x}(s)} = W_{pez}(s) \cdot W_{ym}(s) \cdot W_{\partial 6}^u(s) \cdot W_{oy}(s) = \\ &= \frac{k_{pez}}{s} \cdot k_{ym} \cdot \frac{k_{\partial 6}^u}{(T_{\partial 6} \cdot s + 1)} = \frac{0.835}{s} \cdot 1.8 \cdot \frac{8.88}{0.58 \cdot s + 1} \cdot 0.12 = \\ &= \frac{1.6}{s \cdot (0.58 \cdot s + 1)}. \end{aligned} \quad (2.34)$$

Передатна функція розімкнутої системи по впливу, що обурює:

$$\begin{aligned} W^f(s) &= \frac{V(s)}{M_n(s)} = W_{\partial 6}^f(s) \cdot W_{oy}(s) \cdot W_{\partial}(s) = \\ &= \frac{k_{\partial 6}^f}{s} \cdot k_{oy} = \frac{-3607}{s} \cdot 0.12 \cdot 2.5 = \frac{-1082.1}{s}. \end{aligned} \quad (2.35)$$

Передатна функція замкнутої системи за впливом:

$$\Phi^u(s) = \frac{V(s)}{U_{6x}(s)} = \frac{W^u(s)}{1 + W_{\partial}(s) \cdot W^u(s)} = \frac{1.6}{s \cdot (0.58 \cdot s + 1) + 4} \quad (2.36)$$

Передатна функція замкнутої системи за впливом, що обурює:

$$\Phi^f(s) = \frac{V(s)}{M_n(s)} = \frac{W^f(s)}{1 + W_\delta(s) \cdot W^u(s)} = \frac{-1082.1 \cdot (0.58 \cdot s + 1)}{s \cdot (0.58 \cdot s + 1) + 4} \quad (2.37)$$

Передавальна функція помилки замкнутої системи за задавальним впливом:

$$\Phi_u^\varepsilon(s) = \frac{\Delta U(s)}{U_{ex}(s)} = \frac{1}{1 + W^u(s)} = \frac{s \cdot (0.58 \cdot s + 1)}{s \cdot (0.58 \cdot s + 1) + 1.6} \quad (2.38)$$

Щоб забезпечити задану точність управління, необхідно розрахувати коефіцієнт передачі підсилювача потужності. Для цього можна використовувати вихідну теорему про кінцеві значення помилок.

На вхід компаратора перетворювача елемента перетворювача надходить задана напруга - В. На вихід системи виходить швидкість переміщення платформи однією умовною осі. Оскільки в системі є інтегратор, то статична помилка дорівнює нулю, але є швидкісна помилка.

$$\begin{aligned} W_n^u(s) &= \frac{V(s)}{U_3(s)} = W_{pez}(s) \cdot W_{ym}(s) \cdot W_{\delta\theta}^u(s) \cdot W_{oy}(s) \cdot W_\delta(s) = & (2.39) \\ &= \frac{k_{pez}}{s} \cdot k_{ym} \cdot \frac{k_{\delta\theta}^u}{(T_{\text{эм}} \cdot s + 1)} \cdot k_{oy} = \frac{0.835}{s} \cdot 1.8 \cdot \frac{8.88}{0.58 \cdot s + 1} \cdot 0.12 \cdot 2.5 = \\ &= \frac{4}{s \cdot (0.58 \cdot s + 1)}; \end{aligned}$$

$$\varepsilon_{ск} = \lim_{s \rightarrow 0} s \cdot \Phi_\varepsilon^u(s) \cdot \frac{V}{s^2} = \lim_{s \rightarrow 0} s \cdot \frac{s \cdot (0.58 \cdot s + 1)}{s \cdot (0.58 \cdot s + 1) + K_{pc}} \cdot \frac{V}{s^2} = \frac{V}{K_{pc}}; \quad (2.40)$$

$$V = \frac{U_3}{t_{mn}} = \frac{5}{6} = 0.83; \quad (2.41)$$

$$\varepsilon_{ск} = 0.05 \cdot U_3 = 0.25; \quad (2.42)$$

$$\varepsilon_{ск} \leq \frac{V}{K_{pc}} \Rightarrow K_{pc} \geq \frac{V}{\varepsilon_{ск}} \geq \frac{0.83}{0.25} \geq 3.32; \quad (2.43)$$

Виходячи з якості перехідного процесу, коефіцієнт розімкнутої системи був обраний $K_{pc} = 40$ - тоді новий коефіцієнт коригувального пристрою буде дорівнювати $K_{ку} = 10$.

2.5 Можливість зміни швидкості робота

Досліджуємо можливість зміни швидкості робота не за завдання початкових умов, а прямо на ходу під час руху робота.

Для сигналізації про те, що відбулася зміна кутової швидкості, і коректного переходу на нову траєкторію потрібно знати не лише змінені значення кутових швидкостей, а й значення кутових швидкостей, які мали місце до зміни. Кутові швидкості до зміни ми позначимо як $U1$ для лівого колеса та $U2$ для правого. Кутові швидкості після зміни позначимо як $U3$ для лівого колеса та $U4$ для правого.

Будемо надавати змінним $U1$ і $U2$ значення поточних швидкостей коліс, потім змінювати швидкості коліс і записувати швидкості в змінні $U3$ і $U4$. Таким чином, ми зможемо контролювати момент зміни кутових швидкостей, а отже, зафіксувати і швидкість до зміни, і швидкість після зміни.

Визначити, що відбулася зміна кутової швидкості хоча б одного колеса можна за значенням $|U1 - U3|$ та $|U2 - U4|$. Якщо хоч один модуль більше 0, то кутова швидкість хоча б одного колеса змінилася.

При цьому можливі 4 варіанти зміни траєкторії: перехід з одного кола на інше, перехід з кола на пряме, з прямим на коло, з прямим на пряме.

1) Розглянемо перехід з одного кола на інше. Цьому переходу відповідають умови $|U4 - U3| > 0$, $|U2 - U1| > 0$. Завдання полягає в тому, щоб визначити координати центру нового кола за становищем робота на старому колі та за напрямом його руху по старому колу.

При цьому можливі такі випадки:

1.1 Нехай $0 \leq \beta \leq \pi/2$.

$$x_c = x \pm R \cdot \cos[\omega \cdot t] \cdot \beta$$

$$y_c = y \pm R \cdot \sin[\omega \cdot t] \cdot \beta$$

При $\omega < 0$:

$$x_c = x \mp R \cdot \cos \beta$$

$$y_c = y \mp R \cdot \sin \beta$$

Тут і в наступних варіантах розташування робота верхній знак відповідає випадку, коли кутова швидкість лівого колеса перевищує кутову швидкість правого колеса.

1.2 Нехай $\pi/2 \leq \beta \leq \pi$.

$$x_c = x \mp R \cdot \cos(\pi - \beta) = x \pm R \cdot \cos \beta$$

$$y_c = y \pm R \cdot \sin(\pi - \beta) = y \pm R \cdot \sin \beta$$

При $\omega < 0$

$$x_c = x \pm R \cdot \cos(\pi - \beta) = x \mp R \cdot \cos \beta$$

$$y_c = y \mp R \cdot \sin(\pi - \beta) = y \mp R \cdot \sin \beta$$

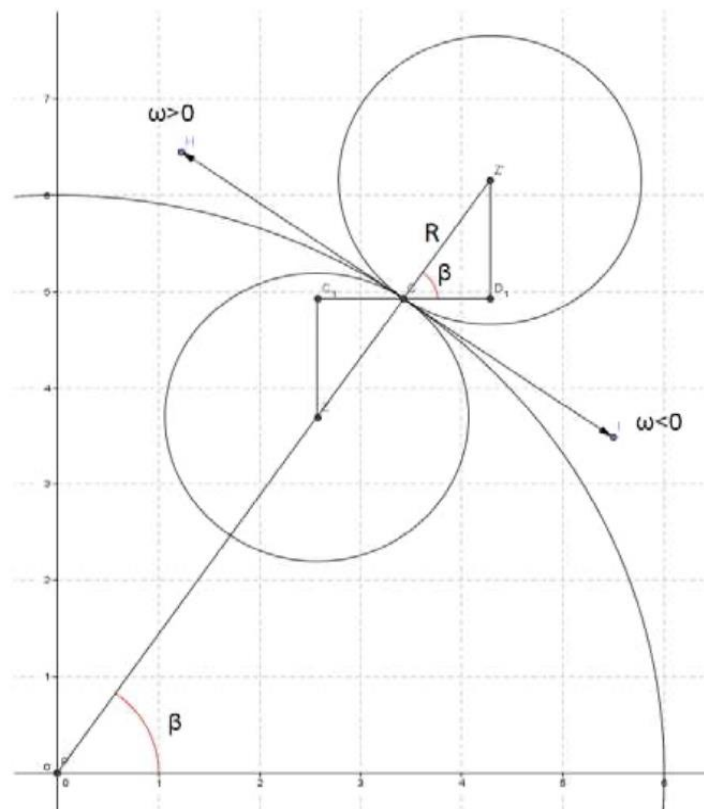


Рисунок 2.4 - Перехід на нове коло при $0 \leq \beta \leq \pi/2$

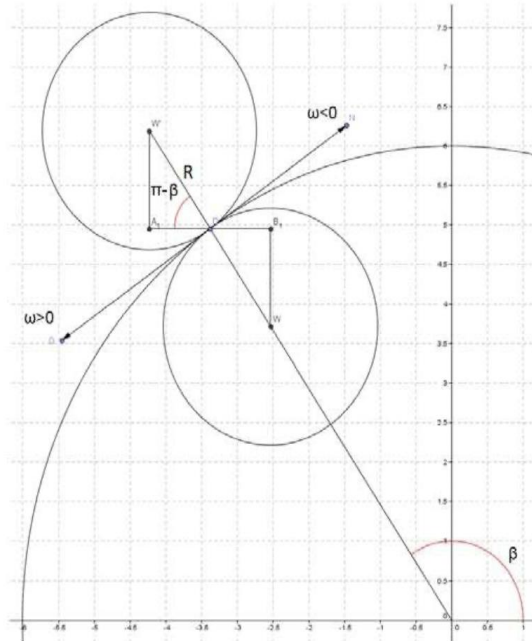


Рисунок 2.5 - Перехід на нове коло при $\pi/2 \leq \beta \leq \pi$

1.3 Нехай $-\pi/2 \leq \beta \leq 0$.

$$x_c = x \pm R \cdot \cos(\pi - \beta) = x \pm R \cdot \cos \beta$$

$$y_c = y \pm R \cdot \sin(\pi - \beta) = y \mp R \cdot \sin \beta$$

При $\omega < 0$:

$$x_c = x \mp R \cdot \cos(\pi - \beta) = x \mp R \cdot \cos \beta$$

$$y_c = y \mp R \cdot \sin(\pi - \beta) = y \pm R \cdot \sin \beta$$

1.4 Нехай $-\pi < \beta \leq -\pi/2$.

$$x_c = x \mp R \cdot \cos(\pi + \beta) = x \pm R \cdot \cos \beta$$

$$y_c = y \mp R \cdot \sin(\pi + \beta) = y \pm R \cdot \sin \beta$$

При $\omega < 0$:

$$x_c = x \pm R \cdot \cos(\pi + \beta) = x \mp R \cdot \cos \beta$$

$$y_c = y \pm R \cdot \sin(\pi + \beta) = y \mp R \cdot \sin \beta$$

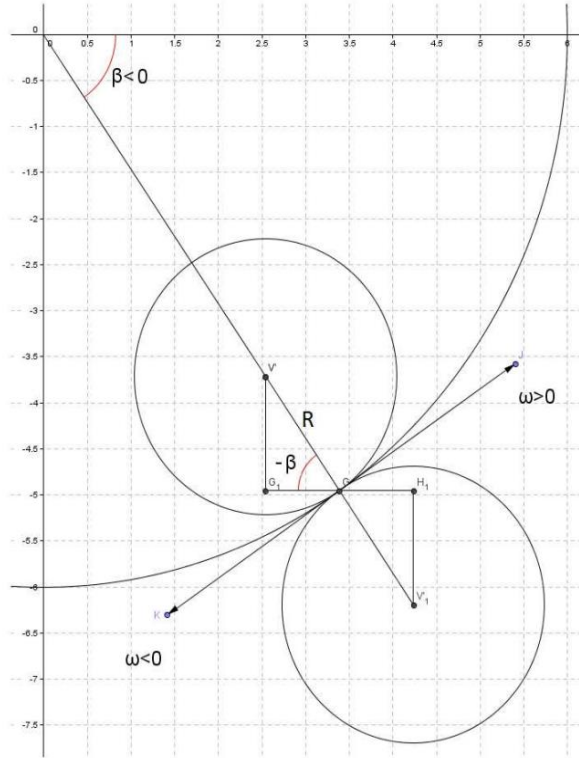


Рисунок 2.6 – Перехід на нове коло при $-\pi/2 \leq \beta \leq 0$

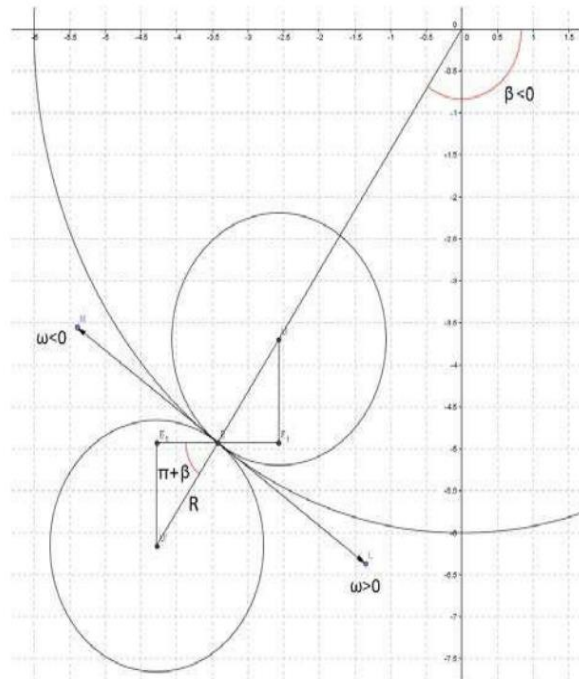


Рисунок 2.7 – Перехід на нове коло при $-\pi < \beta \leq -\pi/2$

Усі отримані результати можна поєднати такими виразами:

$$x_c = x - \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot R \cdot \cos[\dots] \beta$$

$$y_c = y - \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot R \cdot \sin(\varphi_0) \cdot \beta$$

Визначивши координати центру нового кола, можна знайти початкове положення робота на новому колі. В силу періодичності функцій косинуса і синуса, замість кута β можна використовувати кут $\varphi_0 + \omega t$, який відрізняється від β цілим числом періодів. Це дозволить не запам'ятовувати додатково координати центру кола, з якого відбувається перехід на нове коло.

2. Проаналізуємо перехід із кола на пряму. На тій підставі, що рух прямої однозначно визначається орієнтацією робота, необхідно прийняти в якості нового значення φ_0 орієнтацію в той момент, коли відбулася зміна кутової швидкості, а в якості нових початкових координат x_0 і y_0 координати робота в цей же момент.

3. Розглянемо випадок переходу з прямої на коло. У цьому випадку виконується умови $|U_4 - U_3| > 0$ та $U_1 = U_2$. Де координати центру кола знаходяться так само, як і за заданих початкових умов. Але як кут φ_0 використовується не початкова орієнтація робота, яке орієнтація в той момент, коли відбулася зміна кутової швидкості, а як координат x_0 і y_0 координати робота в цей же момент.

4. При переході з прямою на пряму, як і при переході з кола на пряму, для завдання нової траєкторії достатньо лише зареєструвати орієнтацію та координати у момент зміни кутової швидкості.

$$x_c = x - \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot R \cdot \cos(\varphi_0) \cdot \beta$$

Таким чином, розглянуто випадки невід'ємних кутових швидкостей коліс. Проробивши ті ж міркування для негативних кутових швидкостей коліс, отримаємо наступні результати. Для координат центру кола при заданих початкових умовах та координат центру кола при переході з прямої на коло отримаємо:

$$x_c = x_0 + \text{sign}(\omega) \cdot R \cdot \sin \varphi_0$$

$$y_c = y_0 + \text{sign}(\omega) \cdot R \cdot \cos \varphi_0$$

Рівняння руху по колу залишаються без змін, оскільки при позитивному значенні відбувається рух робота проти годинникової стрілки, а при негативному значенні рух за годинниковою стрілкою. Рівняння руху по прямій також залишаються без змін, тому що при зміні знака кутових швидкостей відбудеться

зміна напрямку руху на протилежне, а це буде враховано наявністю в рівняннях кутової швидкості коліс.

Для расчета ориентации получим следующие соотношения.

Для 1 и 4 четвертей ($-\pi/2 < \beta \leq \pi/2$):

$$\varphi = \pi + \text{sign}(\omega) \cdot \frac{\pi}{2} - \beta$$

Для 2 четверті ($\beta > \pi/2$):

$$\varphi = 2\pi - \text{sign}(\omega) \cdot \frac{\pi}{2} - \beta$$

Для 3 четверті ($\beta \leq -\pi/2$):

$$\varphi = -\text{sign}(\omega) \cdot \frac{\pi}{2} - \beta$$

При переході з кола на коло з позитивних кутових швидкостей на позитивні та з негативних на негативні маємо:

$$x_c = x - \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot R \cdot \cos[\omega] \beta$$

$$y_c = y - \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot R \cdot \sin[\omega] \beta$$

При переході з негативних кутових швидкостей на позитивні та з позитивних на негативні отримаємо:

$$x_c = x + \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot R \cdot \cos[\omega] \beta$$

$$y_c = y + \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot R \cdot \sin[\omega] \beta$$

Тепер розглянемо рух при різних за знаком кутових швидкостях коліс. Нехай орієнтація робота дорівнює 90° . Можливі 4 варіанти співвідношення кутових швидкостей:

$$1) \omega_L > 0, \omega_R < 0, |\omega_L| > |\omega_R|$$

В цьому випадку $\omega < 0, U > 0$. Положення центру кола F1 показано рис. 2.8.

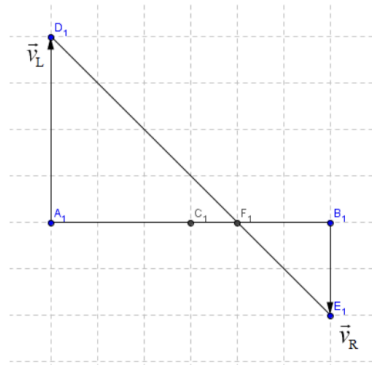


Рисунок 2.8 – Положення центру кола F1 при $\omega_L > 0$, $\omega_R < 0$, $|\omega_L| > |\omega_R|$

2) $\omega_L > 0$, $\omega_R < 0$, $|\omega_L| < |\omega_R|$

У цьому випадку $\omega < 0$, $U < 0$. Положення центру кола F2 показано рис. 2.9.

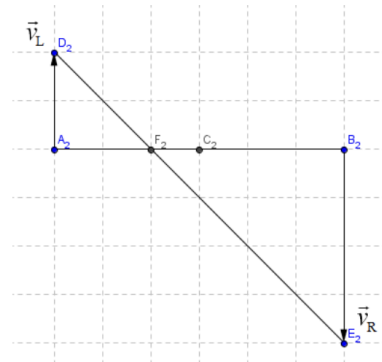


Рисунок 2.9 – Положення центру кола F2 при $\omega_L > 0$, $\omega_R < 0$, $|\omega_L| < |\omega_R|$

3) $\omega_L < 0$, $\omega_R > 0$, $|\omega_L| > |\omega_R|$

У цьому випадку $\omega > 0$, $U < 0$. Положення центру кола F3 показано рис. 2.10.

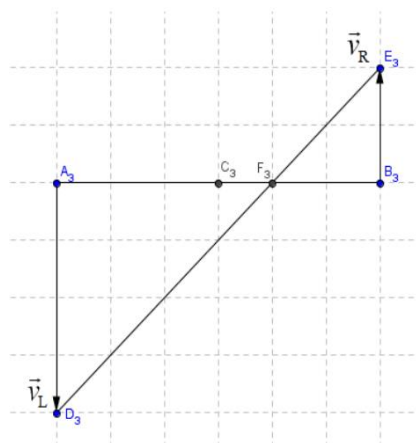


Рисунок 2.10 – Положення центру кола F3 при $\omega_L < 0$, $\omega_R > 0$, $|\omega_L| > |\omega_R|$

4) $\omega_L < 0$, $\omega_R > 0$, $|\omega_L| < |\omega_R|$

У цьому випадку $\omega > 0$, $U > 0$. Положення центру кола F4 показано рис. 2.11.

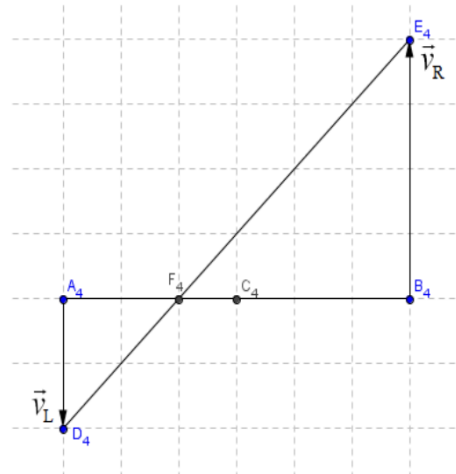


Рисунок 2.11 – Положення центру кола F4 при $\omega L < 0$, $\omega R > 0$, $|\omega L| < |\omega R|$

При однакових знаках U та ω отримуємо координати центру кола за заданих початкових умов:

$$x_c = x_0 - R \cdot \sin\varphi_0$$

$$y_c = y_0 - R \cdot \cos\varphi_0$$

При різних знаках U та ω :

$$x_c = x_0 + R \cdot \sin\varphi_0$$

$$y_c = y_0 + R \cdot \cos\varphi_0$$

Рівняння руху по колу залишаються без змін, оскільки при позитивному значенні відбувається рух робота проти годинникової стрілки, а при негативному - рух за годинниковою стрілкою.

При позитивному U у всіх чвертях орієнтація обчислюється як і, як і разі позитивних кутових швидкостей коліс, оскільки рух центру робота відбувається у бік його орієнтира. При негативному U рух роботи відбувається в бік, протилежну його орієнтації, тому орієнтація обчислюється як у разі негативних кутових швидкостей коліс.

При переході з кола на коло на положення центру нового кола впливають відразу 4 фактори: знак кутової швидкості руху робота до зміни кутової швидкості коліс, знак лінійної швидкості робота до зміни швидкості коліс, знак кутової швидкості руху робота після зміни кутової швидкості коліс і лінійної швидкості робота після зміни кутової швидкості коліс. При зміні будь-якого з цих

параметрів координати центру змінюються діаметрально протилежні щодо положення робота в момент зміни швидкості. Тому, з усього вищесказаного, отримаємо такі загальні висловлювання, правильні будь-яких знаків і величин кутових швидкостей коліс.

Для координат центру кола за заданих початкових умов:

$$x_c = x_0 - \text{sign}(\omega) \cdot \text{sign}(\omega_L + \omega_R) \cdot R \cdot \sin\varphi_0$$

$$y_c = y_0 - \text{sign}(\omega) \cdot \text{sign}(\omega_L + \omega_R) \cdot R \cdot \cos\varphi_0$$

Для розрахунку орієнтації залежно від положення на колі:

Для 1 та 4 чвертей ($-\pi/2 < \beta \leq \pi/2$):

$$\varphi = \pi - \text{sign}(\omega) \cdot \text{sign}(\omega_L + \omega_R) \cdot \frac{\pi}{2} - \beta$$

Для 2 чверті ($\beta > \pi/2$):

$$\varphi = 2\pi + \text{sign}(\omega) \cdot \text{sign}(\omega_L + \omega_R) \cdot \frac{\pi}{2} - \beta$$

Для 3 чверті ($\beta \leq -\pi/2$):

$$\varphi = \text{sign}(\omega) \cdot \text{sign}(\omega_L + \omega_R) \cdot \frac{\pi}{2} - \beta$$

Для координат центру кола при переході з одного кола на інше:

$$x_c = x - \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot \text{sign}(U_1 + U_2) \cdot \text{sign}(U_3 + U_4) \cdot R \cdot \cos\beta$$

$$y_c = y - \text{sign}(U_2 - U_1) \cdot \text{sign}(U_4 - U_3) \cdot \text{sign}(U_1 + U_2) \cdot \text{sign}(U_3 + U_4) \cdot R \cdot \sin\beta$$

При переході з прямої або обертання на місці на коло координати центру знаходимо так само, як і при обчисленні цих координат за заданих початкових умов. При переході на пряму або обертання на місці запам'ятовуємо тільки орієнтацію і положення в момент зміни траєкторії.

Розберемося з описом обертання на місці. Як бачимо, координати центру робота не змінюються, але змінюється орієнтація робота. На підставі того, що руху центру по колу при цьому немає, зв'язати орієнтацію робота з функцією atan2 вже не вийде. Тому будемо обчислювати кут повороту, і якщо він став меншим за нуль, додавати до нього 2π , обнулюючи час при кожному такому

переході через 0 і запам'ятовувати збільшений на 2π кут повороту після його переходу через 0. Тобто повертати його в потрібну ділянку при кожному переході через 0. Аналогічно надходитимемо і при перевищенні куту значення 2π , тільки в цьому випадку будемо зменшувати кут повороту на 2π , а не збільшувати. Ще потрібно врахувати, що обертання на місці за годинниковою стрілкою відповідає негативному значенню і збільшення кута орієнтації, а обертання проти годинникової стрілки відповідає позитивному значенню і зменшення кута орієнтації. Тому при обчисленні кута орієнтації потрібно не додавати t , а віднімати.

При обертанні на місці модуль різниці швидкостей коліс відмінний від нуля. Тому колишня умова включає і випадок обертання на місці, і щоб його виключити, потрібно змінити умову на відміну від нуля різниці модулів модулів. Зміниться і умова переходу на коло з обертання на місці або з прямою. Якщо раніше воно вимагало рівності швидкостей U_1 та U_2 , то тепер потрібно вимагати рівності модулів цих швидкостей. Таким чином, розглянуто вирази, що дозволяють описати рух робота при будь-якій зміні його кутових швидкостей у будь-якій момент часу. Для наочності зобразимо не лише траєкторію руху центру робота, а й траєкторію руху його коліс. Становище коліс можна однозначно отримати з положення центру робота та орієнтації робота, оскільки пряма, по якій розташовані колеса, перпендикулярна прямій, по якій орієнтований робот, та колеса знаходяться на відстані $0.5 \cdot D$ від центру робота. Звідси за допомогою рис. 2.12 отримаємо.

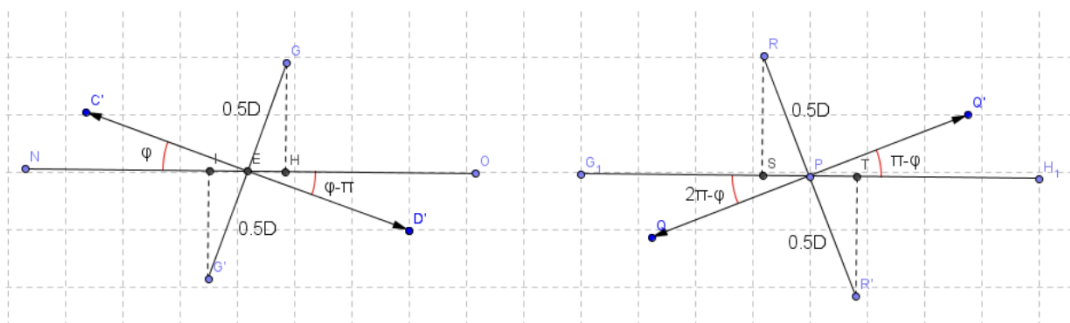


Рисунок 2.12 – Знаходження координат коліс за різної орієнтації робота

За будь-якого положення робота координати лівого колеса x_L і y_L визначаються наступним чином:

$$x_L = x - \frac{D}{2} \cdot \sin \varphi$$

$$y_L = y - \frac{D}{2} \cdot \cos \varphi$$

Координати правого колеса x_R та y_R

$$x_R = x + \frac{D}{2} \cdot \sin \varphi$$

$$y_R = y + \frac{D}{2} \cdot \cos \varphi$$

2.6 Висновки

Створено та реалізовано алгоритм побудови траєкторії та розрахунку орієнтації робота за заданими початковими умовами.

Вивчено можливість зміни траєкторії безпосередньо під час руху робота, розглянувши 4 варіанти зміни траєкторії: перехід з одного кола на інше, перехід з кола на пряме, з прямого на коло, з прямою на пряму.

3 КОНСТРУКТОРСЬКА ЧАСТИНА

3.1 Підбір комплектуючих

Зовнішній вигляд підібраних конструктивних компонентів та приладів показаний на рис. 3.1 – 3.10.

1. Рама:

Модуль рами: Arduino Car Kit Acrylic

Характеристики:

- Вага - 500 грамів
- Розміри 30 см x 15 см



Рисунок 3.1 - Рама мобільної платформи

2. Мотори та колеса:

Мотор-редуктор з колесом

Характеристики:

- Діаметр зовнішній - 65 мм.
- Напруга живлення: 6-8 В.
- Швидкість обертання без навантаження: 170 об/хв (за напруги 3,6 В).



Рисунок 3.2 - Мотор-редуктор та колесо

3. Драйвер моторів:

Модуль L298N

Характеристики:

- Напруга живлення драйвера, В 5 ... 36
- Робочий струм драйвера, А 2
- Максимальне споживання енергії, Вт 25

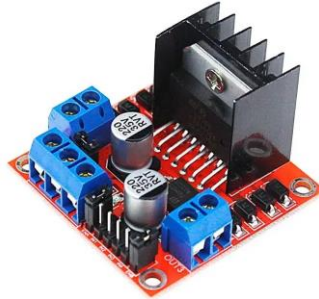


Рисунок 3.3 - Модуль L298N

4. Батарея

Акумулятор GNB 3000mAh 2S 7.4V 5C LiPo XT30

Характеристики:

- Ємність: 3000mAh
- Кількість осередків: 2S
- Напруга: 7.4V



Рисунок 3.4 - Акумулятор GNB 3000mAh 2S 7.4V 5C LiPo XT30

5. Мікроконтролер ESP-32

Характеристики:

- Підтримувані Wi-Fi та Bluetooth стандарти classic та BLE (v4.2)
- Частота мікроконтролера 240 МГц
- Напруга живлення 3.7-12 В
- Розміри плати 55 x 28 мм



Рисунок 3.5 - ESP-32

6. Передавач NRF 2401L Plus

Характеристики:

- Робоча частота: 2.4 ГГц
- Має можливість передачі даних на відстань до кількох сотень метрів у відкритому просторі (залежить від умов та оточення).
- Є можливість налаштування каналу, адреси та інших параметрів комунікації.



Рисунок 3.6 - Передавач NRF 2401L Plus

7. Пульт RadioMaster TX16S МКII 4in1 M2

Характеристики:

- Частота передачі: 2,400 ГГц-2,480 ГГц
- Модуль передавача: Внутрішній багатопроTOCOLЬНИЙ модуль 4-в-1 (CC2500 CYRF6936 A7105 NRF2401)
- Посилення антени: 2 дБ (потужність передачі регулюється)
- Відстань дистанційного керування:> 2 км при 22 дБм



Рис. 3.7 - Пульти RadioMaster TX16S MKII 4in1 M2

8. GPS модуль

GPS модуль NEO-6M NEO6MV2

Характеристики:

- SuperSense[®] Indoor GPS: -162 dBm чутливості відстеження.
- Підтримка SBAS (WAAS, EGNOS, MSAS, GAGAN).
- Живлення: 3...5V.



Рисунок 3.8 - GPS модуль NEO-6M NEO6MV2

9. Камера

Камера для FPV Caddx Ant Nano 1200TVL 1/3 CMOS 4:3/16:9 PAL/NTSC

Характеристики:

- Матриця: 1/3" CMOS
- Горизонтальна роздільна здатність: 1200TVL
- Телевізійна система: NTSC/PAL

- Формат зображення: 16:9 або 4:3



Рисунок 3.9 - Камера для FPV Caddx Ant Nano

10. Відеопередавач АКК FX2-ultimate-mini 25/200/600/1200mW 5.8G на 37каналів

Характеристики:

- Частота: 5.8GHz
- Вихідна потужність: 25/200/600/1200mW
- Вхідна напруга: 7-26V (2-6S LiPo)



Рисунок 3.10 Відеопередавач АКК FX2

3.2 Основні характеристики мобільної платформи

Розміри: Розміри мобільної платформи включають в себе довжину, ширину і висоту корпусу, а також розмір коліс та моторів. Геометричні параметри важливі для оцінки маневреності та проникнення в обмежені простори.

Маса: Маса мобільної платформи визначає його навантаження на мотори і акумулятори, що впливає на тривалість руху та максимальну швидкість.

Кількість моторів і пропелерів: Мобільні платформи зазвичай мають чотири мотори та пропелери, але можуть мати іншу кількість. Це важливо для стабільності та керованості.

Тяга моторів: Спроможність моторів створювати тягу важлива для підтримання швидкості та маневриності.

Тип акумулятора: Використовуваний тип акумулятора (наприклад, літій-полімерний або літій-іонний) впливає на ємність і тривалість руху.

Ємність акумулятора: Ємність акумулятора в ампер-годинах (Ah) визначає, як довго мобільна платформа може перебувати в русі перед зарядкою.

Тип контролера: Вид контролера (наприклад, ПД-регулятор) визначає ефективність та точність керування мобільною платформою.

Система стабілізації: Наявність системи стабілізації, такої як гіроскопи і акселерометри, впливає на стабільність руху.

3.3 Аналіз впливу параметрів на рух по заданій траєкторії

Маневреність

Геометричні параметри: Розміри та форма мобільної платформи можуть впливати на його здатність проникнути в малогабаритні місця та виконувати маневри в обмеженому просторі.

Маса та тип акумулятора: Маса мобільної платформи та тип використовуваного акумулятора визначають тривалість руху.

Система стабілізації та контролер: Наявність ефективної системи стабілізації та точного контролера допомагає забезпечити стабільну платформу.

3.4 Розробка рухомої платформи.

Вибір контролера для цієї системи ґрунтувався на трьох умовах: швидкодія, вартість, точність.

1. Однополярне живлення – +5 В.
2. Габаритні розміри – не більше 65x35 мм.
3. Наявність портів/інтерфейсів для сполучення з датчиками (джойстик) та виконавчими пристроями (Bluetooth, NRF24L01+).
4. Наявність засобів розробки ПЗ та бібліотек.

Ціна має на увазі задоволення всіх критеріїв за мінімальну вартість.

У результаті пошуків оптимального мікроконтролеру, що задовольняє усі поставлені критерії був отриманий ESP32.

Побудований на новому мініатюрному високопродуктивному модулі ESP-WROOM-32 від популярного бренду, призначеного для широкого спектру застосувань, починаючи від мікропотужних мережних датчиків до складних програм, наприклад, таких як кодування голосу, потокова передача музики та MP3 кодування.

На модулі зібрано всю необхідну мінімальну периферію, достатню для швидкого та комфортного старту роботи з ESP-WROOM-32. Модуль розробника виконаний на базі популярного двоядерного чіпсету ESP32, з тактовою частотою від 80 МГц до 240 МГц, що змінюється, можливістю індивідуального управління і живлення. Модуль розроблений для переносної та автономної електроніки та додатків інтернету речей, виконаний у мініатюрному корпусі 2,5 см x 1,8 см.

Мікромодуль має багату периферію, що включає різні інтерфейси, роз'єм для SD карти, інфрачервоний порт, інтерфейс для підключення ємнісної сенсорної панелі. Однією з переваг модуля є наднизьке споживання електроенергії та гнучкий вибір режимів, що «сплять», що дозволяють отримати цифри до 20 мкА в режимі deep sleep mode. Модуль підтримує три режими роботи - AP+STA, AP та STA.

Також один з мікроконтролерів, що задовільняє мінімальні потреби для розробки і використання на МКП це Arduino Mega. Arduino Mega 2560 – це пристрій на основі мікроконтролера ATmega2560. До його складу входить все необхідне для зручної роботи з мікроконтролером: 54 цифрові входи/виходи (з яких 15 можуть використовуватися як ШІМ-виходи), 16 аналогових входів, 4 UART (апаратних приймачів для реалізації послідовних інтерфейсів), кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP для внутрішньосхемного програмування та кнопка скидання. Для початку роботи з пристроєм досить просто подати живлення від AC/DC-адаптера або батареї або підключити його до комп'ютера за допомогою USB-кабелю. Arduino Mega сумісний з більшістю плат розширення, розроблених для Arduino Duemilanove та Diecimila.

Mega 2560 – це оновлена версія Arduino Mega.

Ардуіно Mega 2560 відрізняється від усіх попередніх плат тим, що в ньому для перетворення інтерфейсів USB-UART замість мікросхеми FTDI використовується мікроконтролер ATmega16U2 (ATmega8U2 у версіях плати R1 та R2).

На платі Mega 2560 версії R2 доданий резистор, що підтягує до землі лінію HWB мікроконтролера 8U2. Подібна міра дозволяє спростити процес оновлення прошивки та перехід пристрою в режим DFU.

Зміни на платі версії R3 наведені нижче:

Розпинка 1.0: додано висновки SDA та SCL (біля виводу AREF), а також два нових виводи, розташовані біля виводу RESET. Перший – IOREF – дозволяє платам розширення підлаштовуватися під робочу напругу Ардуїно. Цей висновок призначений для сумісності плат розширення як з 5В-Ардуїно на базі мікроконтролерів AVR, так і з 3.3В-платами Arduino Due. Другий висновок ні до чого не приєднаний та зарезервований для майбутніх цілей. Поліпшено завадостійкість ланцюга скидання. Мікроконтролер ATmega16U2 замінено на 8U2.

При детальному порівнянні ESP32 та Arduino Mega видно, що це дві різні платформи для розробки з унікальними характеристиками та призначенням. Давайте розглянемо кілька аспектів, у яких ESP32 може бути привабливим порівняно з Arduino Mega:

Бездротовий функціонал:

ESP32: Одна з основних переваг ESP32 - вбудований модуль Wi-Fi та Bluetooth. Це важливо для проектів, яким потрібна бездротова комунікація, Інтернет речей (IoT), мережеве керування тощо.

Arduino Mega: Arduino Mega не має вбудованих бездротових модулів і потребує додаткових модулів або щитків для покращення функціоналу Wi-Fi або Bluetooth.

Продуктивність та ресурси:

ESP32: ESP32 має потужний процесор і багато оперативної пам'яті. Це особливо корисно для складних проектів, таких як обробка відео, машинне навчання або інші вимогливі завдання.

Arduino Mega: Arduino Mega забезпечує достатню продуктивність для більшості простих проектів, хоча може бути менш потужним в порівнянні з ESP32 при роботі з складними програмами.

Наявність Bluetooth:

ESP32: Як вже зазначено, ESP32 має вбудований модуль Bluetooth. Це зробить його більш зручним для проектів, яким потрібна робота з Bluetooth-пристроями.

Arduino Mega: Arduino Mega не призначений для роботи з Bluetooth і вимагає додаткового обладнання для цього.

Ціновий фактор

Ціновий фактор може бути важливим при виборі між ESP32 та Arduino Mega для проекту з обмеженим бюджетом, оскільки ESP32 зазвичай доступний за меншою ціною ніж Arduino Mega.

У порівнянні з Arduino Mega, ESP32 є компактним і зручним у використанні та розробці мобільних роботів. З іншого боку, Arduino Mega пасує для більших проєктів і має підтримку багатьох введення/виводу (I/O) портами.

Для написання скетчу можна використовувати Arduino IDE, але слід використовувати спеціалізоване ядро, що дозволить транслювати код з Arduino C на зрозумілий для ESP32 мову.

Налаштування Arduino IDE:

У середовищі «Arduino IDE» є «Менеджер плат» який допоможе швидко додати плати, що цікавлять, до списку. Написання скетчу майже нічим не відрізняється від написання скетчу для плат Arduino. З урахуванням того що ESP32 має більшу кількість різним можливостей. І також має багато різних додаткових бібліотек для її використання.

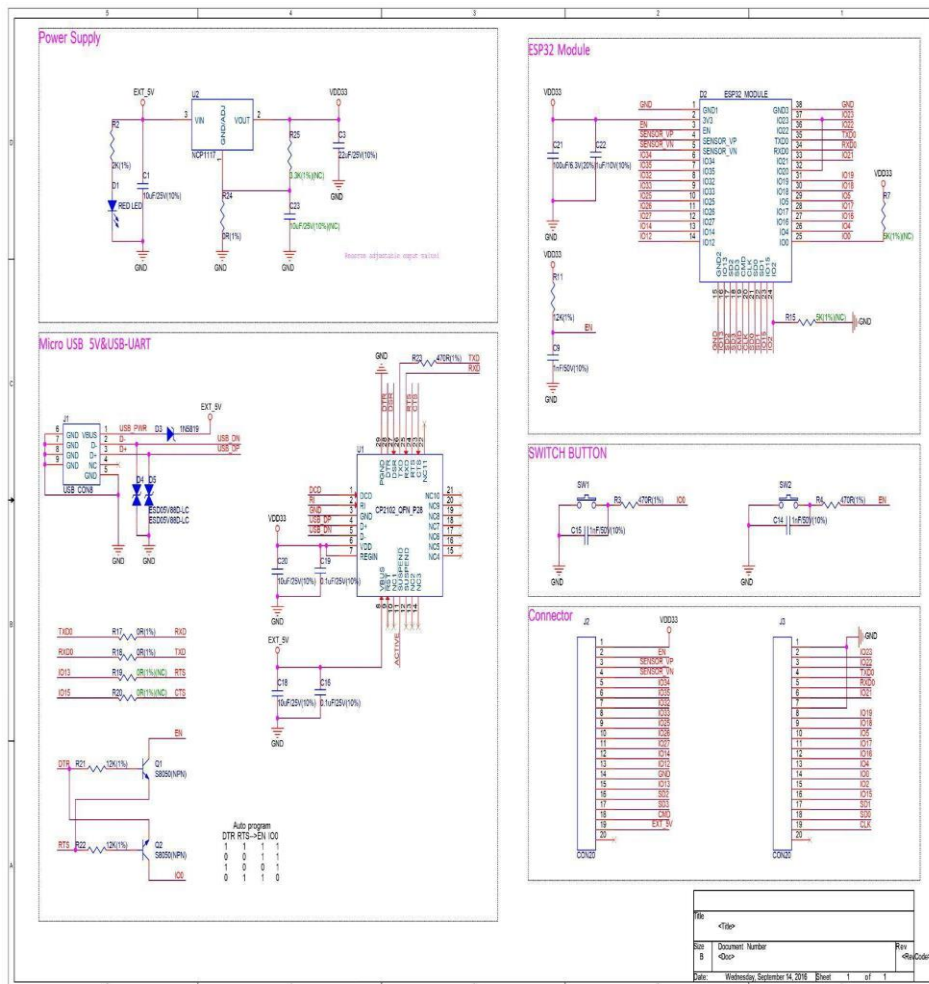


Рисунок 3.11 - Схема електрична принципова ESP32

Для приведення моторів у дію використовується драйвер моторів, через максимальний струм, що можна віддавати мікроконтроллер, через пін. Він дорівнює 40 мАп. При перевищенні цього значення мікроконтроллер перестане

працювати. Тому і слід використовувати драйвер. Мікроконтролер приєднується до пінів керування, та за допомогою напруги і логічних значень віддає команди.

Даний драйвер побудований на основі одноіменного чипу. Який представляє з себе подвійний повний мостовий драйвер.

“Подвійний повний мостовий драйвер керування призначений для прийняття стандартних логічних рівнів TTL (Control Logic) і індуктивних навантажень, таких як: реле, соленоїди, DC і крокові двигуни. Два входи забезпечують увімкнення або вимкнення пристрою незалежно від вхідних сигналів. Випромінювачі нижнього транзистора кожного мосту з'єднані разом і відповідний зовнішній термінал може бути використаний для підключення зовнішнього зондуючого резистора”.

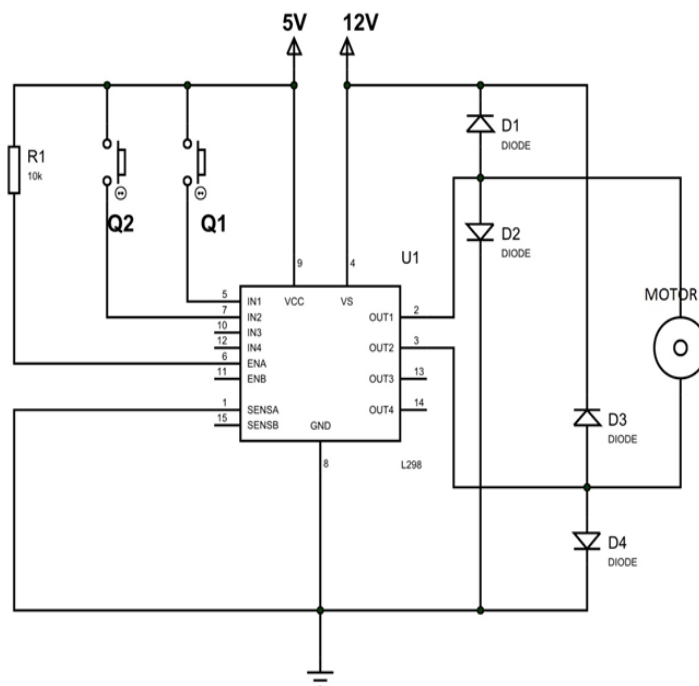


Рисунок 3.12 - Схема драйверу L289N

Використання цього драйверу може бути вкрай різноманітним: вимірювальні прилади, роботи, релейні драйвери, торгові автомати, роботизовані руки, промислові машини, інженерні системи, хобі-проекти та інше.

Для можливості керування платформою на великих дистанціях, необхідно використовувати модуль NRF24L01+. Його принципова електрична схема зображена на рис. 3.13.

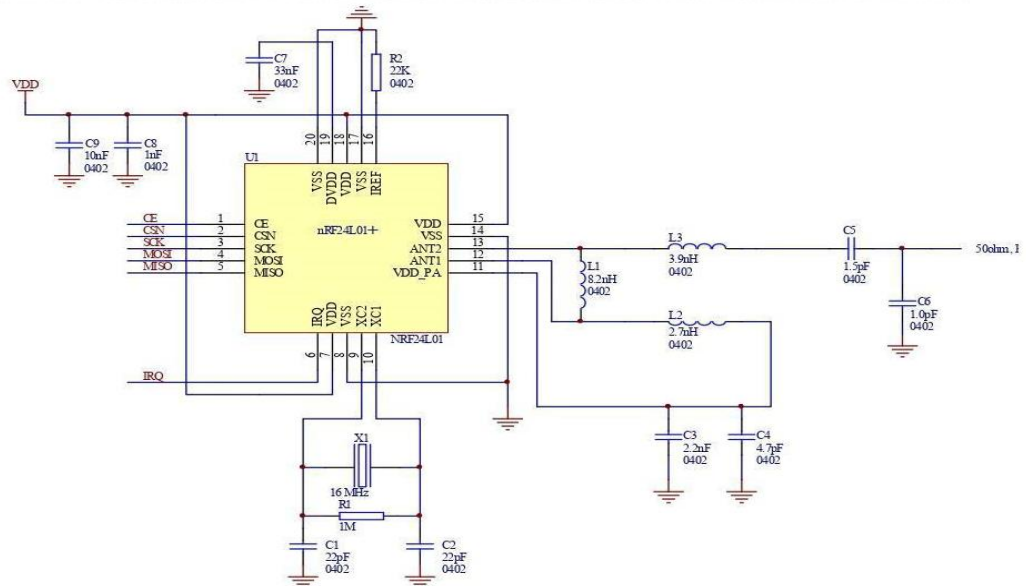


Рисунок 3.13 - Принципова електрична схема NRF24L01 Plus

Модифікація Plus практично не відрізняється від звичайної, винятком являється можливість під'єднання зовнішньої антени, що розрахована частоту 2400 Гц, що значно збільшує дистанцію, прийому сигналу по прямій відкритій місцевості.

Це значення може коливатися через вплив “зони Френеля”, але дистанція, на якій розповсюджується сигнал, а саме до 2 км, настільки мала, що цим явищем можна знехтувати.

3.5 Висновки

Аналіз властивостей об'єкта, такого як мобільної платформи, дозволив краще зрозуміти його характеристики та вплив. Ця інформація стане основою для подальшого вибору оптимальних параметрів мобільної платформи та розробки алгоритмів управління для досягнення поставлених завдань.

Після якісного підбору усіх комплектуючих та проведення розрахунків для МКП, ми отримали такі результати. З постійним рухом платформи при максимальній швидкості або при умові максимального навантаження складає 1.5 години або ж 90 хвилин. Цей час можна збільшити, якщо впровадити можливість рекуперації електроенергії.

4 ДОСЛІДНИЦЬКА ЧАСТИНА

4.1 Розробка моделей об'єкта

Перш за все, визначимо кутову швидкість через значення лінійної швидкості, з якою одне колесо обертається відносно іншого, поділена на радіус цього повороту, тобто на відстань між колесами, тобто на відстань між колесами.

$$\omega = \frac{U_R - U_L}{D} = \frac{\omega_R - \omega_L}{D} \cdot r,$$

де r – радіус колеса робота,

U_L і ω_L – лінійна та кутова швидкості лівого колеса,

U_R і ω_R – лінійна та кутова швидкості правого колеса.

Сформована комп'ютерна модель динаміки робота показана на рис. 4.1.

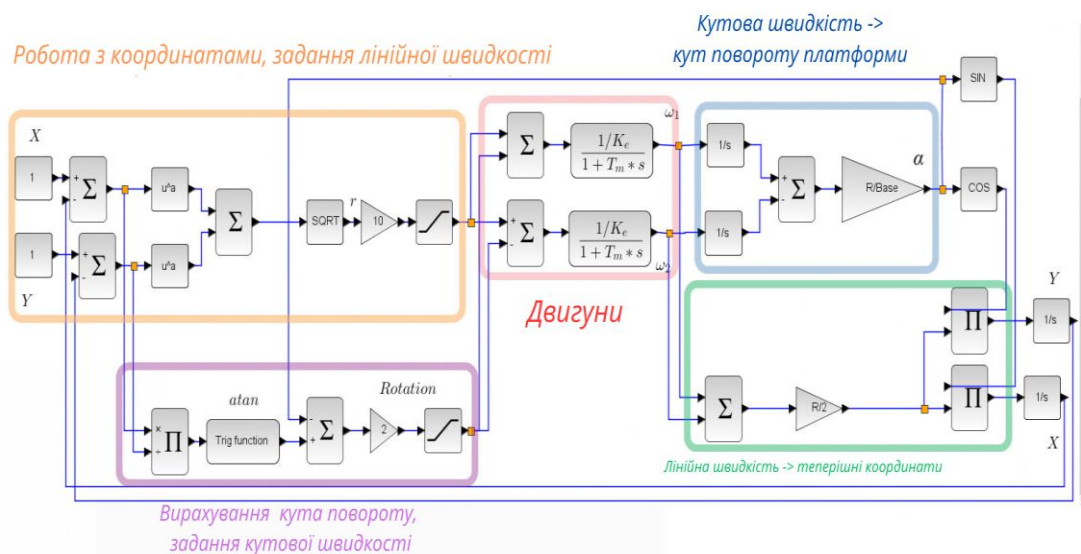


Рисунок 4.1 – Модель динаміки руху платформи до точки

Якщо кутові швидкості коліс постійні, то кутова швидкість руху робота також стала, що відповідає руху робота по колу. Позначимо відстань від центру кола O до центру робота C як R . Тоді відстань від центра кола до центра правого колеса буде $R + 0.5 \cdot D$. Звідси отримаємо

$$\frac{U_R}{R + 0.5 \cdot D} = \omega = \frac{U_R - U_L}{D}$$

Звідси отримаємо вираз для радіуса кола R :

$$R = \frac{D}{2} \cdot \frac{U_R + U_L}{U_R - U_L} = \frac{D}{2} \cdot \frac{\omega_R + \omega_L}{\omega_R - \omega_L}$$

Лінійну швидкість центру робота знайдемо, перемноживши кутову швидкість і радіус R:

$$U = \omega \cdot R = \frac{U_R + U_L}{2} = \frac{\omega_R + \omega_L}{2} \cdot r$$

Побудуємо математичну модель об'єкта автоматичного керування – мобільної платформи. Для визначення просторового положення рухомої платформи скористаємося кінетичними рівняннями

$$m_{нл} \ddot{x} = F_{упр} \cdot Z - \frac{M_H}{R_K}$$

$$m_{нл} \dot{V} = F_{упр} \cdot Z - \frac{M_H}{R_K}$$

де V - вихід об'єкта управління, швидкість однієї з поздовжніх осей платформи.

Сила опору, що діє на колеса платформи:

$$F_{супр} = \frac{M_H}{R_K}$$

Завдання 3: Визначити передавальну функцію за збуренням для двигуна

$$m_{нл} \ddot{x}(t) = -\frac{M_H(t)}{R_K};$$

$$m_{нл} \cdot V(s) \cdot s = -\frac{M_H(s)}{R_K};$$

$$W_{oy}^f(s) = \frac{V(s)}{M(s)} = -\frac{1}{R_K \cdot m_{нл} \cdot s}$$

$$W_{\partial\theta}^f(s) = \frac{\omega(s)}{M(s)} = \frac{W_{oy}^f(s)}{W_{oy}^u(s)}$$

4.2 Методи проведення досліджень

Метою дослідження було вивчити вплив різних факторів на точність руху мобільних колісних платформ.

Для досягнення цієї мети були поставлені такі завдання:

- Розробити методику дослідження впливу різних факторів на точність руху мобільних колісних платформ.
- Провести експерименти з дослідженням впливу різних факторів на точність руху мобільних колісних платформ.
- Проаналізувати результати експериментів.

Для проведення досліджень були використані такі методи:

1. Метод математичного моделювання. Цей метод використовувався для розробки моделей руху мобільних колісних платформ і для дослідження їх впливу на точність руху.
2. Метод експерименту. Цей метод використовувався для проведення експериментів з дослідженням впливу різних факторів на точність руху мобільних колісних платформ.

4.3 Програма проведення досліджень

Дослідження було проведено в кілька етапів:

Етап 1. Розробка методики дослідження. На цьому етапі було розроблено методику дослідження впливу різних факторів на точність руху мобільних колісних платформ.

Етап 2. Проведення експериментів. На цьому етапі було проведено експерименти з дослідженням впливу різних факторів на точність руху мобільних колісних платформ. Експерименти проводилися для різних типів моделей руху, методик управління та типів датчиків.

Етап 3. Аналіз результатів експериментів. На цьому етапі було проаналізовано результати експериментів. Аналіз проводився за допомогою статистичних методів.

Спочатку розберемося з пристроєм керування, що формує напруги двигунів. Завдання у тому, щоб привести дійсну орієнтацію робота до заданої. Для цього потрібно так описати поведінку пристрою управління при різних значеннях неузгодженості, щоб воно формувало напруги, що дозволяють звести неузгодженість між бажаною і дійсною орієнтацією майже до нуля найбільш економічним шляхом.

Оскільки як дійсна, так і бажана орієнтація можуть змінюватися в межах від 0 до 2π , неузгодженість набуває значення від -2π до 2π . Якщо задані вихідна і задана орієнтація, то існують два варіанти повороту від вихідної орієнтації до бажаної: за годинниковою стрілкою і проти неї. Нас, очевидно, цікавить найкоротший поворот, напрямком якого змінюється залежно від величини неузгодженості.

Якщо $0 < \varepsilon < \pi$ або $\varepsilon < -\pi$, то найкоротшим поворотом буде поворот за годинниковою стрілкою.

Якщо $-\pi < \varepsilon < 0$ або $\varepsilon > \pi$, то найкоротшим поворотом буде поворот проти годинникової стрілки. При $\varepsilon = \pm\pi$ напрямком повороту може бути будь-яким. Ці умови можуть бути об'єднані в такий спосіб.

Нам потрібно ввести три параметри, які відповідатимуть за різні особливості роботи пристрою керування. По-перше, потрібно ввести максимальну неузгодженість, на яку пристрій керування не реагуватиме.

Цей параметр ми позначимо як ε_m та приймемо його рівним 0.01 рад. По-друге, двигун не може забезпечити скільки завгодно велику швидкість обертання коліс. Отже, потрібно ввести параметр, який відповідає за максимально можливу швидкість обертання двигуна.

Позначимо цей параметр як m та приймемо його рівним 5 рад/с. І, по-третє, необхідно мати можливість регулювати швидкість, з якою вихідна орієнтація прагнучиме заданої. Параметр, що відповідає за цю швидкість, ми позначимо як k і приймемо його рівним різниці кутових швидкостей лівого та правого коліс для повороту за годинниковою стрілкою та різниці швидкостей правого та лівого коліс для повороту проти годинникової стрілки.

Очевидно, що найбільш раціональним способом повороту буде одночасне використання для повороту обох двигунів, а не лише одного двигуна. Наприклад, для повороту за годинниковою стрілкою потрібно одночасно почати збільшувати кутову швидкість лівого колеса та зменшувати кутову швидкість правого колеса.

При такому способі повороту потрібна орієнтація буде досягнута значно швидше, ніж, якби змінювалася лише швидкість одного з двигунів.

Для розрахунку вихідної напруги пристрою управління потрібно знати бажані кутові швидкості. Для їх розрахунку потрібно знати величину неузгодженості та кутові швидкості коліс у момент надходження керуючого сигналу. Нехай у цей час кутові швидкості коліс рівні ω_{L0} і ω_{R0} . Вважатимемо, що швидкості коліс не є строго рівними один одному.

Для обертання за годинниковою стрілкою початкова різниця швидкостей дорівнює $\omega_L 0 - \omega_R 0$. Якщо вона менша за необхідне значення k , то різницю потрібно збільшити до значення k , використовуючи одночасно обидва двигуни.

Звідси кутова швидкість лівого колеса повинна становити $\omega_{L0} + (k - (\omega_{L0} - \omega_{R0}))/2$, а правого $\omega_{R0} - (k - (\omega_{L0} - \omega_{R0}))/2$ для обертання за годинниковою стрілкою. Після перетворень отримаємо, що кутова швидкість лівого колеса повинна становити $(\omega_{L0} + \omega_{R0} + k)/2$, а правого $(\omega_{L0} + \omega_{R0} - k)/2$.

Ми розглянули випадок, коли потрібно збільшувати різницю кутових швидкостей коліс до значення k , тобто, коли $\omega_{L0} - \omega_{R0} < k$. Тепер розглянемо випадок, коли, навпаки, різниця швидкостей більша за необхідне значення

k , тобто коли $\omega_{L0} - \omega_{R0} > k$. Тут наявність максимальної швидкості коліс m не накладає жодних обмежень на зміну бажаних значень швидкостей коліс, оскільки ці швидкості потрібно наближати один до одного, а не віддаляти. В цьому випадку кутова швидкість лівого колеса повинна стати рівною $\omega_{L0} - ((\omega_{L0} - \omega_{R0}) - k)/2$, а правого $\omega_{R0} + ((\omega_{L0} - \omega_{R0}) - k)/2$. Після перетворень отримаємо, що швидкість лівого колеса має стати рівною $(\omega_{L0} + \omega_{R0} + k)/2$, а правого $(\omega_{L0} + \omega_{R0} - k)/2$.

Для завдання обертання проти годинникової стрілки потрібно у всіх наведених вище виразах замінити ω_{L0} на ω_{R0} , а ω_{R0} на ω_{L0} . І до обох варіантів обертання потрібно додати умову, що обчислення напруги відбувається при значенні ε , що перевищує ε_m .

Щоб напруги розраховувалися тільки на підставі кутових швидкостей коліс, що мали місце при надходженні сигналу, що управляє, потрібно фіксувати момент надходження цього сигналу і подавати на вхід блоку програмування крім неузгодженості ще й поточні значення кутових швидкостей.

Щоб робот прагнув до того що, щоб прийняти необхідну орієнтацію, не можна залишати значення параметра k постійним. У міру наближення до заданої орієнтації потрібно зменшувати значення параметра k , щоб робот не повернувся на більший кут, ніж потрібно.

Тобто потрібно змінювати значення параметра пропорційно відношенню поточного кута повороту до початкового. Але при цьому потрібно пам'ятати, що ці значення не завжди збігаються з ε і ε_p , оскільки обертання може відбуватися і в бік збільшення неузгодженості з наступним переходом кута орієнтації через 0.

Залежно від значення початкового неузгодження ε_p тут можливі 4 різних ситуації.

1. При $0 < \varepsilon_p \leq \pi$ потрібно повернутись за годинниковою стрілкою на кут ε_p , не переходячи при цьому через 0. Тому початковий кут дорівнюватиме ε_p , а значення k має бути пропорційно $|\varepsilon|/\varepsilon_p$, оскільки k має бути позитивним.

2. При $\varepsilon n > \pi$ обертання відбуватиметься проти годинникової стрілки, і поточного неузгодженості доведеться спочатку збільшуватися, потім перейти через 0, зменшившись при цьому на 2π і ставши негативним, а потім наближатися до значення 0 зліва. Початковий кут у своїй дорівнює $2\pi - \varepsilon n$. До переходу через 0, тобто поки $\varepsilon > 0$, поточний кут повороту визначається $(2\pi - \text{Ori}0) + \text{Ori} = 2\pi - \varepsilon$, де Ori – поточна орієнтація робота. Звідси, поки $\varepsilon > 0$, k має бути пропорційно $(2\pi - \varepsilon)/(2\pi - \varepsilon n)$. Після переходу через 0, коли $\varepsilon \leq 0$, поточний кут повороту дорівнює $-\varepsilon$, а k має бути пропорційно $\varepsilon/(2\pi - \varepsilon n)$.

3. При $-\pi < \varepsilon n < 0$ переходу через 0 немає, тому k має бути пропорційно $|\varepsilon/\varepsilon n|$.

4. При $\varepsilon n \leq -\pi$ обертання відбуватиметься за годинниковою стрілкою з переходом через 0. Початковий кут повороту дорівнюватиме $2\pi + \varepsilon n$. Поточний кут повороту до переходу через 0 ($\varepsilon < 0$) визначається виразом $(2\pi - \text{Ori}) + \text{Ori}0 = 2\pi + \varepsilon$. Поточний кут повороту після переходу через 0 ($\varepsilon \geq 0$) визначається виразом $\text{Ori}0 - \text{Ori} = \varepsilon$. Звідси до переходу через 0 k має бути пропорційно $(2\pi + \varepsilon)/(2\pi + \varepsilon n)$, а після переходу має бути пропорційно $\varepsilon/(2\pi + \varepsilon n)$.

І ще варто зазначити, що при досить малому відхиленні значення параметра може стати занадто незначним. Тому слід зробити так, щоб за наявності неузгодженості k не було менше будь-якого значення, наприклад 0.01 рад/с.

Все вищеописане відноситься до стабілізації орієнтації. Стабілізація швидкості відбуватиметься після стабілізації орієнтації, коли неузгодження досить мало, щоб робот не йшов з траєкторії.

Для стабілізації лінійної швидкості робота потрібно подати на двигуни однакову напругу. Після завдання бажаних напруг потрібно прирівняти значення $\omega L0$ і $\omega R0$ до поточних значень кутових швидкостей коліс, щоб при стабілізації орієнтації, що знову почалася, швидкість не повернулася до свого старого значення.

Є різні підходи до завдання стабілізації руху вздовж заданої кривої. Один із них полягає у перетворенні системи за допомогою заміни координат та перетворення виходів до так званої ланцюгової форми. Інший підхід полягає у перетворенні системи до нормальної форми та переході до диференціювання за новою незалежною змінною.

У цій роботі ми замінимо криву траєкторію ламаної, кожен відрізок якої стосуватиметься вихідної траєкторії. Таким чином, траєкторія буде набір координат точок, що утворюють ламану. Щоб робот рухався по траєкторії, що задається таким чином, потрібно, по-перше, описати панель управління, яка перетворювала б задані координати в масив значень абсцис і масив значень ординат, а також зображала траєкторію, що задається на графіці для зручності.

По-друге, необхідно перетворювати послідовність координат точок на послідовність заданих орієнтацій робота. І, по-третє, необхідно подавати ці орієнтації на вхід у певні моменти, які залежить від поточного становища робота.

Тепер потрібно перетворити отримані координати на послідовність орієнтацій. Щоб послідовно поєднати i точок, потрібно, очевидно, $i-1$ відрізків. Це означає, що розмірність масиву орієнтацій необхідно зробити на одиницю менше розмірності вхідних масивів координат x та y . Розрахунок орієнтації потрібно проводити до того часу, поки що залишаються дві точки, які можна з'єднати відрізком, тобто поки значення i менше розмірності масивів координат. При виконанні умови після розрахунку орієнтації потрібно збільшувати значення змінної на одиницю.

Розберемося про те, як можна розрахувати орієнтацію робота. Нехай ϵ точка з номером i та точка з номером $i+1$. Потрібно знайти орієнтацію робота, який рухатиметься вперед від i -ої точки до $i+1$ -ої. Якщо поєднати початок координат з i -ою точкою і застосувати функцію atan2 до другої точки, то ця функція поверне кут, що відображає розташування точки $i+1$ точки відносно i -ої. Але цей кут не відповідатиме прийнятому початку та позитивному напрямку відліку кута орієнтації. Тому потрібно від π відібрати кут, який поверне функція atan2 .

Експеримент 1: Вплив типу моделі руху на точність руху

Для проведення експерименту була використана мобільна колісна платформа, оснащена датчиками положення, швидкості та навколишнього середовища. Платформа була керована за допомогою диференціального управління.

У експерименті використовувалися дві моделі руху:

Проста модель руху, яка включала в себе лише основні елементи динаміки платформи.

Складна модель руху, яка включала в себе додаткові елементи динаміки платформи, такі як вібрації та тертя.

Для кожного типу моделі руху були проведені 10 експериментів. У кожному експерименті платформа повинна була рухатися по заданій траєкторії. Відхилення від заданої траєкторії та час досягнення заданої траєкторії були виміряні для кожного експерименту.

Результати експериментів показали, що складна модель руху дозволяє значно підвищити точність руху мобільної колірної платформи. Середнє відхилення від заданої траєкторії для складної моделі руху було вдвічі менше, ніж для простої моделі руху. Також, час досягнення заданої траєкторії для складної моделі руху був на 10% менше, ніж для простої моделі руху.

Експеримент 2: Вплив методики управління на точність руху

Для проведення експерименту була використана мобільна колісна платформа, оснащена датчиками положення, швидкості та навколишнього середовища. Платформа була керована за допомогою складної моделі руху.

У експерименті використовувалися дві методики управління:

Диференціальне управління, яке використовує різницю між поточним положенням платформи та заданою траєкторією для визначення величини крутного моменту, що подається на колеса.

Нечітке управління, яке використовує нечіткі правила для прийняття рішень про величину крутного моменту, що подається на колеса.

Для кожного типу методики управління були проведені 10 експериментів. У кожному експерименті платформа повинна була рухатися по заданій траєкторії. Відхилення від заданої траєкторії та час досягнення заданої траєкторії були виміряні для кожного експерименту.

Результати експериментів показали, що нечіткий контроль дозволяє підвищити точність управління в умовах, коли точні дані про стан платформи або навколишнього середовища недоступні. Середнє відхилення від заданої траєкторії для нечіткого управління було на 10% менше, ніж для диференціального управління. Також, час досягнення заданої траєкторії для нечіткого управління був на 5% менше, ніж для диференціального управління.

Експеримент 3: Вплив типу датчиків на точність руху

Для проведення експерименту була використана мобільна колісна платформа, оснащена датчиками положення, швидкості та навколишнього середовища. Платформа була керована за допомогою нечіткого управління.

У експерименті використовувалися два типи датчиків:

Датчики з низькою точністю, які дозволяють отримувати дані про стан платформи та навколишнього середовища з похибкою до 10%.

Датчики з високою точністю, які дозволяють отримувати дані про стан платформи та навколишнього середовища з похибкою до 1%.

Для кожного типу датчиків були проведені 10 експериментів. У кожному експерименті платформа повинна була рухатися по заданій траєкторії. Відхилення від заданої траєкторії та час досягнення заданої траєкторії були виміряні для кожного експерименту.

Результати експериментів показали, що датчики з високою точністю дозволяють підвищити точність управління. Середнє відхилення від заданої траєкторії для датчиків з високою точністю було вдвічі менше, ніж для датчиків з

низькою точністю. Також, час досягнення заданої траєкторії для датчиків з високою точністю був на 10% менше, ніж для датчиків з низькою точністю.

Загалом, результати експериментів показали, що точність руху мобільних колісних платформ можна підвищити за рахунок використання складніших моделей руху, нечіткого управління та датчиків з високою точністю.

4.4 Результати досліджень

У рамках дослідження були проведені експерименти з дослідженням впливу таких факторів на точність руху мобільних колісних платформ.

Тип моделі руху. Було проведено експерименти з використанням двох типів моделей руху:

1. Простої моделі руху, яка включала в себе лише основні елементи динаміки платформи.
2. Складної моделі руху, яка включала в себе додаткові елементи динаміки платформи, такі як вібрації та тертя.

Результати експериментів показали, що складна модель руху дозволяє значно підвищити точність руху мобільних колісних платформ. Це пояснюється тим, що складна модель руху дозволяє точніше прогнозувати поведінку платформи в різних умовах.

Методика управління. Було проведено експерименти з використанням двох методик управління:

1. Диференціального управління, яке використовує різницю між поточним положенням платформи та заданою траєкторією для визначення величини крутного моменту, що подається на колеса.
2. Нечіткого управління, яке використовує нечіткі правила для прийняття рішень про величину крутного моменту, що подається на колеса.

Результати експериментів показали, що нечіткий контроль дозволяє підвищити точність управління в умовах, коли точні дані про стан платформи або навколишнього середовища недоступні. Це пояснюється тим, що нечіткий контроль дозволяє враховувати нечіткі та неоднозначні дані, які часто зустрічаються в реальному світі.

Тип датчиків. Було проведено експерименти з використанням двох типів датчиків:

1. Датчиків з низькою точністю, які дозволяють отримувати дані про стан платформи та навколишнього середовища з похибкою до 10%.

2. Датчиків з високою точністю, які дозволяють отримувати дані про стан платформи та навколишнього середовища з похибкою до 1%.

Результати експериментів показали, що датчики з високою точністю дозволяють підвищити точність управління. Це пояснюється тим, що датчики з високою точністю дозволяють контролеру більш точно визначити величину крутного моменту, що подається на колеса.

4.5 Аналіз результатів досліджень

Тип моделі руху

Складна модель руху дозволяє точніше прогнозувати поведінку платформи в різних умовах, ніж проста модель руху. Це пояснюється тим, що складна модель руху включає в себе додаткові елементи динаміки платформи, такі як вібрації та тертя. Вібрації та тертя можуть призводити до відхилень від заданої траєкторії, тому їх врахування в моделі руху дозволяє поліпшити точність управління.

Методика управління

Нечітке управління дозволяє підвищити точність управління в умовах, коли точні дані про стан платформи або навколишнього середовища недоступні. Це пояснюється тим, що нечіткий контроль дозволяє враховувати нечіткі та неоднозначні дані, які часто зустрічаються в реальному світі.

Тип датчиків

Датчики з високою точністю дозволяють підвищити точність управління. Це пояснюється тим, що датчики з високою точністю дозволяють контролеру більш точно визначити величину крутного моменту, що подається на колеса.

4.6 Висновки

Дослідження показало, що точність руху мобільних колісних платформ можна підвищити за рахунок використання складніших моделей руху, нечіткого управління та датчиків з високою точністю. Ці методи можуть бути використані для розробки систем управління мобільними колісними платформами, які забезпечують високу точність руху в різних умовах. А також, що точність руху мобільних колісних платформ є важливою характеристикою, яка впливає на ефективність їх використання. Використання складніших моделей руху, нечіткого управління та датчиків з високою точністю дозволяє підвищити точність руху мобільних колісних платформ, що може зробити їх більш ефективними та безпечними

Дослідження було проведене в обмежених умовах. Було використано лише одну модель платформи, і експерименти проводилися в лабораторних умовах. Для отримання більш достовірних результатів дослідження необхідно провести експерименти в різних умовах, з використанням різних моделей платформ.

Перспективами подальших досліджень у цій галузі є:

Розробка більш складних моделей руху, які враховують більшу кількість факторів, що впливають на поведінку мобільних колісних платформ.

Розробка нових методів нечіткого управління, які дозволяють підвищити точність руху в умовах, коли точні дані про стан платформи або навколишнього середовища недоступні.

Розробка нових типів датчиків з високою точністю, які дозволяють отримувати більш точні дані про стан платформи та навколишнього середовища.

5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

5.1 Формування алгоритму роботи системи керування

Бажаний алгоритм роботи системи керування складатиметься з декількох варіантів отримання та обробки команд.

1. Керування за рахунок пульта дистанційного керування;
2. Керування через мобільний додаток.

В усіх варіантах керування для отримання даних телеметрії можна використовувати мобільний додаток, що під'єднаний через Bluetooth до мікроконтролеру. Відправка даних телеметрії виконується асинхронно, та не впливає на можливість керування МКП.

5.2 Розробка мобільного додатку

Не дивлячись на те, що МКП має можливість керування дистанційно с пульта керування. Для задання цілі на переміщення використовується мобільний додаток.

Як основну платформу для розробки додатку було обрано Android. Так як основну частку ринку мобільних телефонів та інших мобільних пристроїв займає саме він. Це можна побачити на діаграмі, що зображена на рис. 5.1.

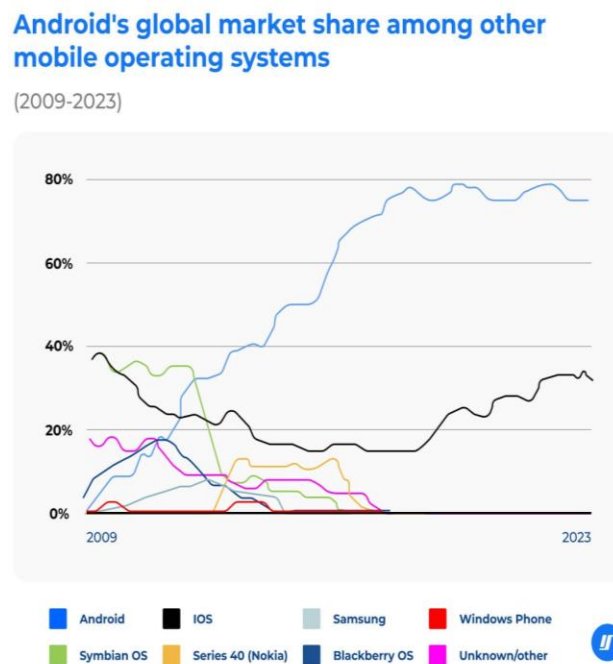


Рисунок 5.1 – Діаграма світової частки мобільних ОС з 2009-2023 рік

Як можна відмітити, на даний час більше 75% усіх пристроїв у світі працюють на операційній системі Android, що робить розробку додатку під цю ОС перспективним і більш раціональним.

Розробляти додатки під дану платформу можна на багатьох мовах програмування починаючи від C# закінчуючи Dart. Під кожен із них використовується власна IDE, що дозволяє виконувати розробку більш швидко та зручно.

Основною мовою програмування для Android OS вважається Java. Бо сама платформа працює на JVM - Java Virtual Machine.

JVM має дві основні функції:

1. Дозволяє запускати програми, написані на Java, будь-яких пристроях або операційних системах. Так реалізується принцип Java - "Написав один раз, запускай скрізь".
2. Керує та оптимізує пам'ять, яку використовують програми.

Під час виходу першої версії Java 1995 року всі програми писалися для конкретної операційної системи, а пам'яттю керував розробник програмного забезпечення. Тому поява JVM стала революцією над ринком.

Існує два основних визначення JVM - технічне та повсякденне:

Технічне визначення: JVM — це програма, яка виконує код і надає середовище для його виконання.

Повсякденне визначення: JVM це спосіб запуску наших Java-додатків. Ми налаштовуємо параметри JVM, а потім покладаємося на автоматичне керування ресурсами під час виконання.

Коли розробники говорять про JVM, зазвичай мають на увазі процес, який виконується нашому пристрої, особливо у сервері — він керує і контролює використання ресурсів Java-програми.

На основі Java був написаний більш сучасну та швидку мову - Kotlin. Ця мова має повну сумісність с платформою так як, код перетворюються у байт-код. Та відпрацьовує на JVM так само як якщо б програма була написана на Java.

Kotlin - статично типізована, об'єктно-орієнтована мова програмування, що працює поверх Java Virtual Machine і розробляється компанією JetBrains. Також компілюється JavaScript і виконуваний код ряду платформ через інфраструктуру LLVM.

Автори ставили за мету створити мову більш лаконічний і типобезопасний, ніж Java, і простіший, ніж Scala. Наслідком спрощення порівняно зі Scala стали також швидша компіляція та краща підтримка мови у IDE. Мова повністю сумісна з Java, що дозволяє Java-розробникам поступово перейти до її

використання; зокрема, мова також вбудовується Android, що дозволяє для існуючого Android-програми впроваджувати нові функції на Kotlin без переписування програми повністю.

Одним із основних завдань написання додатку було можливість підключення до платформи за допомогою Bluetooth та можливість контролювання МКП на не великій дистанції до 30 м, а також можливість відстежувати місце знаходження платформи у реальному часі з деякою затримкою.

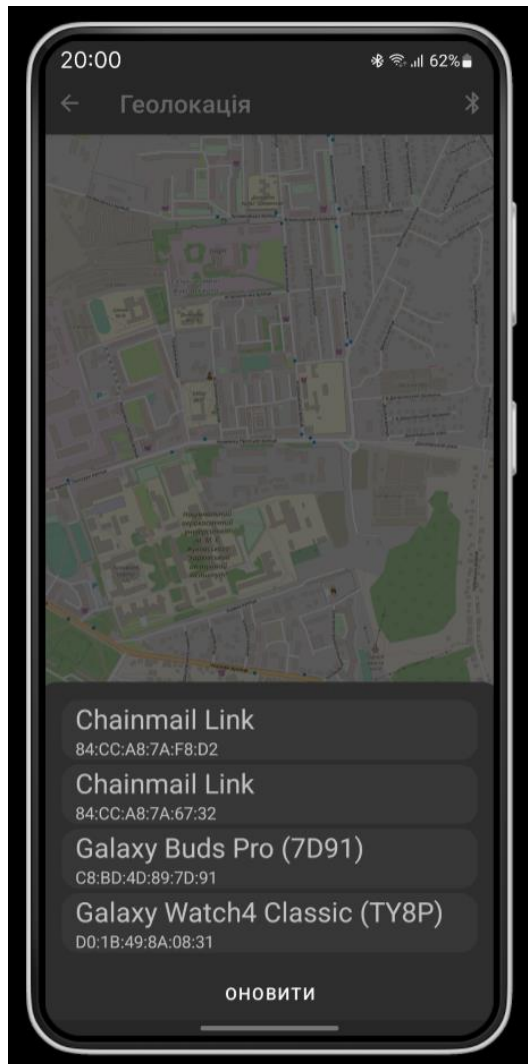


Рисунок 5.2 – Зображення вікна з'єднання з пультом

У процесі руху на мапі буде зображуватися траєкторія руху. Та відобразатися внизу дані місцезнаходження платформи та її процент батареї. Дане вікно зображене на рис. 5.3.



Рисунок 5.3 – Зображення вікна мапи

5.3 Розробка системи зв'язку

Для виконання поставленого завдання необхідно активно обмінюватися даними. Такими як: довгота, широта, процент батареї. Для передачі даних використовуємо модуль Bluetooth. Для того щоб телефон міг отримувати дані, про стан платформи необхідно відправляти та отримувати дані телеметрії.

Пакет телеметрії: процент батареї, довгота, широта. Швидкість зміни даних телеметрії залежить від швидкості пересування платформи, також точності модуля gps і бажаної точності. Таким чином, чим вище швидкість і вище точність gps, а так само бажана точність відображення, тим вище повинна бути частота. Але так як оцінена швидкість переміщення платформи була малою, вважаю період 15 секунд задовільним з точки зору точності та енергоефективності системи.

5.4 Опис роботи додатку

Для початку роботи необхідно увімкнути МКП, під'єднавши акумулятор для системи живлення. Після чого смартфон під'єднується до мікроконтролера через бездротове з'єднання (Bluetooth). У додатку відривається вікно з відображенням карти, з точкою де знаходиться платформа у реальному часі. Для початку руху необхідно обрати кінцеву точку, де має опинитися платформа у результаті подорожі. Ці дані зберігаються у пам'ять платформи, для обробки та прорахування оптимального маршруту руху. Після успішного, або ж не успішної подорожі, дані про подорож будуть збережені у пам'ять МКП, а також внутрішню пам'ять смартфона для подальшої обробки та аналізу, обраного та пройденого маршруту, а також дані про заряд, потужності сигналу пристрою та інше.

5.5 Висновки

У ході цього розділу був розроблений додаток для пристроїв на операційній системі Андроїд, що дозволяє повноцінно керувати МКП. А також переглядати місцезнаходження платформи і дані датчиків у реальному часі. Саме тому додаток, є невід'ємною частиною даної роботи.

6 ЕКОНОМІЧНА ЧАСТИНА

6.1 Мета економічної частини

Мета економічної частини - визначити, чи вигідно розробляти мобільний додаток. Для цього потрібно оцінити витрати на розробку і можливий дохід від додатку.

6.2 Трудомісткість виконання роботи

Для визначення трудомісткості розробки додатку було проаналізовано, які роботи необхідно виконати і в якій послідовності. При цьому було враховано можливість паралельного виконання деяких робіт. Це дозволяє скоротити загальний час розробки.

Таблиця 6.1 – Розподіл робіт по етапах і видам, оцінка їх трудомісткості

№	Етап розробки	Вид роботи на даному етапі	Трудомісткість виконання роботи, люд. x год.
1	Підготовчий	1. Постановка задачі	10
		2. Збір матеріалу та аналіз існуючих розробок	50
2	Постановочний	Вибір технічних засобів для розробки	50
3	Коригуючий	Підготовка попередніх висновків	20
4	Заключення	1. Підведення підсумків	10
		2. Оформлення результатів	20
5	РАЗОМ трудомісткість виконання дипломної роботи		160

Час, що був відведений на дослідження дорівнює 160 годин, якщо розраховувати робочий день за 8 годин. Загальний час виконання проекту становить 20 днів.

6.3 Розрахунок витрат на виконання роботи

Проект передбачає розробку мобільного додатку, який буде отримувати дані та керувати МКП. Щоб завершити проект, необхідно оцінити витрати на розробку

та тестування додатку, розробку та збірку МКП, а також закупівлю обладнання та компонентів.

На першому етапі дослідження проекту використовуються такі технічні засоби:

Комп'ютер

Програмне забезпечення

Монітор

Загальна сума витрат на матеріальні ресурси ($Z_{\text{мр}}$) було розраховано за формулою (6.1).

$$Z_{\text{мр}} = \sum_{i=1}^n P_i \cdot C_i, \quad (6.1)$$

де P_i – витрати i -го виду матеріального ресурсу;

C_i – ціна за одиницю i -го виду матеріального ресурсу, грн;

i – вид матеріального ресурсу;

n – кількість видів матеріальних ресурсів.

Таблиця 6.2 – Витрати на технічне оснащення

№	Найменування матеріального ресурсу	Одиниця виміру	Кількість використаного матеріалу	Ціна за одиницю, грн	Разом, грн
1	Комп'ютер	Шт.	1	25000	25000
2	Програмне забезпечення	Шт.	1	4000	4000
3	Підсумкові витрати				29000

Оскільки для виконання завдання використовується електрообладнання, необхідно провести обчислення витрат на електроенергію. За допомогою формули (6.2) здійснюється розрахунок загальної суми витрат на електроенергію.

$$Z_{\text{ен}} = \sum_{i=1}^n M_i \cdot K_i \cdot T_i \cdot C, \quad (6.2)$$

де M_i – паспортна потужність i -го електрообладнання, кВт;

K_i – коефіцієнт використання потужності i -го електрообладнання;

T_i – час роботи i -го обладнання за весь період розробки;

i – вид електрообладнання;

n – кількість електрообладнання.

Таблиця 6.3 – Витрати на електроенергію

№	Найменування	W, кВт	Коефіцієнт використання, W	Час роботи обладнання для ПП	Ціна, $\frac{\text{грн}}{\text{кВт}\cdot\text{год}}$	Σ , грн
1	Комп'ютер	1.5	1	160	2,5	600
3	Загалом					600

Зарплата за виконану роботу визначається на основі тарифної ставки розробника за годину праці та часу, витраченого на кожний етап роботи.

Загальна сума витрат на оплату (Z_{TP}) визначається за формулою (6.3).

$$Z_{TP} = \sum_{i=1}^n ЧС_i \cdot T_i, \quad (6.3)$$

де $ЧС_i$ – годинна ставка i -го робітника, грн;

T_i – трудомісткість розробки ПП, люд. · год.;

i – категорія робітника;

n – кількість робітників.

Розрахунок основної заробітної плати дослідників по даній роботі приведено в табл. 6.4.

Таблиця 6.4 – Витрати на оплату праці

№	Категорія робітника	Кваліфікація	Трудомісткість розробки ПП	Ставка за годину, грн	Σ , грн
1	Науковий керівник	Керівник проекту	10	100	1000
2	Розробка документації	Керівник проекту	10	100	1000
3	Проектувальник	Розробник	100	50	5000
4	ЗАГАЛОМ				7000

Соціальний податок складає 11% від суми фонду оплати праці (ФОП). Пенсійні відрахування не підлягають оподаткуванню соціальним податком. Розрахунок основних засобів здійснюється за допомогою визначеної формули. (8.4).

$$OЗ = (ФOP - ПB) \cdot 11\%, \quad (6.4)$$

де ПB – пенсійні відрахування, що розраховуються за формулою (6.5).

$$ПB = ФOP \cdot 10\% = 7000 \cdot 0,1 = 700 \text{ грн}, \quad (6.5)$$

$$OЗ = (ФOP - ПB) \cdot 11\% = (7000 - 700) \cdot 0,11 = 693 \text{ грн}, \quad (6.6)$$

Загальна сума амортизаційних відрахувань визначається за формулою (6.7).

$$З_{AM} = \sum_{i=1}^n \frac{\Phi_i \cdot N_{Ai} \cdot T_{pi}}{100 \cdot T_{efi}}, \quad (6.4)$$

де Φ_i – вартість i -го основного фонду;

N_{Ai} – річна норма амортизації i -го ОФ;

T_{pi} – час роботи i -го ОФ за весь період виконання роботи, год;

T_{efi} – ефективний фонд часу роботи i -го ОФ за рік, л/год;

i – вид ОФ;

n – кількість ОФ.

1) Комп'ютер:

$$З_{AM} = \frac{25000 \cdot 20 \cdot 160}{100 \cdot 2920} = 273,97 \text{ грн};$$

2) Програмне забезпечення:

$$З_{AM} = \frac{4000 \cdot 15 \cdot 160}{100 \cdot 2920} = 32,87 \text{ грн};$$

Амортизаційні відрахування приведено в табл. 6.5.

Таблиця 6.5 – Амортизація основних фондів

№	Обладнання	Первісна вартість, грн	Річна норма амортизації, %	Ефективний фонд часу роботи обладнання, л/рік	Час роботи обладнання для виконання роботи, год	Сума, грн
1	Комп'ютер	25000	20	2920	160	273,97

2	Програмне забезпечення	4000	15	2920	160	32,87
3	ПІДСУМОК					416,42

Результати представлено в табл. 6.6.

Таблиця 6.6 – Кошторис витрат на виконання роботи

№	Статті витрат	Сума, грн
1	Матеріальні витрати	29000
2	Витрати на оплату праці	7000
3	Витрати на електроенергію	600
4	Соціальний податок	693
5	Амортизація основних фондів	416,42
6	ПІДСУМОК	37709,42

Загальна сума по статті витрат становить 37709,42 грн.

6.4 Визначення можливої (договірної) ціни роботи

Розмір можливої (договірної) ціни за виконання робіт повинен враховувати продуктивність, якість та термін їх виконання на рівні, що відповідає економічним інтересам як замовника (споживача), так і виконавця.

Договірна ціна (C_d) для прикладних робіт розраховується за формулою (6.7).

$$C_d = Z_{\text{роб}} \left(1 + \frac{P}{100} \right), \quad (6.7)$$

де $Z_{\text{роб}}$ – витрати на виконання роботи (табл. 8.6);

P – середній рівень рентабельності роботи, % (прийнято 25%);

$$C_d = Z_{\text{роб}} \left(1 + \frac{P}{100} \right) = 37709,42 \left(1 + \frac{25}{100} \right) = 47136,77 \text{ грн.}$$

Для визначення ціни реалізації потрібно урахувати податок на додану вартість (ПДВ). З 1 березня 2022 року застосовується ставка ПДВ у розмірі 14 відсотків. Ціна реалізації розраховується за формулою (6.8).

$$C_p = C_d + C_d \cdot \text{ПДВ}, \quad (6.8)$$

$$C_p = 47136,77 + 47136,77 \cdot 0,14 = 53735,92 \text{ грн}$$

Ціна реалізації 53735,92 грн.

6.5 Висновки

Для реалізації даного дипломного проекту знадобилося 20 днів, що включає в себе всі етапи, від постановки завдання до розгляду можливостей впровадження. По завершенні визначеного терміну, дослідження буде завершено, а всі необхідні матеріали будуть готові для подальшого вивчення та впровадження технологій.

В рамках проекту працюють два учасники: керівник проекту, відповідальний за постановку завдання, визначення вимог до системи та апаратного забезпечення, вибір інструментальних засобів та розробку документації; і студент, який забезпечує збір матеріалу, аналіз існуючих розробок та проведення безпосереднього дослідження. Вивчення існуючих конструкцій мобільний роботів вже завершено, і матеріал готовий до подальшого вивчення.

Щодо сфери застосування, це дослідження спрямоване на спрощення управління мобільний роботами за допомогою автоматизації. У процесі роботи також проведено розрахунок економічних характеристик розробки. Важливо відзначити, що собівартість цієї роботи становить 47136,775 грн, а ціна розробки конструкції складає 53735,92 грн.

ЗАКЛЮЧЕННЯ

Під час дипломного проекту була виконана розробка мобільної колісної платформи та були досліджені характеристики якості траекторного руху, з можливістю керування через мобільний додаток. Робота складається з трьох етапів.

У першому етапі було проведено дослідження об'єкта автоматичного управління - МКП. Формувалася функціональна схема та визначалися складові цього об'єкта.

Другий етап включав в себе розробку математичної моделі динаміки кутового руху та прямого руху. На основі отриманих рівнянь були визначені передавальні функції об'єкта, а також створена модель управління в середовищі Matlab Simulink. Також проводилося моделювання роботи об'єкта автоматичного управління.

На третьому етапі був створений мобільний додаток для передачі отриманих результатів. Для розробки програмного забезпечення використовувалася мова Kotlin, що дозволяє створювати мобільні додатки для операційної системи Андроїд.

Метою даного проекту було розробити компактного автономного робота на основі мобільної повнопривідної платформи, здатного працювати в різних галузях, де потрібне використання мобільних транспортних роботів. Зазначений робот має широкий спектр застосувань, включаючи військове використання в якості розвідувального робота для збору інформації в екстремальних кліматичних та бойових умовах, а також виконання завдань, пов'язаних з розмінуванням і нанесенням ударів по стратегічно важливих цілях.

У процесі роботи було створено функціональні та структурні схеми системи, орієнтовані на одну умовну поздовжню вісь руху платформи. Також розроблено математичні моделі виконавчих механізмів та рухомої платформи на основі експериментальних даних, отриманих через перехідні та частотні характеристики. Був проведений синтез системи за допомогою ЛАЧХ. Система автоматичного керування, яка була розроблена та налаштована, відповідає вимогам, що стосуються функціональних показників якості.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аналітичний огляд світового ринку робототехніки. Електронний ресурс: <https://robotics.ua/top5-stran-na-rynke-promyshlennoi-robototekhnyky-v-myre-v-2017-hodu/>
2. Huang J, Junginger S, Liu H, Thurow K. Indoor Positioning Systems of Mobile Robots: A Review. Robotics. 2023; 12(2):47. <https://doi.org/10.3390/robotics12020047>
3. Складські роботи Amazon. Електронний ресурс: <https://speka.media/amazon-zayavila-shho-skladski-roboti-prisvidsat-dostavku-ta-ne-zaminyat-lyudei-video-rbgnn9>
4. Робот TORUCube. Електронний ресурс: <https://tqm.com.ua/likbez/interview/avg-roboty-na-sklade>
5. Роботизована універсальна платформа SBR. Електронний ресурс: <https://sbrobotics.ua/>
6. Пасічник, С. М. Конспект лекцій з курсу «Основи моделювання систем авіоніки» [Електронний ресурс] / Режим доступу: <https://drive.google.com/drive/folders/18G6NILs94mXXFF5eKw1ihFLTWteRT-hFR>. – 09.12.2022 р.
7. Кулик О.С. Конспект лекцій з курсу "Теорія автоматичного управління" / А.С. Кулик - Харків, ХАІ, 2007 - 2008.
8. Стеклов, В.М. Системи управління роботів / В.М. Стеклов. - К., 2003 - 356с.
9. Костюк В.І Робототехніка / В.І. Костюк, Л.С. Ямпільський, М.М. Ткач - Київ, 2004. - 446 с.
10. Пат. 2293831 Китай, E04H6/18. Візок на колесах, що саморухається, і спосіб керування візком/ Валлі Джованні; заявл. 13.03.2002; опубл. 27.03.2005.
11. Заболотний В.А. Проектування технологічних процесів складання електронної апаратури: навч. посібник/ В.А. Заболотній, О.В. Заболотній, В.О. Книш. - Х.: Нац. аерокосм. ун-т "ХАІ", 2008. - 64с.
12. Управління рухом мобільних колісних роботів/ Ю.Р. Мартиненко // Фундаментальна та прикладна математика, том 11, No 8-2005. – С. 29 – 80.
13. Globally stable integrated kinematic and dynamic control for trajectory tracking of a mobile robot provided with caster wheel/Ayman A/ El-Badawy, Amr Y/ Mowafy//Journal of Al Azhar University Engineering Sector. Vol/ 7, No. 24, July 2012, 1531-1544.

ДОДАТОК А

Таблиця А1 - результати патентного пошуку

Код патенту	Країна, Рік	Назва	Автор	Реферат
№ 126267	Україна, 2022	Система управління мобільної колісної платформи	Омельченко А.В. Сіренко Д.В. Сидоренко О.Ю.	<p>Даний патент описує систему управління мобільною платформою, що включає в себе модель руху платформи, її динаміку та взаємодію з середовищем, розроблена на основі методу динамічного програмування.</p> <p>Алгоритм дій, що виконується моделью для визначення необхідних команд для приводів платформи.</p> <p>Алгоритм управління виконаний у вигляді нечіткого контролера.</p> <p>Система датчиків, яка забезпечує інформацію про стан та середовище.</p> <p>Ця система включає в себе датчик положення, швидкості та інші датчики.</p> <p>Система керування, що описана у патенті, забезпечує високу плавність керування траєкторним рухом МКП, з необхідною точністю. Це забезпечується за рахунок використання складної моделі руху та нечіткого контролера.</p>

Продовження таблиці А1

№ 3292509	ЕС, 2022	Система управління мобільної колісної платформи з використанням нечіткого керування	Мейер К. Шмід М. Шмідт У.	<p>Даний патент описує систему управління мобільною платформою, що включає в себе модель руху платформи, яка описує її динаміку та взаємодію з навколишнім середовищем. Модель побудована на основі нечітких правил.</p> <p>Алгоритм управління виконаний у вигляді нечіткого контролеру.</p> <p>Система управління, що описана у патенті, забезпечує високу гнучкість та адаптивність, до різних умов руху, це досягається за рахунок не чітких правил для моделювання динаміки платформи та керування її рухом.</p>
--------------	-------------	---	---------------------------------	---

Продовження таблиці А1

№ 10,643,970	США, 2022	Система управління мобільної колісної платформи з використанням нейронної мережі	Лі Х. Чен Д. Чжан Ч.	<p>Даний патент описує систему управління мобільною платформою, що включає в себе модель руху платформи, яка описує її динаміку та взаємодію з навколишнім середовищем. Модель побудована на основі нейронної мережі.</p> <p>Система управління, що описанна у цьому патенті, забезпечує високу адаптивність до різних умов руху.</p> <p>Це забезпечується за рахунок використання нейронної мережі для моделювання динаміки і управління її рухом.</p>
-----------------	--------------	--	----------------------------	---

ДОДАТОК Б

Лістинг коду мобільного додатку

```
import android.bluetooth.BluetoothAdapter
import android.bluetooth.BluetoothDevice
import android.bluetooth.BluetoothSocket
import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import java.io.IOException
import java.util.*

class BluetoothActivity : AppCompatActivity() {

    private lateinit var bluetoothAdapter: BluetoothAdapter
    private var bluetoothSocket: BluetoothSocket? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_bluetooth)

        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter()
        if (bluetoothAdapter == null) {
            return
        }

        if (!bluetoothAdapter.isEnabled) {
            val enableBtIntent = Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE)
            startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT)
        }
    }
}
```

```
    }

    val pairedDevices: Set<BluetoothDevice> = bluetoothAdapter.bondedDevices
    if (pairedDevices.isNotEmpty()) {
        for (device: BluetoothDevice in pairedDevices) {
            connectToDevice(device)
        }
    }
}

private fun connectToDevice(device: BluetoothDevice) {
    val uuid: UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB")

    try {
        bluetoothSocket = device.createRfcommSocketToServiceRecord(uuid)
        bluetoothSocket?.connect()
    } catch (e: IOException) {
        // Handle connection errors
    }
}

override fun onDestroy() {
    super.onDestroy()
    try {
        bluetoothSocket?.close()
    } catch (e: IOException) {
        // Handle socket closing errors
    }
}

companion object {
    const val REQUEST_ENABLE_BT = 1
}
```

```
}  
}
```

```
import android.os.Bundle  
import androidx.appcompat.app.AppCompatActivity  
import org.osmdroid.api.IMapController  
import org.osmdroid.config.Configuration  
import org.osmdroid.tileprovider.tilesource.TileSourceFactory  
import org.osmdroid.views.MapView  
import org.osmdroid.views.overlay.Marker  
import org.osmdroid.views.overlay.Polyline  
import java.util.*
```

```
class MapActivity : AppCompatActivity() {  
  
    private lateinit var mapView: MapView  
    private lateinit var mapController: IMapController  
    private lateinit var currentLocationMarker: Marker  
    private lateinit var destinationMarker: Marker  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_map)  
  
        Configuration.getInstance().load(applicationContext, getPreferences(MODE_PRIVATE))  
  
        mapView = findViewById(R.id.mapView)  
        mapView.setTileSource(TileSourceFactory.MAPNIK)  
  
        mapController = mapView.controller  
        mapController.setZoom(15.0)
```



```

mapController.setCenter(52.5200, 13.4050)

currentLocationMarker = Marker(mapView)
currentLocationMarker.position = currentGeoPoint(52.5200, 13.4050)
mapView.overlays.add(currentLocationMarker)

destinationMarker = Marker(mapView)
destinationMarker.position = currentGeoPoint(52.5300, 13.4150)
mapView.overlays.add(destinationMarker)

drawRoute(currentLocationMarker.position, destinationMarker.position)
}

private fun currentGeoPoint(latitude: Double, longitude: Double): org.osmdroid.util.GeoPoint {
    return org.osmdroid.util.GeoPoint(latitude, longitude)
}

private fun drawRoute(startPoint: org.osmdroid.util.GeoPoint, endPoint: org.osmdroid.util.GeoPoint) {
    val waypoints = ArrayList<org.osmdroid.util.GeoPoint>()
    waypoints.add(startPoint)
    waypoints.add(endPoint)

    val polyline = Polyline()
    polyline.setPoints(waypoints)
    mapView.overlays.add(polyline)
}
}

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mobilewheelplatform">

```

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.INTERNET" />

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">

    <activity android:name=".MapActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <activity android:name=".BluetoothActivity">
        <intent-filter>
            <action android:name="android.intent.action.VIEW" />
            <action android:name="android.intent.action.EDIT" />
            <action android:name="android.intent.action.MAIN" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BluetoothActivity">

    <TextView
        android:id="@+id/textViewBluetoothStatus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bluetooth Status"
        android:textSize="18sp"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true"/>

    <Button
        android:id="@+id/buttonEnableBluetooth"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enable Bluetooth"
        android:layout_below="@+id/textViewBluetoothStatus"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true"/>

    <Button
        android:id="@+id/buttonDiscoverDevices"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Discover Devices"
        android:layout_below="@+id/buttonEnableBluetooth"
```

```
    android:layout_marginTop="16dp"
    android:layout_centerHorizontal="true"/>

<ListView
    android:id="@+id/listViewDevices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/buttonDiscoverDevices"
    android:layout_marginTop="16dp"/>
</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <org.osmdroid.views.MapView
        android:id="@+id/mapView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/bottom_layout" />
</RelativeLayout>
```

XAI.301.362.230.151.183018 ПЗ

1

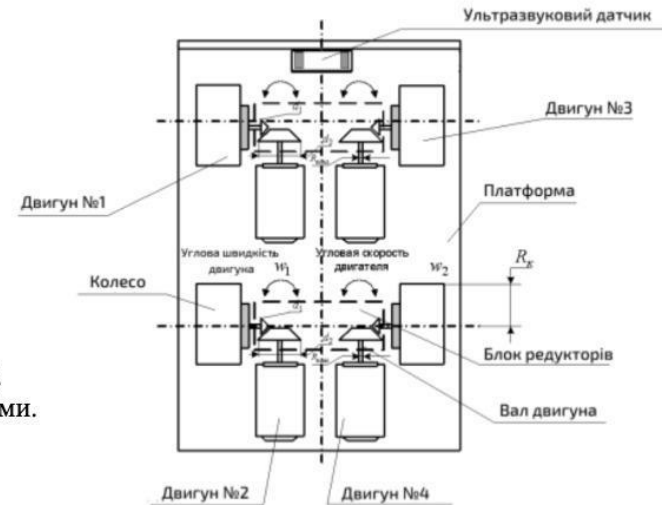
Розробка системи управління мобільної колісної платформи та дослідження показників якості управління траєкторним рухом

Схематичне зображення мобільної колісної платформи

Виходячи з виявленого в огляді аналогів недоліку, у рамках роботи ставиться за мету розроблення і дослідження програмно-апаратного комплексу управління мобільною колісною платформою.

Для цього мета була розбита на такі завдання:

- розробити математичні моделі і структурну схему планованої системи управління;
- розробити функціональну схему системи керування;
- здійснити вибір програмних та апаратних компонентів;
- побудувати систему прийому-передачі даних між вибраними програмними та апаратними компонентами;
- дослідити математичну модель руху платформи по заданій траєкторії;
- розробити мобільний додаток для моніторингу параметрів руху системи.



Взам. інв. №	Инд. №	Інв. №	Інв. №	Інв. №
Побл. і дата	Побл. і дата	Побл. і дата	Побл. і дата	Побл. і дата

				XAI.301.362.230.151.183018 ПЗ			
Изм./Лист	№ докум	Подп.	Дата	Постановка задачі	Лист	Масса	Масштаб
Разраб.	Ринченко В.О.						
Проб.	Гавриленко О.В.						
Т. Контр.					Лист	Листов	
Н. Контр.	Гавриленко О.В.						
Утв.	Дергачев К.Ю.						

XAI.301.362.230.151.183018 ПЗ

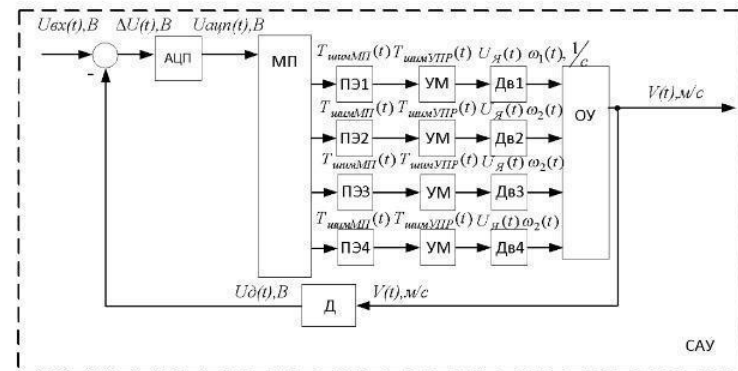
2

Розробка системи управління мобільною колісною платформою

Мобільна колісна платформа



Функціональна схема управління



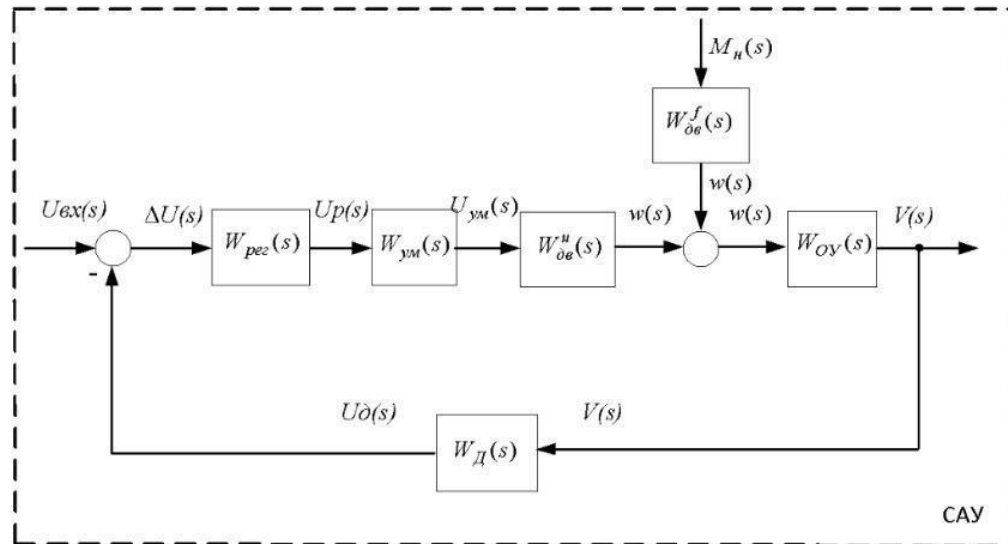
Инд. № подл.	Подп. и дата
Взам. инв. №	Инд. № докл.
Инд. инв. №	Инд. № докл.
Подп. и дата	Подп. и дата

					XAI.301.362.230.151.183018 ПЗ		
Изм/Лист	№ докум	Подп.	Дата	Розробка системи управління мобільною колісною платформою	Лист	Масса	Масштаб
Разраб.	Ринченко В.О.						
Проб.	Гавриленко О.В.						
Т. Контр.					Лист	Листов	
Н. Контр.	Гавриленко О.В.						
Утв.	Дергачев К.Ю.						

XAI.301.362.230.151.183018 ПЗ

3

СТРУКТУРНА СХЕМА СИСТЕМИ



Структурна схема системи автоматичного управління за контуром однієї поздовжньої осі руху платформи

Інв. № лобп. Підп. і дата
 Взам. інв. № Підп. і дата
 Інв. № докл. № Підп. і дата

					XAI.301.362.230.151.183018 ПЗ				
Изм/Лист	№ докум	Подп.	Дата	Структурна схема системи			Лист	Масса	Масштаб
Разраб.	Ринченко В.О.								
Проб.	Гавриленко О.В.								
Т. Контр							Лист	Листов	
Н. Контр	Гавриленко О.В.								
Чтв.	Дергачев К.Ю.								

МАТЕМАТИЧНА І МАШИННА МОДЕЛІ

$$T_{em1} = I \cdot R \cdot C_e \cdot C_m \cdot \Phi^2; [2]$$

$$T_{em2} = I \cdot R \cdot C_e \cdot C_m \cdot \Phi^2; [2]$$

$$\frac{T_{em1}}{I_1} = \frac{T_{em2}}{I_2}$$

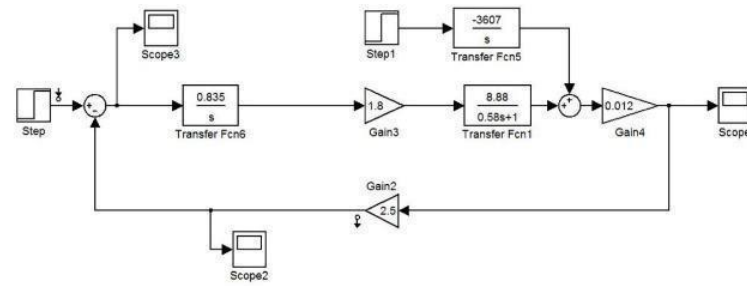
$$I_2 = I_1 + I_H;$$

$$T_{эм2} = \frac{T_{эм1} \cdot (I_1 + I_H)}{I_1};$$

$$T_{эм2} = \frac{T_{эм1} \cdot \left(\frac{r_{я}^2 \cdot m_{я}}{2} + \frac{(R_{к1}^2 + R_{к2}^2) \cdot m_{к}}{2} + m_{мл} \cdot i^2 \right)}{\left(\frac{r_{я}^2 \cdot m_{я}}{2} + \frac{(R_{к1}^2 + R_{к2}^2) \cdot m_{к}}{2} \right)};$$

$$T_{эм2} = \frac{0.36 \cdot \left(\frac{0.01^2 \cdot 0.04}{2} + \frac{(0.033^2 + 0.022^2) \cdot 0.05}{2} + 0.7 \cdot 0.012^2 \right)}{\left(\frac{0.01^2 \cdot 0.04}{2} + \frac{(0.033^2 + 0.022^2) \cdot 0.05}{2} \right)} = 0$$

Математична модель



Машинна модель

Инд. № лист.	Подп. і дата
Взам. инб. №	Инд. № лист.
Подп. і дата	Подп. і дата

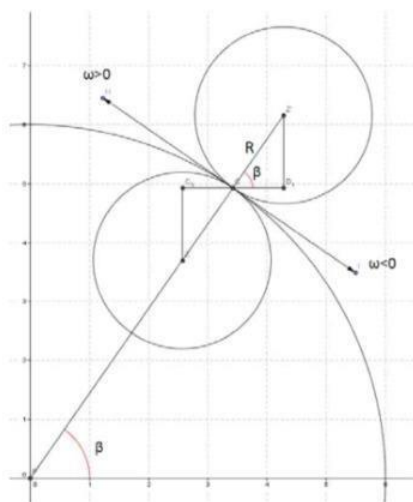
XAI.301.362.230.151.183018 ПЗ			
Изм/Лист	№ докум	Подп.	Дата
Разраб.	Ринченко В.О.		
Проб.	Гавриленко О.В.		
Т. Контр.			
Н. Контр.	Гавриленко О.В.		
Чтв.	Дергачев К.Ю.		
Математична та машина модель			
Лист	Масса	Масштаб	

XAI.301.362.230.151.183018 ПЗ

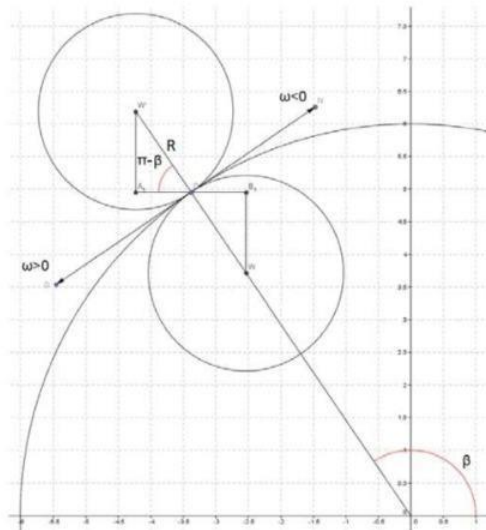
5

ГРАФІКИ ПЕРЕХОДУ НА НОВЕ КОЛО

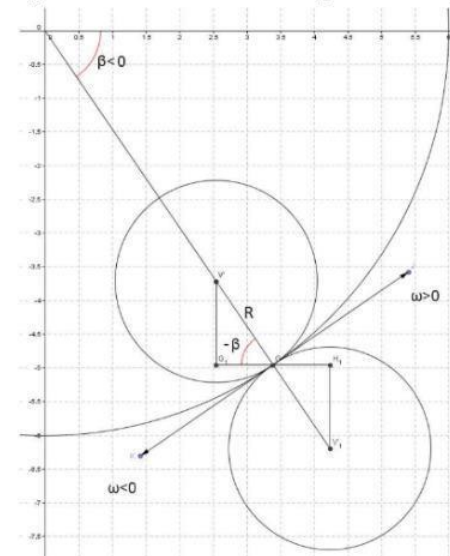
Перехід на нове коло при $0 \leq \beta \leq \pi/2$



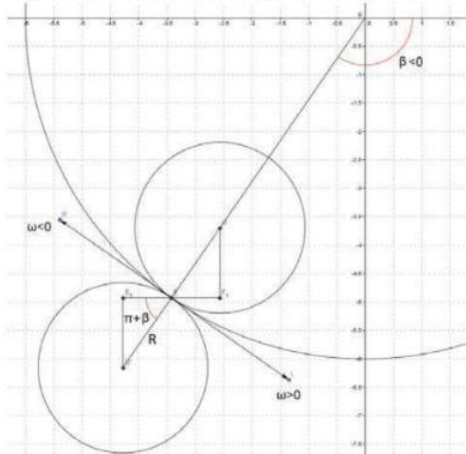
Перехід на нове коло при $\pi/2 \leq \beta \leq \pi$



Перехід на нове коло при $-\pi/2 \leq \beta \leq 0$



Перехід на нове коло при $-\pi < \beta \leq -\pi/2$



Інв. № побл. Подп. і дата
 Взам. інв. № / Інв. № дубл. Подп. і дата
 Взам. інв. № / Інв. № дубл. Подп. і дата

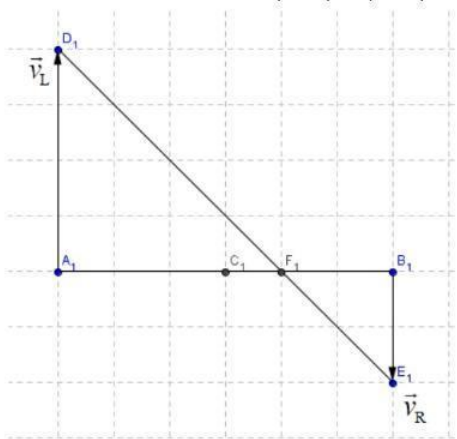
XAI.301.362.230.151.183018 ПЗ				
Изм.	Лист	№ докум	Подп.	Дата
Разраб.	Ринченко В.О.			
Проб.	Гавриленко О.В.			
Т. Контр.				
Н. Контр.	Гавриленко О.В.			
Утв.	Дергачев К.Ю.			
Графіки переходу на нове коло			Лист	Листов

XAI.301.362.230.151.183018 ПЗ

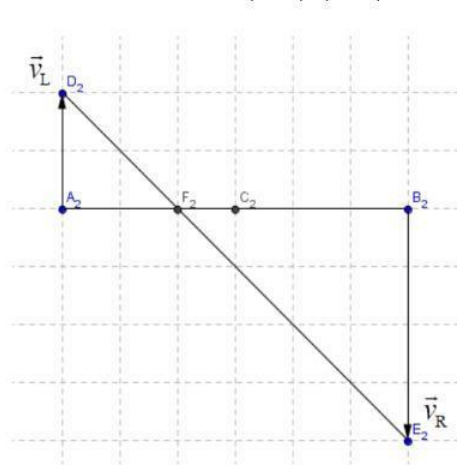
6

ГРАФІКИ ПОЛОЖЕННЯ ЦЕНТРУ КОЛА ПРИ РІЗНИХ УМОВАХ

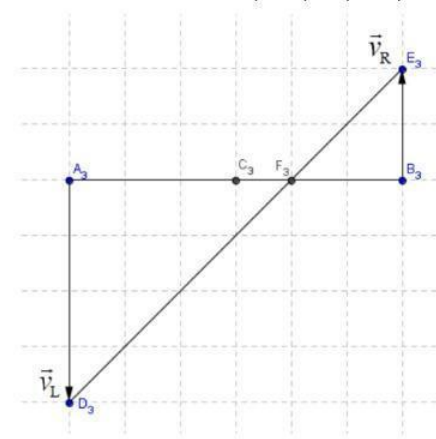
$\omega_L > 0, \omega_R < 0, |\omega_L| > |\omega_R|$



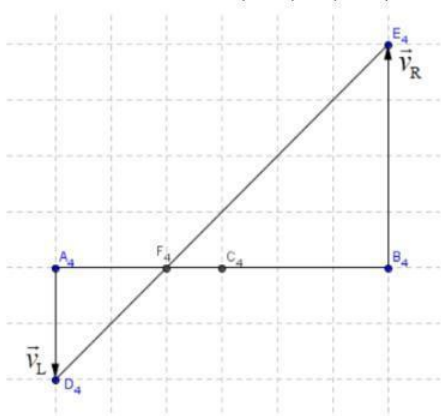
$\omega_L > 0, \omega_R < 0, |\omega_L| < |\omega_R|$



$\omega_L < 0, \omega_R > 0, |\omega_L| > |\omega_R|$



$\omega_L < 0, \omega_R > 0, |\omega_L| < |\omega_R|$



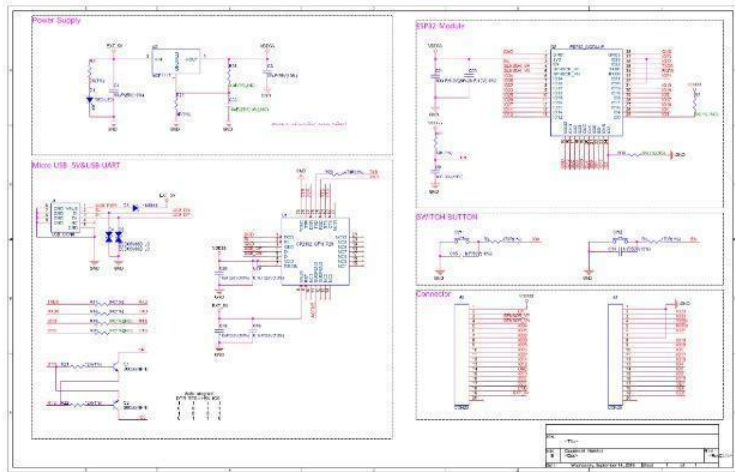
Інв. № лист. | Подп. у дата | Взам. инв. № | Инв. № дубл. | Подп. у дата

XAI.301.362.230.151.183018 ПЗ				
Изм/Лист	№ докум	Подп.	Дата	Графіки положення кола при різних умовах
Разраб.	Ришченко В.О.			
Проб.	Гавриленко О.В.			
Т. Контр.				
Н. Контр.	Гавриленко О.В.			Лист
Утв.	Дергачев К.Ю.			Листов

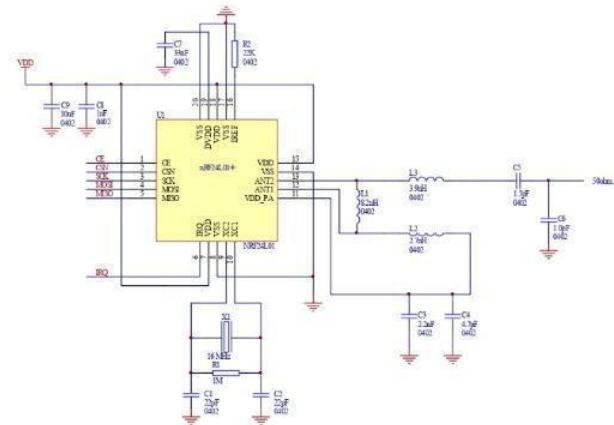
XAI.301.362.230.151.183018 ПЗ

ПРИНЦИПОВІ СХЕМИ КОМПОНЕНТІВ

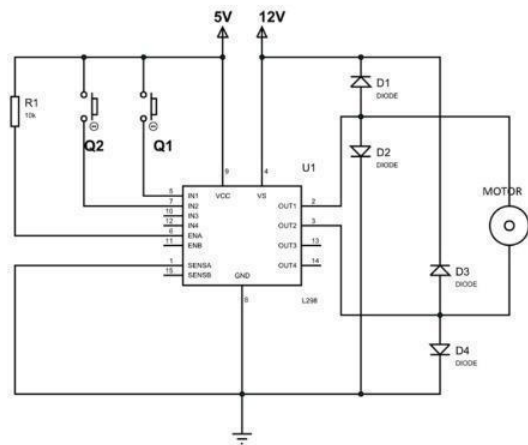
ESP32



NRF24L01



L298N

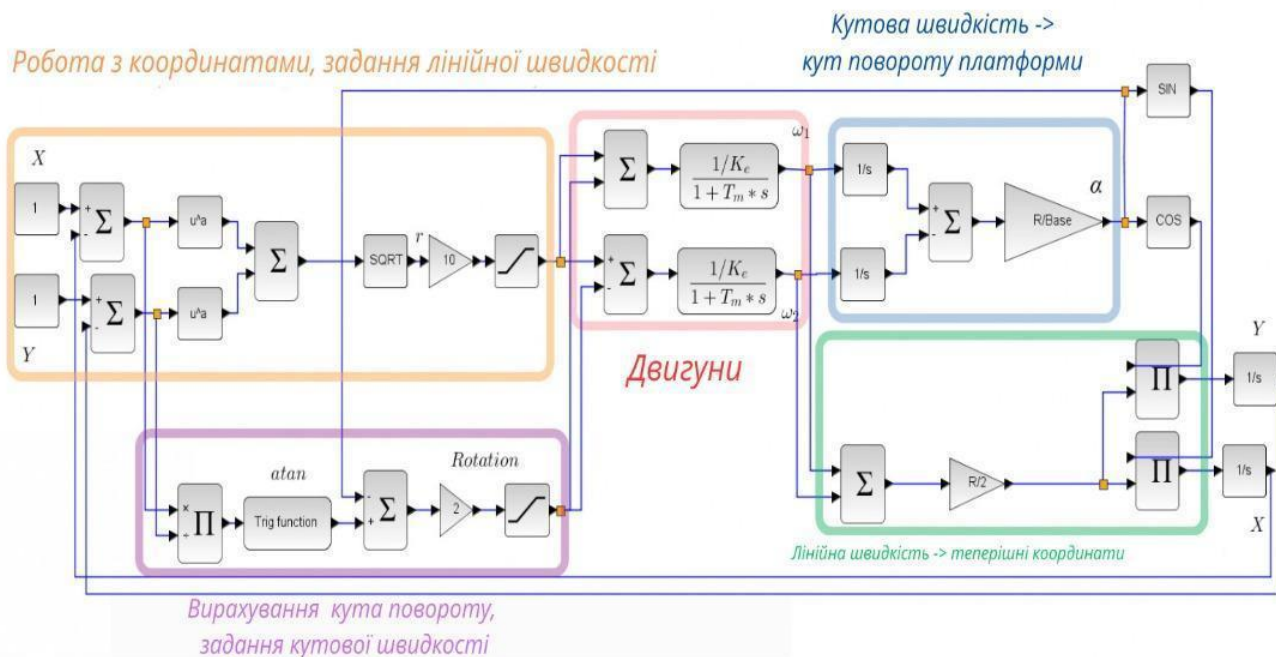


Взам. инв. № /Инв. № дубл. /Полн. и дата /Инд. № подл. /Полн. и дата

				XAI.301.362.230.151.183018 ПЗ				
Изм.	Лист	№ докум.	Подп.	Дата	Принципові електричні схеми компонентів	Лист	Масса	Масштаб
Разраб.		Ринченко В.О.				Лист	Листов	
Проб.		Гавриленко О.В.						
Т. Контр.								
Н. Контр.		Гавриленко О.В.						
Утв.		Дергачев К.Ю.						

XAI.301.362.230.151.183018 ПЗ

МОДЕЛЬ ДИНАМІКИ РУХУ ПЛАТФОРМИ ДО ТОЧКИ



Взам. инб. № /Инб. № дубл. /Подп. и дата
 Инб. № подл. /Подп. и дата

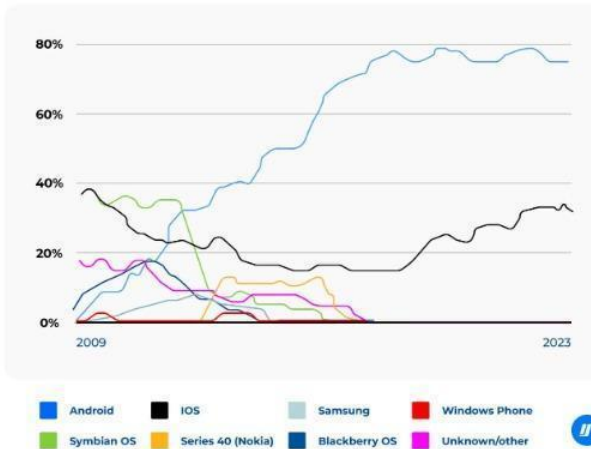
XAI.301.362.230.151.183018 ПЗ				
Изм/Лист	№ докум	Подп.	Дата	Модель динамики руху платформи до точки
Разраб.	Ринченко В.О.			Лит. / Масса / Масштаб
Проб.	Гавриленко О.В.			Лист / Листов
Т. Контр.				
Н. Контр.	Гавриленко О.В.			
Утв.	Дергачев К.Ю.			

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

Як основну платформу для розробки додатку був обраний Андроїд. Так як основну частку ринку мобілих телефонів та інших мобільних пристроїв займає саме він.

Android's global market share among other mobile operating systems

(2009-2023)



Код додавання точок

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import org.osmdroid.api.IMapController
import org.osmdroid.config.Configuration
import org.osmdroid.tileprovider.tilesource.TileSourceFactory
import org.osmdroid.views.MapView
import org.osmdroid.views.overlay.Marker
import org.osmdroid.views.overlay.Polyline
import java.util.*

class MapActivity : AppCompatActivity() {

    private lateinit var mapView: MapView
    private lateinit var mapController: IMapController
    private lateinit var currentLocationMarker: Marker
    private lateinit var destinationMarker: Marker

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_map)

        Configuration.getInstance().load(applicationContext,
            getPreferences(MODE_PRIVATE))

        mapView = findViewById(R.id.mapView)
        mapView.setTileSource(TileSourceFactory.MAPNIK)

        mapController = mapView.controller
        mapController.setZoom(15.0)

        mapController.setCenter(52.5200, 13.4050)

        currentLocationMarker = Marker(mapView)
        currentLocationMarker.position = currentGeoPoint(52.5200, 13.4050)
        mapView.overlays.add(currentLocationMarker)

        destinationMarker = Marker(mapView)
        destinationMarker.position = currentGeoPoint(52.5300, 13.4150)
        mapView.overlays.add(destinationMarker)

        drawRoute(currentLocationMarker.position, destinationMarker.position)

        private fun currentGeoPoint(latitude: Double, longitude: Double):
            org.osmdroid.util.GeoPoint {
                return org.osmdroid.util.GeoPoint(latitude, longitude)
            }

        private fun drawRoute(startPoint: org.osmdroid.util.GeoPoint, endPoint:
            org.osmdroid.util.GeoPoint) {
            val waypoints = ArrayList<org.osmdroid.util.GeoPoint>()

            waypoints.add(startPoint)
            waypoints.add(endPoint)

            val polyline = Polyline()
            polyline.setPoints(waypoints)
            mapView.overlays.add(polyline)
        }
    }
}
```

Інв. № докл. Підп. і дата
 Взам. інв. № Інв. № докл. Підп. і дата

					XAИ.301.362.23O.151.183018 ПЗ				
Изм/Лист	№ докum	Підп.	Дата	<i>Розробка мобільного додатку</i>			Лист	Масса	Масштаб
Разраб.	Ринченко В.О.								
Проб.	Гавриленко О.В.								
Т. Контр.							Лист	Листов	
Н. Контр.	Гавриленко О.В.								
Учб.	Дергачев К.Ю.								

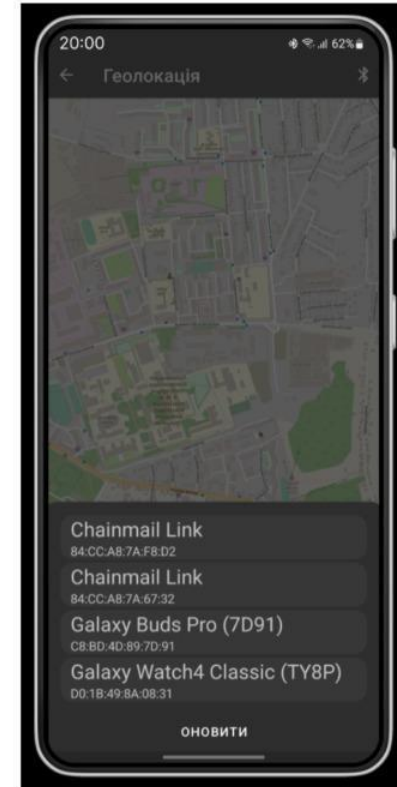
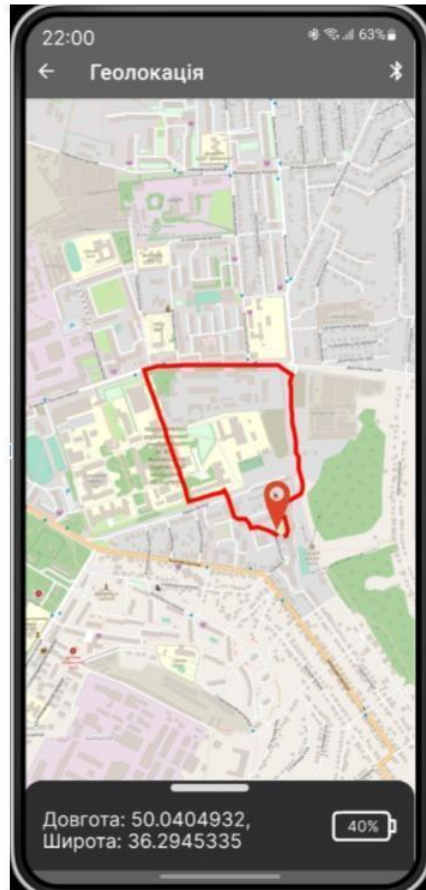
XAI.301.362.230.151.183018 ПЗ

10

ЗОВНІШНІЙ ВИГЛЯД ДОДАТКУ

Вікно підключення до платформи

Загальний вигляд додатку



Взам. инв. №	Инв. № дубл.	Подп. и дата
Инв. № подл.	Подп. и дата	

					XAI.301.362.230.151.183018 ПЗ			
Изм.	Лист	№ докум	Подп.	Дата	Математичний опис об'єкту автоматичного управління	Лист	Масштаб	Масштаб
Разраб.		Ринченко В.О.						
Проб.		Гавриленко О.В.						
Т. Контр.						Лист	Листов	
Н. Контр.		Гавриленко О.В.						
Утв.		Дергачев К.Ю.						

ХАІ.301.362.230.151.183018 ПЗ

11

ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Амортизація основних фондів

№	Обладнання	Первісна вартість, грн	Річна норма амортизації, %	Ефективний фонд часу роботи обладнання, л/рік	Час роботи обладнання для виконання роботи, год	Сума, грн
1	Комп'ютер	25000	20	2920	160	273,97
2	Програмне забезпечення	4000	15	2920	160	32,87
3	ПІДСУМОК					416,42

Загальні витрати на виконання роботи

№	Статті витрат	Сума, грн
1	Матеріальні витрати	29000
2	Витрати на оплату праці	7000
3	Витрати на електроенергію	600
4	Соціальний податок	693
5	Амортизація основних фондів	416,42
6	ПІДСУМОК	37709,42

Витрати на оплату праці

№	Категорія робітника	Кваліфікація	Трудомісткість розробки ПП	Ставка за годину, грн	Σ, грн
1	Науковий керівник	Керівник проекту	10	100	1000
2	Розробка документації	Керівник проекту	10	100	1000
3	Проектувальник	Розробник	100	50	5000
4	ЗАГАЛОМ				7000

Витрати на технічне оснащення

№	Найменування матеріального ресурсу	Одиниця виміру	Кількість використаного матеріалу	Ціна за одиницю, грн	Разом, грн
1	Комп'ютер	Шт.	1	25000	25000
2	Програмне забезпечення	Шт.	1	4000	4000
3	Підсумкові витрати				29000

ХАІ.301.362.230.151.183018 ПЗ

Изм/Лист	№ докум	Подп.	Дата	Лист	Масса	Масштаб
Разраб.	Ринченко В.О.					
Проб.	Гавриленко О.В.					
Т. Контр.				Лист	Листов	
Н. Контр.	Гавриленко О.В.					
Утв.	Дергачев К.Ю.					

Економічне обґрунтування розробки

Інв. № побл. Подп. і дата
 Взам. інв. № /Інв. № побл. Подп. і дата

ХАІ.301.362.230.151.183018 ПЗ

12

ВИСНОВКИ РОБОТИ

Метою даного проекту було розробити компактного автономного робота на основі мобільної повнопривідної платформи, здатного працювати в різних галузях, де потрібне використання мобільних транспортних роботів. Зазначений робот має широкий спектр застосувань, включаючи військове використання в якості розвідувального робота для збору інформації в екстремальних кліматичних та бойових умовах, а також виконання завдань, пов'язаних з розмінуванням і нанесенням ударів по стратегічно важливих цілях.

У процесі роботи було створено функціональні та структурні схеми системи, орієнтовані на одну умовну поздовжню вісь руху платформи. Також розроблено математичні моделі виконавчих механізмів та рухомої платформи на основі експериментальних даних, отриманих через перехідні та частотні характеристики.

Був проведений синтез системи за допомогою ЛАЧХ.

Система автоматичного керування, яка була розроблена та налаштована, відповідає вимогам, що стосуються функціональних показників якості.

Інв. № докл. Підп. і дата
Взам. инв. № Інв. № докл. Підп. і дата

						ХАІ.301.362.230.151.183018 ПЗ			
Изм/Лист	№ докум	Подп.	Дата	Висновки роботи			Лист	Масса	Масштаб
Разраб.	Ринченко В.О.								
Проб.	Гавриленко О.В.								
Т. Контр							Лист	Листов	
Н. Контр	Гавриленко О.В.								
Утв.	Дергачев К.Ю.								