

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 мобільним роботом з маніпулятором
та дослідження руху у середовищі з перешкодами

ХАІ.301.362.24О.151.00193073 ПЗ

Виконав: студент 2 курсу, групи 362

Галузь знань 15 «Автоматизація та
приладобудування»

Спеціальність

151 «Автоматизація та комп'ютерно-
інтегровані технології»

Освітня програма

«Інженерія мобільних додатків»

Гаркуша Владислав Михайлович

(прізвище та ініціали студента)

Керівник Гавриленко О. В.

(прізвище та ініціали)

Рецензент Ковтун І.В.

(прізвище та ініціали)

Харків – 2024

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 (№301)

Рівень вищої освіти другий (магістерський)

Галузь знань 15 Автоматизація та приладобудування

(шифр і назва)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології

(шифр і назва)

Освітня програма Інженерія мобільних додатків

(назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
 К.Т.Н., доц.

Костянтин ДЕРГАЧОВ
 “ ” _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ ЗДОБУВАЧУ

Гаркуша Владислав Михайлович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка системи управління мобільним роботом з маніпулятором та дослідження руху у середовищі з перешкодами».

Керівник роботи Гавриленко Олена Володимирівна., к.т.н., доцент,

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

затверджені наказом вищого навчального закладу від 06.11.2023 року
№ 1968-уч

2. Строк подання студентом роботи: 08 січня 2024 року

3. Вихідні дані до роботи

Розробка системи управління мобільним роботом

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналітичний огляд літератури предметної області, аналіз шляхів і способів рішення завдання, опис предметної області, опис функціональної схеми системи управління, розробка програмного продукту, постановка задачі експериментального дослідження системи, опис функціональної схеми системи, розробка принципової схеми системи, розрахунок собівартості

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Презентація, 10 змістовних слайдів за суттю роботи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стан проблеми	доц..к.т.н Гавриленко О.В	12.09.23	08.01.24
Аналіз і синтез СУ	доц..к.т.н Гавриленко О.В	12.09.23	08.01.24
Конструкт. Частина	доц..к.т.н Гавриленко О.В	12.09.23	08.01.24
Дослідницька частина	доц..к.т.н Гавриленко О.В	12.09.23	08.01.24
Експеримент. Частина	доц..к.т.н Гавриленко О.В	12.09.23	08.01.24
Економічне обґрунтування	доц..к.т.н Гавриленко О.В	12.09.23	08.01.24

Нормоконтроль _____ Олена ГАВРИЛЕНКО 08.01.2024 р.
 (підпис) (ініціали та прізвище)

7.Дата видачі завдання _____ 12.09.2023 _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Початок переддипломної практики	01.09.2023	
2.	Формулювання теми роботи. Розробка технічного завдання	12.09.2023	
3.	Математичний опис системи управління. Аналіз і синтез системи управління. Проведення експериментальних досліджень	20.10.2023	Залік з переддипломної практики
4.	Конструкторська частина роботи. Дослідницька частина роботи. Експериментально-практична частина. Економічне обґрунтування розробки. Розробка питань охорони праці і безпеки в надзвичайних ситуаціях	15.12.2023	
5.	Оформлення розрахунково-пояснювальної записки і графічного матеріалу	08.01.2024	
6.	Попередній захист роботи. Рецензування роботи	12.01.2024	
7.	Захист роботи	16.01.2024	

Здобувач _____ Владислав ГАРКУША

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_
Гаркуша Владислав Михайлович

- 1 Тема роботи: Розробка системи управління мобільним роботом з маніпулятором та дослідження руху у середовищі з перешкодами
затверджена наказом по університету від 06.11.2023 р. № 1968-уч .
- 2 Строк здачі студентом закінченої роботи « 08 » _____ січня _____ 2024 р.
- 3 Область застосування розробки: системи управління мобільними роботами

- 4 Початкові дані для розроблювальної системи
 - 4.1 Призначення і мета створення системи: Вихідна математична модель об'єкта управління
 - 4.2 Загальні відомості дослідження руху у середовищі з перешкодами
- 5 Технічні вимоги до каналів системи управління
 - 5.1 Питання, що підлягають розробці:
 - 5.1.1 У розрахунковій частині: оцінка стану проблеми, отримання моделі об'єкта управління;
 - 5.1.2 В експериментальній частині: провести реалізація моделі в дослідженні систем управління ;
 - 5.1.3 У конструкторській частині: створити схему управляючого пристрою на базі датчиків і плат ;
 - 5.2 Режим роботи системи (безперервний, циклічний, одноразової дії):
циклічний
 - 5.3 Показники якості системи управління: Показники якості роботи системи: час перехідного процесу не більше 60 секунд, перегулювання не більше 20%;
 - 5.4 Вимоги до приладового складу системи: вибрати і обґрунтувати в процесі проектування;

5.5 Вимоги до взаємозамінності блоків: взаємозамінність блоків, в разі відмови, аналогічними за технічними характеристиками;

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.) діапазон температур 15-25 градусів Цельсія, відносна вологість до 100%;

6.2 Механічні вимоги (вібрація, тряска, можливі перекося, удари, нахили і т.ін.): вібрації: амплітуда - 0,01 мм, частота - 0 ... 100 Гц;

6.3 Наявність перешкод (електричні наведення радіоперешкоди, магнітні впливи): не задані;

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

7 Додаткові функції, реалізовані системою (сигналізація про несправності, реєстрація необхідної інформації, самоконтроль самої системи і т.ін.): не передбачається;

8 Обсяг виконуваних розроблювачем робіт

8.1 Етапи проведення роботи:

1. Аналіз існуючих мобільних платформ маніпуляційних роботів;
- 2) Аналіз технічного завдання. Загальна характеристика проблеми. Огляд науково-технічної літератури і патентів;
- 3) Математичний опис системи управління мобільного робота;
- 4) Проектування додатка. Вибір засобів реалізації систем управління мобільного робота. Реалізація в середовищі розробки;
- 5) Експериментально-практична частина розробки системи управління мобільного робота;
- 6) Оформлення розрахунково-пояснювальної записки і листів графічного матеріалу

8.2 Обсяг розробки по кожному етапу:

1. Стан проблеми розробки та аналіз існуючих мобільних платформ маніпуляційних роботів –14 стор.;
2. Аналіз і синтез системи управління мобільним роботом із маніпулятором – 14 стор.;
3. Конструкторська частина та опис використаних компонентів - 20 стор.;

4. Дослідницька частина – 15 стор.;
5. Експериментальна частина – 7 стор.;
6. Економічне обґрунтування розробки - 6 стор.;

9 Параметри устаткування системи:

9.1. Граничні розміри: визначаються в процесі проектування;

9.2. Маса: визначається в процесі проектування;

9.3. Вимоги по конструктивному виконанню і розміщенню: визначаються в процесі проектування;

9.4. Інші вимоги: не передбачається;

10 Вимоги безпеки: визначаються в процесі проектування

11 Дослідницька частина: розрахувати і розробити дослідження для роботи мобільного робота

12 Експериментально-практична частина: розробити мобільний додаток для роботи з бездротовим інтерфейсом мобільного робота.

13. Економічна частина

13.1 Розробити (розрахувати, одержати): основне обладнання та програмне забезпечення, заробітну плату, амортизацію обладнання; оренду приміщення

13.2 Умови і вимоги: припустимі для виконання поставлених завдань

13.3 Очікуваний результат: собівартість виробу в гривнях, з урахуванням всіх витрат;

14 Перелік графічних матеріалів із зазначенням форматів:

1-й плакат (формат А1)- стан проблеми розробки та аналіз існуючих мобільних платформ

2-й плакат (формат А1)- кінематична модель мобільного робота

3-й плакат (формат А1)- принципи управління сервоприводами та постійним струмом

4-й плакат (формат А1)- характеристика двигуна та його комплектуючих запчастин

5-й плакат (формат А1)- моделювання динаміки комплектуючих плат та датчиків

6-й плакат (формат А1)- Схема живлення елементів мобільного робота

7-й плакат (формат А1)- Моделювання динаміки системи управління при заданих початкових умов

8-й плакат (формат А1)- Конструкторська частина

9-й плакат (формат А1)- Дослідницька частина

10-й плакат (формат А1)-Експериментальна частина. Економічна частина

Керівник роботи <u>Гавриленко О.В</u> (П.І.Б.)	Прийняв до виконання <u>Гаркуша В.М</u> (П.І.Б. студента)
_____ « » _____ 2023 р.	_____ _____ « » _____ 2023 р.

Погоджено з питань:

конструкції <u>Гавриленко О.В</u> (П.І.Б.)	дослідницької частини <u>Гавриленко О.В</u> (П.І.Б.)
_____ « » _____ 2023 р.	_____ « » _____ 2023 р.
економіки <u>Гавриленко О.В</u> (П.І.Б.)	
_____ « » _____ 2023 р.	

СПИСОК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

КМП - Колісні Мобільні Платформи

PLA - пластик- Полілактид

СУ - Система Управління

ШІМ – Широтно-імпульсна Модуляція

TCP - Transfer Control Protocol

IP - Internet Protocol

ОЗК - Обернена Задача Кінематики

РЕФЕРАТ

Випускна кваліфікаційна робота на тему "Розробка системи управління мобільним роботом з маніпулятором та дослідження руху у середовищі з перешкодами " містить 93 сторінок текстового документа, 2 додатків, 13 використаних джерел.

ІНТЕЛЕКТ, МОБІЛЬНИЙ РОБОТ, МАНІПУЛЯТОР, МОБІЛЬНИЙ ДОДАТОК, РОЗПІЗНАВАННЯ ОБРАЗІВ, БЕЗДРОТОВИЙ ЗВ'ЯЗОК, СИСТЕМА УПРАВЛІННЯ, МЕТОДИ ПОШУКУ ШЛЯХІВ, ЗАВДАННЯ КІНЕМАТИКИ.

Цілі роботи:

- провести аналіз існуючих мобільних платформ та розробити конструкцію, зібрати мобільний робот;
- розробити систему інтелектуального управління;
- розробити алгоритми планування траєкторії у середовищі з перешкодами;
- розробити систему управління робота в середовищі з перешкодами.

У ході дипломного проектування було проведено аналіз існуючих мобільних платформ, створено мобільний маніпуляційний робот, розроблено програму формування керуючих команд, отримавши зображення з камери, встановленої зверху, реалізовано мобільний додаток для управління роботом у середовищі з перешкодами. Ефективність вибраних методів була підтверджена експериментами.

ЗМІСТ

ВСТУП	13
1. СТАН ПРОБЛЕМИ РОЗРОБКИ ТА АНАЛІЗ ІСНУЮЧИХ МОБІЛЬНИХ ПЛАТФОРМ МАНІПУЛЯЦІЙНИХ РОБОТІВ	14
1.1 Класифікація кінематичних моделей колісних мобільних платформ.....	14
1.2 Огляд літератури щодо мобільних маніпуляційних роботів.....	21
1.3 Постановка задач дипломної роботи	25
1.3 Висновки за розділом.....	25
2. АНАЛІЗ І СІНТЕЗ СИСТЕМИ УПРАВЛІННЯ МОБІЛЬНОГО МАНІПУЛЯЦІЙНОГО РОБОТА.....	26
2.1 Кінематична модель мобільного робота.....	26
2.2 Маніпулятор.....	31
2.2.1 Кінематична схема маніпулятора.....	31
2.2.2 Визначення діапазонів кутів поворотів ланок.....	32
2.2.3 Проектування схвату маніпулятора.....	33
2.3 Принципи управління сервоприводами.....	34
2.4 Принципи управління моторами постійного струму.....	35
2.5 Протокол TCP/IP для обміну даними з wi-fi модулем.....	36
2.6 Інтерфейс I ² C для обміну даними гіроскопа і мікроконтролера.....	37
2.7 Висновки за розділом.....	38
3. КОНСТРУКТОРСЬКА ЧАСТИНА ТА ОПИС ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ.....	39
3.1 Привідна система коліс	39
3.2 Сервопривод MD996R	40
3.3 Сервопривод MG90S.....	43
3.4 Реверсивний драйвер двигунів L293D	44
3.5 Arduino Mega 2560	45

	11
3.6 Лінійний стабілізатор напруги L7805.....	47
3.7 Блок живлення	48
3.8 Оптичний енкодер FC-03	48
3.9 Wi-fi модель NodeMCU LUA R2	51
3.10 Акселерометр і гіроскоп MPU6050 54.....	52
3.11 Система підключення живлення для всіх елементів	53
3.12 Схема управління кожним елементом	55
3.13 Висновки за розділом.....	57
4.ДОСЛІДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ МОБІЛЬНИМ РОБОТОМ...56	
4.1 Постановка завдання.....	56
4.2 Отримання знімка з камери.....	57
4.2.1 Виділення основних кольорів.....	63
4.2.2 Зменшення обсягу матриці і пошук початкової і кінцевої координати....	64
4.2.3 Апроксимація перешкод прямокутниками.....	65
4.2.4 Хвильовий алгоритм Лі.....	66
4.3 Керуючі команди для мобільного робота.....	67
4.4 Рішення зворотної задачі кінематики маніпулятора.....	68
4.5 Висновки за розділом.....	69
5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	70
5.1 Опис структури програмного забезпечення.....	70
5.2 Функція отримання зображення з камери.....	70
5.3 Функція підрахунку кількості кроків і напрямки мобільного робота.....	72
5.4 Функція формування керуючих команд і передача по бездротовому зв'язку.....	73
5.5 Функція пошуку об'єкта для маніпулювання і рішення оборотної задачі кінематики маніпулятора.....	75
5.6 Висновки за розділом.....	77
6 ЕКОНОМІЧНА ОБҐРУНТУВАННЯ	78
6.1 Розрахунок витрат на обладнання та програмне забезпечення.....	78

6.2 Розрахунок фонду заробітної плати розробникам.....	79
6.3 Витрати на електроенергію.....	80
6.4 Витрати на амортизацію обладнання.....	80
6.5 Розрахунки на оренду приміщення.....	81
6.6 Кошторис витрат.....	81
6.7 Висновки за розділом.....	83
ВИСНОВОК	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	85
ДОДАТОК А.....	87
ДОДАТОК Б.....	91

ВСТУП

Протягом багатьох років людина все частіше прибігає до допомоги роботів в різних сферах діяльності. Сьогодні польоти виконують досить складну роботу: їх застосовують на виробництві, у військовій справі, в медицині і просто для розрахунків. Наприклад, у середовищі, де не відоме положення об'єктів (такі осередки називаються не термінованими) застосовують інтелектуальні мобільні роботи. Такі роботи, які включають в себе чутливі елементи, механізми, комп'ютери, і які мають елементи штучного інтелекту представляють дуже зручний об'єкт для постановки, вивчення і знаходження рішень суміжних.

На сьогоднішній день здійснює багато видів мобільних роботів, які значно спрощують життя людини. Сучасний світ вже важко уявити без щоденної допомоги роботів. Таким чином вивчення роботів корисно у всіх відношеннях: по-перше, що зрозуміти принцип роботи робота, по-друге, щоб самостійно спроектувати і створити мобільного робота.

Отже, актуальність обраної теми обумовлена необхідністю вивчення класифікацій мобільних роботів і, згодом, створенням свого інтелектуального мобільного маніпуляційного робота, який використовується в якості досвідченої моделі. Вивчивши його властивості і застосувавши алгоритми управління, можна керувати більш складним і потужним пристроєм.

В якості теоретичної бази дослідження були використані публікації, присвячені технологіям роботи маніпуляційних роботів.

Практична значимість даної роботи полягає в тому, що продукт дослідження може бути використаний в небезпечному для людини середовищі, а також в якості наочного матеріалу для навчання студентів і школярів.

1 СТАН ПРОБЛЕМИ РОЗРОБКИ ТА АНАЛІЗ ІСНУЮЧИХ МОБІЛЬНИХ ПЛАТФОРМ МАНІПУЛЯЦІЙНИХ РОБОТІВ

1.1 Класифікація кінематичних моделей колісних мобільних платформ

Здатність мобільних платформ автоматично і цілеспрямовано переміщатися в досить складних умовах нерозривно пов'язана зі здатністю швидко справлятися із задачами, для рішення яких треба винятково високий рівень інтелекту. Більше того, при сучасному рівні розвитку техніки, будівництво пристроїв, за своїми чисто механічними можливостями у порівнянні з ногами людини або тварини, викликає значні труднощі [1]. Всі механічні пристрої, призначені для пересування в просторі - танки, трактори, автомобілі і інші - виконують свої функції під безпосереднім керуванням людини. Не мають принципових відмінностей від перерахованих пристроїв і різного виду радіокеровані апарати – людина хоч і знаходиться на відстані, але, тим не менш, контролює кожен "крок" дистанційно керованих машин. На перший погляд до автономних мобільних пристроїв можна було б віднести нові електровози з системами автоматичного управління. Однак не слід забувати про те, що в даному випадку автономність досягається за рахунок дуже жорстких обмежень, що накладаються на умови пересування: поїзд рухається тільки по коліях, які, до всього іншого, потрібно обнести надійною огорожею, так як автоматика, на відміну від машиніста, не може подати сигнал про небезпеку пішоходу або тварині, які забарилися на шляху.

Колісні мобільні платформи (далі КМП) становлять клас механічних систем, що характеризуються кінематичними зв'язками, які не є інтегрованими і, отже, не можуть бути виключені з рівнянь руху моделі. У результаті стандартні алгоритми планування та управління рухом, розвинені для роботів-маніпуляторів без зв'язків, не застосовуються до таких систем. Ця обставина була відзначена нещодавно у великій літературі, присвяченій побудові планування та управління рухом КМП, особливо специфічних спрощених кінематичних моделей "робопоїзда" та "робокара". Однак,

комерційні КМП, наявні у продажу, мають значно складнішу конструкцію, ніж найпростіші моделі, які, зазвичай, розглядаються в науковій літературі. Наприклад, для триколісних або чотириколісних комерційних мобільних роботів, з усіма поворотними та приводними колесами, суттєвим є навіть питання, що передує дослідженням з керованості, – питання про побудову математичної моделі робота.

Робот з омніколесами, розташованими у вершинах корпусу. Омніколісний робот з шведськими колесами. Робот, що розглядається (рис. 1.1) має три омніколеса, розташованих у вершинах корпусу, що має форму рівностороннього трикутника.

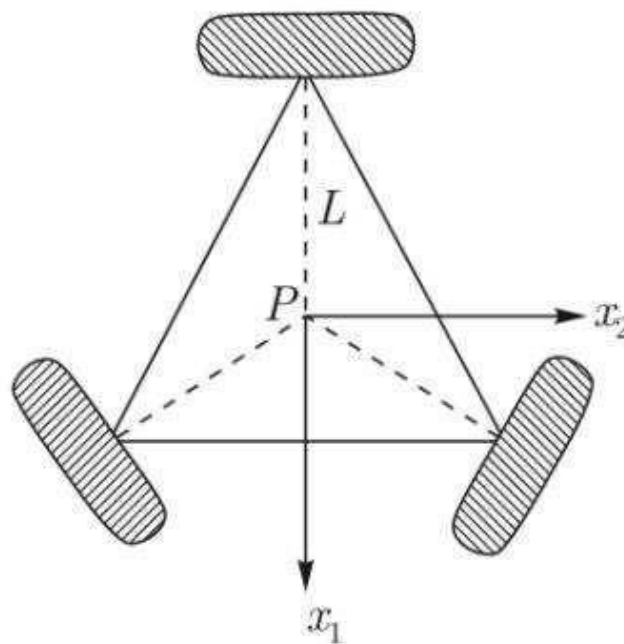


Рисунок 1.1 – Робот з омніколесами, розташованими у вершинах корпусу

Зв'язки, накладені на систему, мають вигляд:

$$J_1(\beta_c, \beta_{oc}) \cdot R(\theta) \cdot \dot{\epsilon} + J_2 \cdot \dot{\phi} = 0, \quad (1.1)$$

де s – позиційні координати, необхідні для опису положення корпусу робота в площині руху;

β_c – кути орієнтації центрально орієнтованих коліс;

β_{oc} – кути орієнтації нецентрально орієнтованих коліс;

φ – обертальні координати для опису обертання коліс щодо відповідних горизонтальних осей обертання.

При цьому J_1 і J_2 дорівнюють:

$$J_1 = [J_{1cw}] = \begin{bmatrix} -\frac{\sqrt{3}}{2} & \frac{1}{2} & L \\ 0 & -1 & L \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & L \end{bmatrix}. \quad (1.2)$$

$$J_2 = \text{diag}(r). \quad (1.3)$$

Функція $\text{diag}(v)$ формує квадратну матрицю X із вектором v на головній діагоналі.

Омнімобільний робот із нецентрально орієнтованими колесами. Цей робот має три традиційні нецентрально орієнтовані колеса, як показано на рис.1.2.

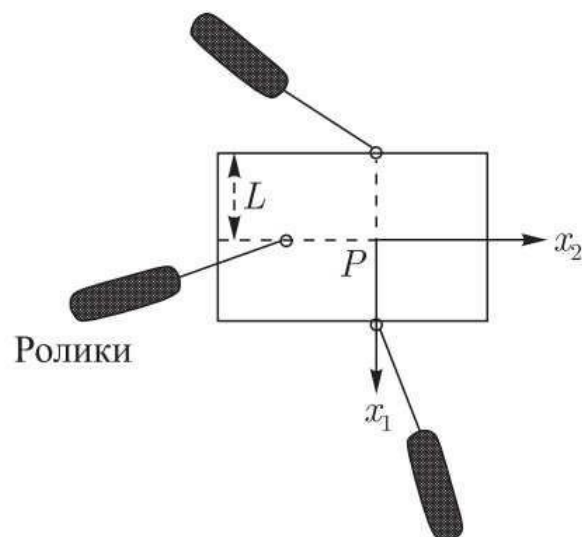


Рисунок 1.2 – Омнімобільний робот із нецентрально орієнтованими колесами

Зв'язки мають вигляд (1.1) та (1.4):

$$J_1 = [U_{1oc}(\beta_{oc})] = \begin{bmatrix} -\sin(\beta_{oc1}) & \cos(\beta_{oc1}) & L\cos(\beta_{oc1}) \\ \sin(\beta_{oc2}) & -\cos(\beta_{oc2}) & L\cos(\beta_{oc2}) \\ \sin(\beta_{oc3}) & \sin(\beta_{oc3}) & L\cos(\beta_{oc3}) \end{bmatrix}. \quad (1.5)$$

$$J_2 = \text{diag}(r). \quad (1.6)$$

$$C_1 = [C_{1oc}(\beta_{oc})] = \begin{bmatrix} \cos(\beta_{oc1}) & \sin(\beta_{oc1}) & d + L\cos(\beta_{oc1}) \\ -\cos(\beta_{oc2}) & -\sin(\beta_{oc2}) & d + L\cos(\beta_{oc2}) \\ \sin(\beta_{oc3}) & -\cos(\beta_{oc3}) & d + L\cos(\beta_{oc3}) \end{bmatrix}. \quad (1.7)$$

$$C_2 = |C_{2oc}| = \text{diag}(r). \quad (1.8)$$

Двоколісний робот із некерованим третім колесом. Цей робот має два традиційно фіксованих колеса, розташованих на одній осі, і одне нецентрально орієнтоване колесо (рис.1.3).



Рисунок 1.3 – Двоколісний робот із некерованим третім колесом

Зв'язки мають вигляд (1.1) та (1.4):

$$J_1 = \begin{bmatrix} J_{2f} \\ J_{2oc} \end{bmatrix} = \begin{bmatrix} 0 & 1 & L \\ 0 & -1 & L \\ \cos(\beta_{oc3}) & \sin(\beta_{oc3}) & L\cos(\beta_{oc3}) \end{bmatrix}. \quad (1.9)$$

$$J_2 = \text{diag}(r). \quad (1.10)$$

$$C_2 = \begin{bmatrix} C_{2f} \\ C_{2oc} \end{bmatrix} = \begin{bmatrix} 0 & 1 & L \\ 0 & -1 & L \\ \sin(\beta_{oc3}) & -\cos(\beta_{oc3}) & L\cos(\beta_{oc3}) \end{bmatrix}. \quad (1.11)$$

$$C_2 = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}. \quad (1.12)$$

Робот із двома некерованими колесами. Є робота з одним традиційно центрально орієнтованим колесом і двома нецентрально орієнтованими колесами (рис.1.4)

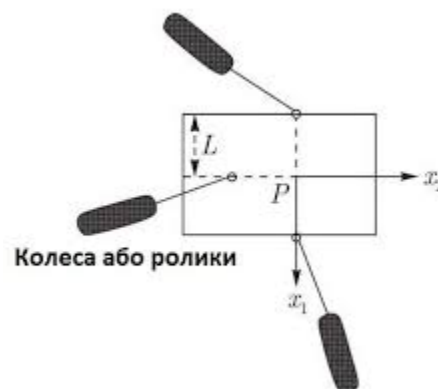


Рисунок 1.4 – Робот із двома некерованими колесами

Рівняння зв'язку мають вигляд (1.1) та (1.4), де:

$$J_1 = \begin{bmatrix} J_{1c}(\beta_{c1}) \\ J_{1oc}(\beta_{oc2}, \beta_{oc3}) \end{bmatrix} = \begin{bmatrix} -\sin(\beta_{c1}) & \cos(\beta_{c1}) & l\cos(\beta_{oc1}) \\ \sin(\beta_{oc2} + \frac{\pi}{4}) & -\cos(\beta_{oc2} + \frac{\pi}{4}) & l\cos(\beta_{oc2}) \\ -\sin(\beta_{oc3} - \frac{\pi}{4}) & \sin(\beta_{oc3} - \frac{\pi}{4}) & l\cos(\beta_{oc3}) \end{bmatrix}. \quad (1.13)$$

$$J_2 = \text{diag}(r). \quad (1.14)$$

$$C_1 = \begin{bmatrix} C_{1c}(\beta_{c1}) \\ C_{1oc}(\beta_{oc2}, \beta_{oc3}) \end{bmatrix} = \begin{bmatrix} \cos(\beta_{c1}) & \sin(\beta_{c1}) & l\sin(\beta_{oc1}) \\ -\cos(\beta_{oc2} + \frac{\pi}{4}) & -\sin(\beta_{oc2} + \frac{\pi}{4}) & d + l\sin(\beta_{oc2}) \\ \cos(\beta_{oc3} - \frac{\pi}{4}) & \sin(\beta_{oc3} - \frac{\pi}{4}) & d + l\cos(\beta_{oc3}) \end{bmatrix} \quad (1.15)$$

$$C_2 = \begin{bmatrix} 0 \\ \text{diag}(d) \end{bmatrix}. \quad (1.16)$$

Модель типу триколісного дитячого велосипеда. Дана модель є роботом із двома традиційно фіксованими колесами на одній осі та одним традиційно центрально орієнтованим колесом (рис. 1.5).



Рисунок 1.5 – Модель типу триколісного дитячого велосипеда

Рівняння зв'язку мають вигляд (1.1) та (1.4):

$$J_1 = \begin{bmatrix} J_{2f} \\ J_{2oc} \end{bmatrix} = \begin{bmatrix} 0 & 1 & L \\ 0 & -1 & L \\ \cos(\beta_{oc3}) & \sin(\beta_{oc3}) & L\cos(\beta_{oc3}) \end{bmatrix}. \quad (1.16)$$

$$J_2 = \text{diag}(r). \quad (1.17)$$

$$C_1 = \begin{bmatrix} C_{1f} \\ C_{1oc} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ \sin(\beta_{oc3}) & -\cos(\beta_{oc3}) & d + L\cos(\beta_{oc3}) \end{bmatrix}. \quad (1.18)$$

$$C_2 = 0. \quad (1.19)$$

Робот із чотирма провідними колесами.

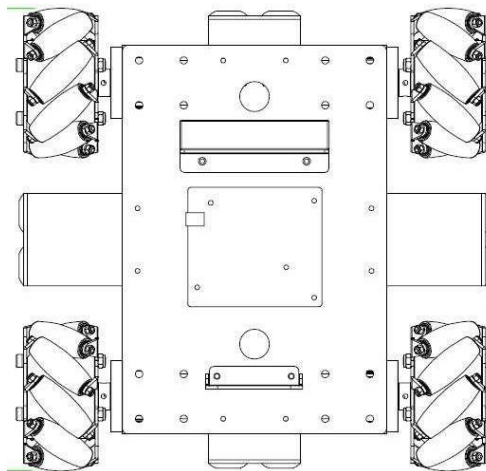


Рисунок 1.6 – Модель робот із чотирма провідними колесами

Зв'язки, накладені на систему:

$$\left. \begin{aligned} \dot{X} &= v1 \cdot \cos(f1) \\ \dot{Y} &= v2 \cdot \sin(f1) \\ f1 &= \omega \end{aligned} \right\}. \quad (1.20)$$

Співвідношення (1.20) є системою нелінійних диференціальних рівнянь 3-го порядку щодо фазового вектора з компонентами (X, Y, f_1) . Скаляри v_1 і v_2 , задані тим чи іншим чином (як функції часу або функції фазових координат), визначають рух платформи. Отже, співвідношення (1.20) можна розглядати як кінематичну модель мобільного робота.

Робот з гусеничними шасі.

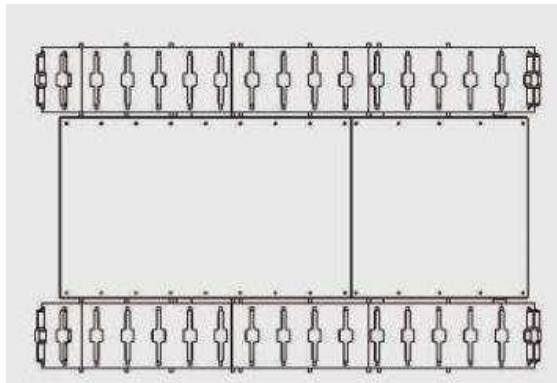


Рисунок 1.7 – Модель робота з гусеничними шасі

Важлива відмінна якість гусеничних мобільних роботів полягає в їхній маневреності [2]. Маючи незалежний привод для кожної з гусениць окремо, мобільний робот може легко змінювати напрямок власного руху.

1.2 Огляд літератури щодо мобільних маніпуляційних роботів

Робототехніка досягла свого найбільшого успіху на сьогоднішній день у світі промислового виробництва. Роботи-маніпулятори становлять 2 мільярди доларів у промисловості. Маніпулятор – це механізм для керування просторовим положенням знарядь і об'єктів праці, призначений для виконання завдань, що повторюються, таких як точкове зварювання і фарбування [3].

Проте, за всіх своїх успіхів ці комерційні роботи страждають від фундаментального недоліку: відсутність мобільності. Фіксований маніпулятор має обмежений діапазон руху.

На відміну від цього мобільний робот зможе переміщатися по всьому промислового підприємству.

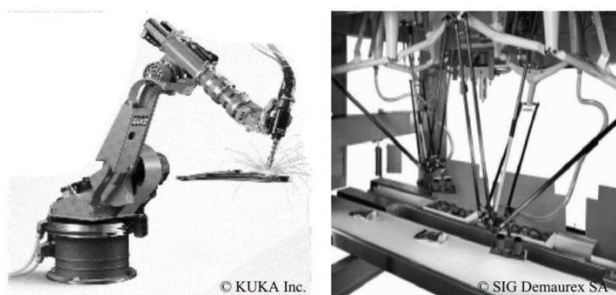


Рисунок 1.8 – Робот-маніпулятор Кука

Такі робототехнічні системи набули популярності в небезпечних та важкопрохідних середовищах [4]. Далі представлені перші та останні розробки мобільних роботів з маніпуляторами від провідних виробників усього світу.

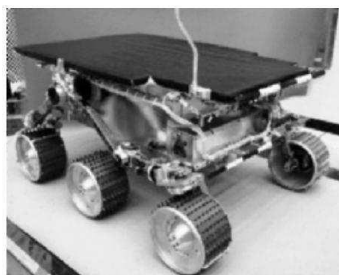


Рисунок 1.9 – Мобільний робот Sojourner

Мобільний робот Sojourner використовувався під час місії Pathfinder для вивчення Марса влітку 1997 року [5].



Рисунок 1.10 – Перший керований ходовий робот

Plustech розробив перший керований ходовий робот. Він призначений для переміщення деревини із лісу. Координація ніг автоматизована, але навігація, як і раніше, виконується оператором-людиною [7].



Рисунок 1.11 – Мобільний маніпулятор XL-MICO виробництва RobotnikAutomation

Застосування маніпулятора XL-MICO:

- дослідження;
- огляд;
- медицина;
- логістика;
- пульт дистанційного моніторингу;
- доступ до жорстких середовищ.



Рисунок 1.12 – Мобільні роботи Сервосила «Інженер»

Робот призначений для роботи на вулиці, поза приміщеннями. Робот повністю герметичний, здатний працювати під дощем, у снігу та в пилу. Залежно від встановлених корисних навантажень, робот здатний працювати при розширеному діапазоні температур. Мобільний робот здатний забиратися сходами в сходових прольотах багатоповерхових будинків [8].

Мобільний робот оснащений робототехнічним маніпулятором для захоплення об'єктів. Маніпулятор дозволяє використовувати робота для

роботи з потенційно небезпечними предметами, не наражаючи на ризик людське життя. На «пальцях» маніпулятора є місця кріплення додаткового інструменту, який може знадобитися в особливих ситуаціях. Це дозволяє адаптувати робота під конкретні ситуації та застосування [9].



Рисунок 1.13 – Мобільний маніпулятор RB-1

Пересувний маніпулятор, автономний та конфігурований робот, виробництва Robotnik Automation & Kinova Robotics, Spain & Canada на рис.1.14 [10].

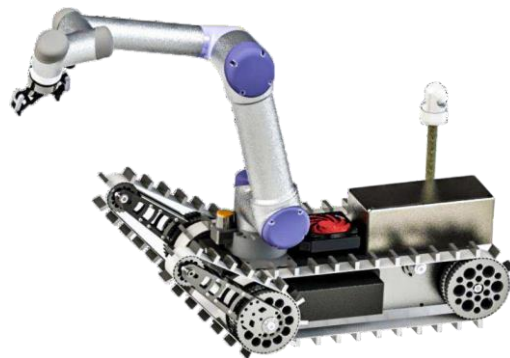


Рисунок 1.14 – Мобільний робот з маніпулятором виробництва Robotnik Automation & Barrett Technologies

Характеристики:

- вантажопідйомність платформи: 20 кг / важіль 5 кг;
- швидкість: 2,5 м / с;
- час роботи: 3,5 год;
- батареї: 48v LiPO;
- діапазон температур: - 10°C...+50°C;
- максимальний кут підйому: 80%.

1.3 Постановка задач дипломної роботи

Об'єкт дипломної роботи: чотириколісний мобільний робот з маніпулятором.

Предмет дипломної роботи: управління маніпуляційним роботом в середовищі з перешкодами.

Мета: розробка мобільного додатка для управління маніпуляційним роботом з інтелектуальним управлінням.

Завдання дипломного проєкту.

1. Провести аналіз сучасних мобільних платформ, розробити конструкцію і кінематичну схему мобільного робота, що включає в себе мобільну площу.

2. Підібрати комплектуючі і реалізувати в «залізі» мобільний робот.

3. Розробити алгоритми управління мобільним роботом (рухання вперед, назад; повороти; переміщення ланок маніпулятора; зв'язок по бездротовому зв'язку).

4. Опрацювати зображення з камери і знайти найкоротший шлях до об'єкта.

5. Створити керуючі команди і відправити їх по безпроводовому зв'язку на контроль мобільного робота.

6. Розв'язати зворотну задачу робототехніки.

7. Провести тест-драйви робота і визначити ефективність використання корисних методів.

Методологічною основою для дослідження є наукові праці відомих вчених-робототехніків.

1.4 Висновки за розділом

В розділі були розглянуті та проаналізовані стан проблеми розробки та аналіз існуючих мобільних платформ маніпуляційних роботів і сформульовано задачі, що вирішено у дипломній роботі.

2 АНАЛІЗ І СІНТЕЗ СИСТЕМИ УПРАВЛІННЯ МОБІЛЬНОГО МАНІПУЛЯЦІЙНОГО РОБОТА

2.1 Кінематична модель мобільного робота

Мобільний робот є чотириколісним візком з усіма провідними колесами, характеристики якого приведено на рис.2.1 – 2.2.

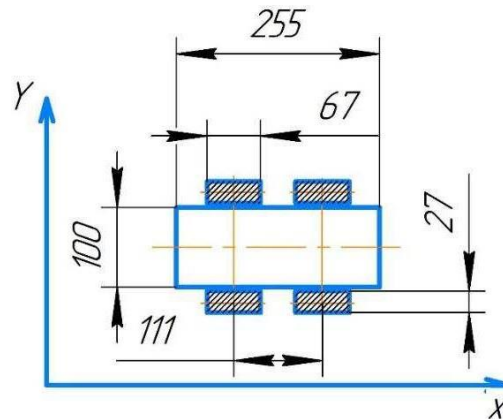


Рисунок 2.1 – Геометричні характеристики чотириколісного візка

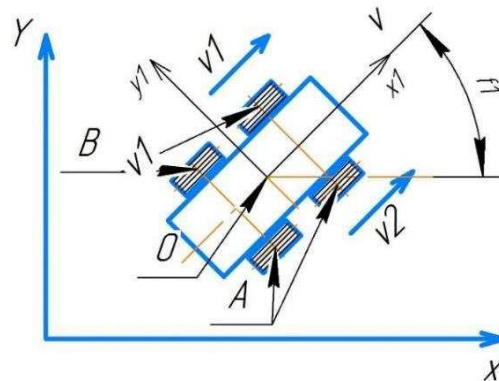


Рисунок 2.2 – Конструктивна схема чотириколісного візка

Модель робота представлена у вигляді з 5 твердих тіл: В – два колеса лівої частини, А – два колеса правої частини, О – це кришка з чорного оргскла (воно має два рівні).

В якості змінних стану колісного робота розглядаються такі величини: x_1, y_1 - координати базової точки робота; ϕ_1 - кут повороту центральної лінії платформи; v_1 - модуль вектора швидкості робота лівих коліс; v_2 - модуль вектора швидкості робота правих коліс; v - модуль вектора швидкості.

Кінематична модель робота описується наступною системою рівнянь:

$$\left. \begin{aligned} \dot{X} &= v \cdot \cos(f1) \\ \dot{Y} &= v \cdot \sin(f1) \\ f1 &= \omega \end{aligned} \right\} \quad (2.1)$$

де ω - швидкість обертання платформи довкола вертикальної осі.

Управління швидкостями руху:

- для прямолінійного руху колеса правої та лівої частини повинні обертатися з однаковими швидкостями
- для того, щоб робот розвернувся на місці, необхідно встановити швидкості однаковими по модулю, але спрямованими протилежно
- інші комбінації швидкостей призводять до руху дугою.

Формування функціональної схеми САП

Сформована функціональна схема системи автоматичного позиціонування кута положення МРК, основана на принципі управління за відхиленням. Функціональна схема представлена на рис. 3.1.

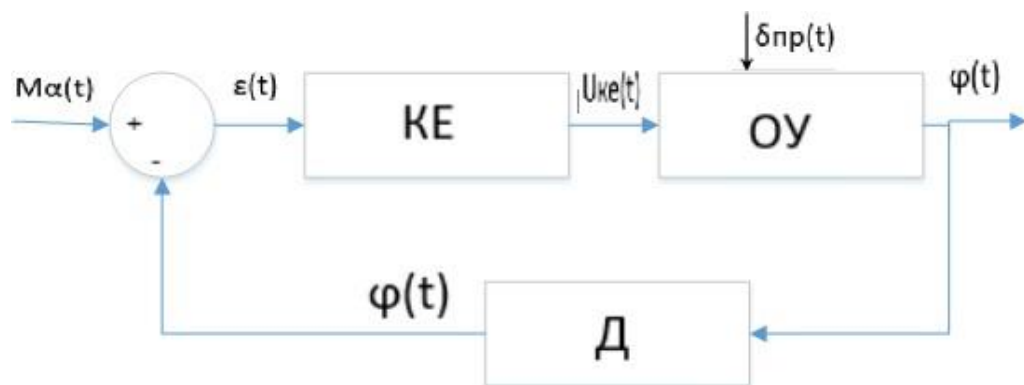


Рисунок 2.3 - Функціональна схема САП

На функціональній схемі (рис. 2.3) прийняті наступні позначення:

КЕ – коректуючий елемент;

ОУ – об’єкт управління при задаючому впливі;

Д – датчик виміру куту повороту МРК;

$M\alpha(t)$ – момент повороту МРК;

$\delta_{пр}(t)$ – збурюючий вплив;

$U_{ке}(t)$ – величина напруги з коригувального елемента;

$\varphi(t)$ – величина кута повороту МРК;

$\varepsilon(t)$ – різниця між вхідним та сигналом зворотного зв’язку.

Формування структурної схеми САП

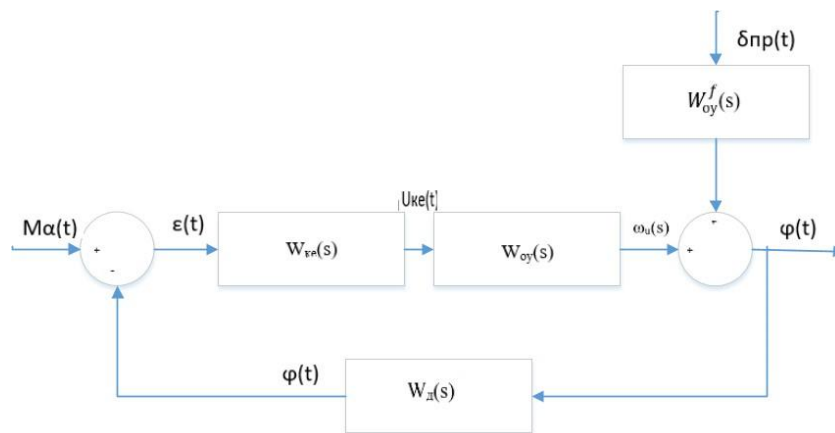


Рисунок 2.4 - Структурна схема САП

На структурній схемі (рис. 2.4) прийняті наступні позначення:

Передаточна функція коригувального пристрою:

$$W_{ke}(s) = \frac{U_{ke}(s)}{\varepsilon(s)} = K_p$$

$$W_{oy}(s) = \frac{\varphi(s)}{\psi(s)} = \frac{0.12844 s^3 + 11.34564 s^2 + 3s + 64.955}{0.12844 s^3 + 11.34564 s^2}$$

$$W_{oy}^f(s) = \frac{\varphi(s)}{\delta_{np}(s)} = \frac{-3.77}{0.12844 s^3 + 11.34564 s^2}$$

$$K_D = 1;$$

(2.2)

Передаточна функція розімкненої системи за керуючим впливом:

$$W_H^u(s) = \frac{\varphi(s)}{M_\alpha(s)} = W_{ke}(s)W_{oy}^u(s) \cdot \frac{1}{s} = \frac{K_p(0.12844 s^3 + 11.34564 s^2 + 3s + 64.955)}{0.12844 s^3 + 11.34564 s^2},$$

(2.3)

Передаточна функція розімкненої системи за збурюючим впливом:

$$W_H^f(s) = \frac{\varphi(s)}{\delta_{np}(s)} = W_{ke}(s)W_{oy}^f(s) \cdot \frac{1}{s} = \frac{K_p(-3.77)}{0.12844 s^3 + 11.34564 s^2}$$

(2.4)

Передаточна функція замкненої системи за керуючим впливом:

$$\begin{aligned} \phi_H(s) &= \frac{\varphi(s)}{M_\alpha(s)} = \frac{W_H^u(s)}{1 + W_H^u(s) \cdot W_D(s)} = \\ &= \frac{K_p(0.12844 s^3 + 11.34564 s^2 + 3s + 64.955)}{0.12844 s^3 + 11.34564 s^2 + K_p(0.12844 s^3 + 11.34564 s^2 + 3s + 64.955)} \end{aligned}$$

(2.5)

Передаточна функція замкнутої системи за збурюючим впливом:

$$\phi_H(s) = \frac{\varphi(s)}{\delta_{np}(s)} = \frac{W_H^f(s)}{1 + W_H^f(s)W_\partial(s)} = \frac{K_p(0.12844s^3 + 11.34564s^2 + 3s + 64.955)}{0.12844s^3 + 11.34564s^2 - 3.77K_p} \quad (2.6)$$

2.2 Маніпулятор

Маніпулятор – сукупність просторового важільного механізму та системи приводів, що здійснює під управлінням програмованого автоматичного пристрою або людини-оператора дії (маніпуляції), аналогічні діям руки людини.

2.2.1 Кінематична схема маніпулятора

Перед розробкою комп'ютерної моделі маніпулятора необхідно скласти структурну кінематичну схему. На схемі вказані 1 і 2 ланки, які повертаються на кути q_1 і q_2 , основа, навколо якої маніпулятор повертається на кут q_0 , і схоплення, а також проставлені довжини всіх частин маніпулятора.

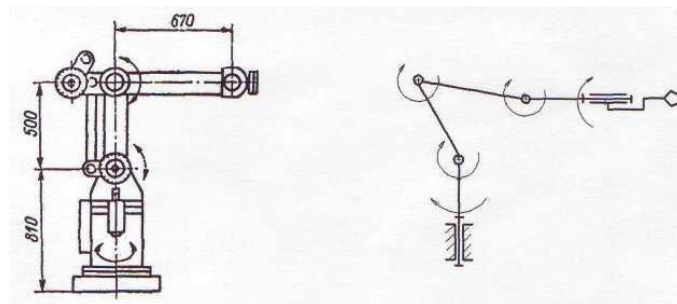


Рисунок 2.5 – Кінематична схема маніпулятора

У цій кінематичній схемі використовуються деякі додаткові ланки та елементи, які утворюють паралельний механізм, який дозволяє схвату маніпулятора залишатися завжди паралельно підставі незалежно від того, на які кути q_1 і q_2 повернуться ланки маніпулятора.

На рис. 1.4 показаний принцип роботи паралельного механізму, на якому видно, що при зміні кутів повороту ланок схват робота зберігає паралельне положення щодо основи. Дана властивість для захоплення, переміщення та встановлення предмета в нове положення, при цьому зберегти його вертикальне положення. Так само відпадає необхідність ставити 3 керовану ланку.

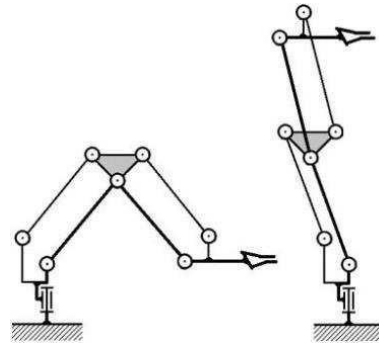


Рисунок 2.6 – Принцип роботи паралельного механізму

2.2.2 Визначення діапазонів кутів поворотів ланок

Побудувавши 3D модель маніпулятора з урахуванням усіх довжин ланок та розмірів деталей, можна визначити діапазони кутів поворотів ланок робота щодо початкового положення. Ці діапазони будуть враховуватися щодо робочої зони динамічної моделі маніпулятора.

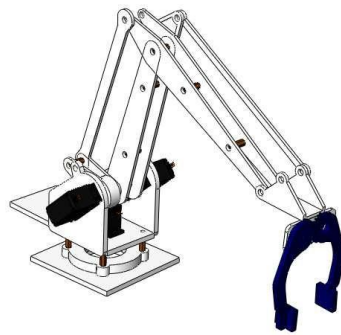


Рисунок 2.7 – 3D модель робота

Кут повороту основи q_0 . За поворот основи відповідає сервопривід MG996R, що обертається на підшипнику, діапазон його повороту становить 180° . У конструкції робота немає перешкод, що обмежують цей діапазон, отже, робот повертатиметься на $\pm 90^\circ$ щодо початкового положення.

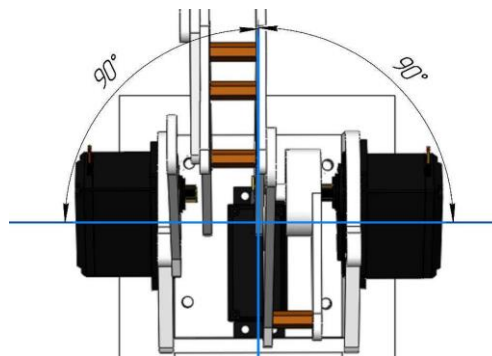


Рисунок 2.8 – Діапазон зміни кутів основи

Кут повороту 1-ї ланки q_1 .

За поворот першої ланки відповідає сервопривід MG996R, встановлений в основі робота, що передає кут повороту через важелі. Діапазон повороту: від 90 до 160 градусів (значення кутів сервоприводу).

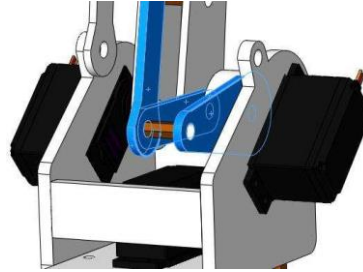


Рисунок 2.9 – Положення сервоприводу для керування ланки 1

Кут повороту 2-ї ланки q_2 .

За поворот другої ланки відповідає сервопривід MG996R, встановлений в основі робота, що передає кут повороту через важелі. Діапазон повороту: від 90 до 180 градусів (значення кутів сервоприводу).

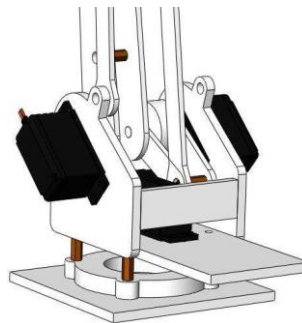


Рисунок 2.10 – Положення сервоприводу для керування ланки 2

2.2.3 Проектування схвату маніпулятора

Схват – це механічний захватний пристрій, аналог кисті руки людини. 3D модель схвата буде виконана із пластику на 3D-принтері.

Міжосьова відстань – відстань між осями зубчастих коліс.

Міжосьова відстань вибирається у зв'язку з конструкцією робота і дорівнює 28.8 мм.

У зубчастій передачі контактуючі елементи двох профілів виконуються по евольвентах кола і утворюють так зване евольвентне зачеплення.

Основними елементами зубчастого колеса є зуби, кожен зуб складається з головки зуба і ніжки: зубці знаходяться на ободі колеса і разом

з ободом складають зубчастий вінець: тонша частина колеса – диск з'єднує маточину з ободом.

Параметри зубчастого колеса:

- діаметр вершин зуба: 30.85 мм;
- діаметр западин зуба: 26.66 мм;
- кількість зубів: 29;
- модуль: 1.

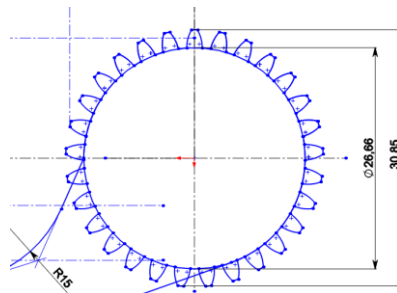


Рисунок 2.11 – Геометричні характеристики зубчастого колеса

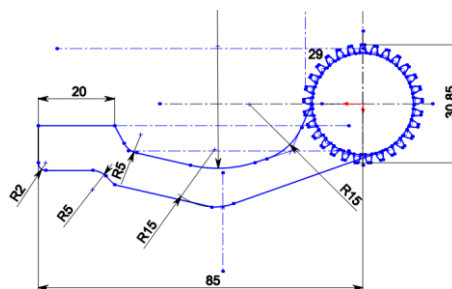


Рисунок 2.12 – Геометричні характеристики схвату

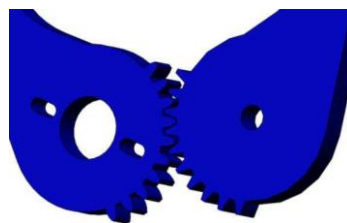


Рисунок 2.13 - Зубчасте зачеплення, 3D модель

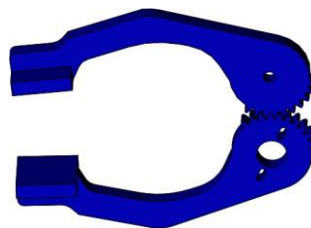


Рисунок 2.14 – Схват для маніпулятора

Для друку використовуємо PLA -пластик.

PLA-пластик (полілактид, ПЛА) – є біорозкладним, біосумісним, термопластичним аліфатичним поліефіром, структурна одиниця якого - молочна кислота.

Робота PLA-пластиком на 3D-принтері ведеться за допомогою технології моделювання методом пошарового наплавлення (FDM-Fused Deposition Modeling).

Нитка розплавляється, після чого доставляється за спеціальною насадкою на поверхню для роботи і осідає. У результаті будівництва моделі розплавленим пластиком створюється повністю готовий до застосування об'єкт.

Таблиця 2.1 – Характеристики PLA-пластику.

Величина	Значення
Температура плавлення	173-178°C
Температура розм'якшення	50°C
Твердість (по Роквеллу)	R70-R90
Відносне подовження при розриві	3,8%
Температура склювання	60-65°C
Густина	1,23-1,25 г/см ³
Точність друку	± 0,1%

2.3 Принципи управління сервоприводами

До сервоприводу тягнуться три дроти. Два з них відповідають за живлення мотора, третій доставляє керуючий сигнал, який використовується для виставлення положення пристрою.

Щоб вказати сервоприводу бажане положення, призначеному для цього дроту необхідно посилати керуючий сигнал. Керуючий сигнал - імпульси постійної частоти і змінної ширини.

Те, яке положення має зайняти сервопривід, залежить від довжини імпульсів. Коли сигнал надходить у схему, що управляє, наявний в ній генератор імпульсів виробляє свій імпульс, тривалість якого визначається через потенціометр. Інша частина схеми порівнює тривалість двох імпульсів.

Якщо тривалість різна, вмикається електромотор. Напрямок обертання визначається тим, який із імпульсів коротший. Якщо довжини імпульсів рівні, електромотор зупиняється.

Найчастіше у хобі-серверах імпульси проводяться із частотою 50Гц. Це означає, що імпульс випромінюється і приймається раз на 20мс. Зазвичай тривалість імпульсу в 1520мкс означає, що сервопривод повинен зайняти середнє положення. Збільшення або зменшення довжини імпульсу змусить сервопривід повернутися за годинниковою або проти годинникової стрілки відповідно. При цьому існує верхня та нижня межі тривалості імпульсу. У бібліотеці Servo для Arduino за замовчуванням виставлені значення імпульсу: 544 мкс — для 0° и 2400 мкс — для 180°.

Також варто відзначити, що це лише загальноприйняті довжини. Навіть у рамках однієї і тієї ж моделі сервоприводу може існувати похибка, що допускається при виробництві, що призводить до того, що робочий діапазон довжин імпульсів трохи відрізняється. Для точної роботи кожен конкретний сервопривід може бути відкалібрований: шляхом експериментів необхідно підібрати коректний діапазон, характерний саме нього.

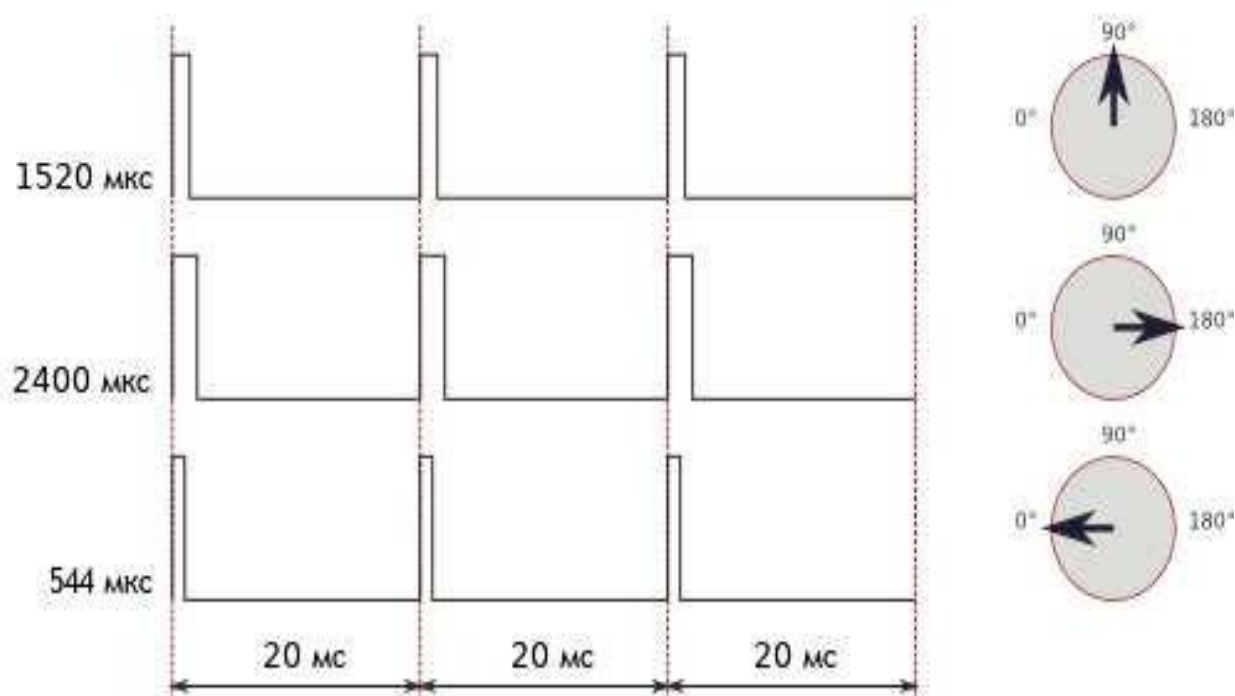


Рисунок 2.15 – Управління сервоприводом. Форми керуючих сигналів

2.4 Принципи управління моторами постійного струму

Управління двигуном постійного струму найпростіше організувати за допомогою ШІМ-регулятора. ШІМ – це широтно-імпульсна модуляція, в англійській мові це називається PWM — Pulse Width Modulation.

Широтно-імпульсно модульований сигнал - це імпульсний сигнал постійної частоти, але змінної шпаруватості (співвідношення тривалості імпульсу та періоду його прямування). Через те, що більшість фізичних процесів у природі мають інерцію, різкі перепади напруги від 1 до 0 будуть згладжуватися, приймаючи деяке середнє значення. За допомогою завдання шпаруватості можна змінювати середню напругу на виході ШІМ.

Якщо шпаруватість дорівнює 100%, то весь час на цифровому виході Arduino буде логічна напруга "1" або 5 Вольт. Якщо задати шпаруватість 50%, то половину часу на виході буде логічна "1", а половину - логічний "0", і середня напруга дорівнюватиме 2.5 Вольтам, і так далі.

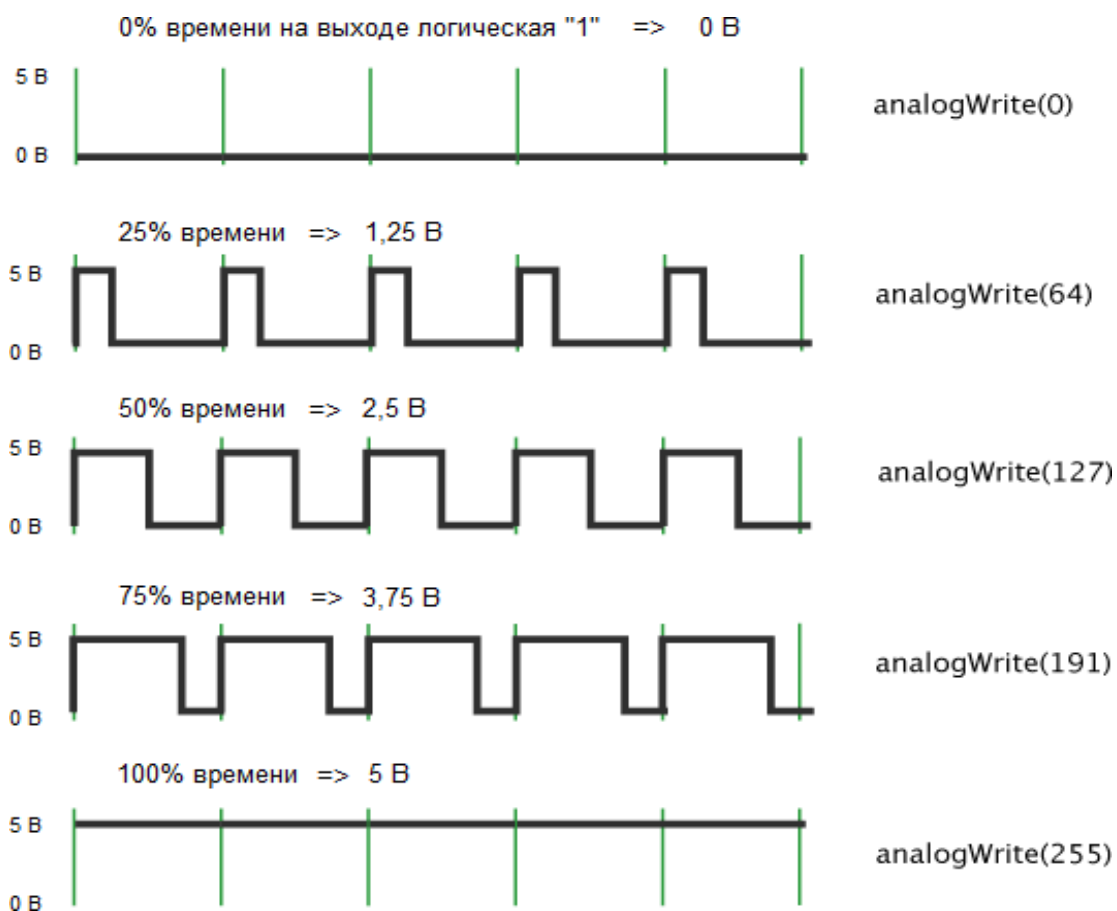


Рисунок 2.16 – Принцип роботи ШІМ в Arduino

Використовуючи L293D ми можемо управляти двигунами з досить широким діапазоном напруги живлення від 4.5 до 36 Вольт, але при цьому, L293D може видати всього лише 600mA тривалого струму навантаження на кожен канал. Піковий (максимальний) струм може швидко підскочити до 1.2A.

Вихід ENABLE1 це головна фігура в управлінні лівим каналом, без логічної одиниці на його виході нічого не працюватиме, незалежно від того, що відбувається на виходах INPUT1 і INPUT2.

Виходи INPUT1 та INPUT2 задають напрямок обертання мотора. Їх можна порівняти з кермом машини, тим більше, що в цьому випадку порівняння підходить ідеально, адже ми не можемо повернути кермо відразу в дві сторони, а необхідно вибрати одну з двох. Зі сказаного випливає, що для повороту нам треба подати логічну одиницю на вихід INPUT1, а на INPUT2 подати логічний нуль. Для зміни напрямку поміняти місцями INPUT1 "0", INPUT2 "1".

При подачі однакових логів двигун обертатися не буде, отже, обертання можна зупинити або подачею логічного нуля на висновок ENABLE1, при будь-якій конфігурації IN1 і IN2, або однаковими логам на IN1 і IN2, не змінюючи конфігурації виведення EN1 (даний варіант ми і розглянемо нижче) .

2.5 Протокол TCP/IP для обміну даними з wi-fi модулем

Керуючі команди, складені в середовищі матлаб, необхідно відправити на сервер, встановлений на wi-fi модулі. Передача даних здійснюється за протоколом TCP/IP.

TCP - Transfer Control Protocol. Протокол керування передачею. Він служить для забезпечення та встановлення надійного з'єднання між двома пристроями та надійної передачі даних. При цьому протокол TCP контролює оптимальний розмір пакета даних, що передається, здійснюючи нову посилку при збої передачі.

IP - Internet Protocol. Інтернет протокол чи адресний протокол - основа всієї архітектури передачі. Протокол IP служить доставки мережного пакета даних за потрібною адресою. При цьому інформація розбивається на пакети, які незалежно пересуваються мережею до потрібного адресата.

Схема передачі за протоколом TCP/IP.

1. Протокол TCP розбиває весь обсяг даних на пакети, нумерує їх, упаковуючи в TCP-конверти, що дозволяє відновити порядок отримання частин інформації. При поміщенні даних у такий конверт відбувається обчислення контрольної суми, яка записується потім у TCP-заголовок.

2. Далі за допомогою протоколу IP всі пакети передаються безпосередньо отримувачу.

3. Потім за допомогою протоколу TCP відбувається перевірка того, чи всі пакети отримані. Якщо під час прийому обчислення заново контрольна сума не збігається із зазначеною на конверті, це свідчить про те, що частина інформації була втрачена або спотворена під час передачі, протокол TCP/IP знову запитує пересилання цього пакета. Також потрібне підтвердження надходження даних від одержувача.

4. Після підтвердження отримання всіх пакетів протокол TCP упорядковує їх відповідним чином та збирає наново в єдине ціле. Протоколом TCP використовуються повторні передачі, періоди очікування (чи тайм-аути), що забезпечує надійність доставки інформації. Пакети можуть передаватися у двох напрямках одночасно.

2.6 Інтерфейс I²C для обміну даними гіроскопа і мікроконтролера

I²C (ІІС, англ. Inter-Integrated Circuit) – послідовна асиметрична шина для зв'язку між інтегральними схемами всередині електронних приладів. Використовує дві двоспрямовані лінії зв'язку (SDA та SCL), застосовується для з'єднання низькошвидкісних периферійних компонентів з процесорами та мікроконтролерами.

Дані передаються по двох дротах - дроту даних та дроту тактів. Є ведучий (master) і ведений (slave), такти генерує master, ведений лише "підтакує" при прийомі байта. Усього на одній дводротовій шині може бути до 127 пристроїв.

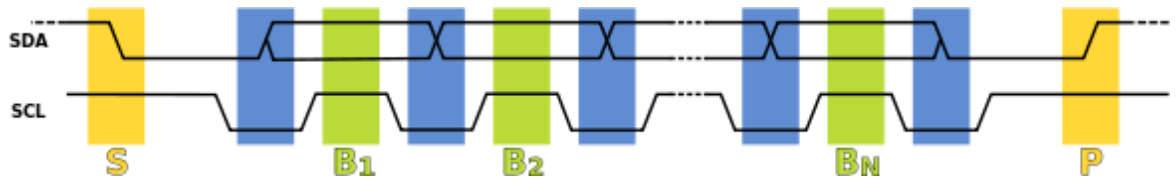


Рисунок 2.17 – Тактування послідовності передачі даних

I²C використовує дві двонаправлені лінії, підтягнуті до напруги живлення і керовані через відкритий колектор або відкритий стік - послідовна лінія даних (SDA, англ. Serial Data) та послідовна лінія тактування (SCL, англ. Serial CLock). Стандартні напруги +5В або +3.3В, проте допускаються й інші.

Класична адресація включає 7-бітовий адресний простір із 16 зарезервованими адресами. Це означає, що розробникам доступно до 112 вільних адрес для підключення периферії на одну шину.

Основний режим роботи – 100 кбіт/с; 10 кбіт/с у режимі роботи зі зниженою швидкістю. Також важливо, що стандарт допускає припинення тактування для роботи з повільними пристроями.

2.7 Висновки за розділом

В розділі були зроблені аналіз і синтез системи управління мобільного маніпуляційного робота.

3 КОНСТРУКТОРСЬКА ЧАСТИНА ТА ОПИС ВИКОРУСТОВУВАННЯ КОМПОНЕНТІВ

3.1 Привідна система коліс

Для управління рухом мобільного робота використовуються 4 мотор-редуктори. Управління двигунами здійснюється незалежно один від одного.



Рисунок 3.1 – Мотор-редуктор DC-6v

Мікро-електродвигун постійного струму встановлений в пластмасовий корпус, в якому знаходиться редуктор, що знижує швидкість обертання пластмасових шестерень і збільшує силу на валу механізму.

На вал мотор-редуктора насаджується колесо з гумовою шиною. Вал виходить із двох сторін корпусу редуктора. Колесо може встановлюватись з будь-якої сторони корпусу. На другу частину валу мотор-редуктора встановлюють диск з отворами, що дозволяє працювати оптичному датчику контролю параметра обертання валу.

Таблиця 3.1 – Характеристика двигуна постійного струму для платформи

Характеристика	Значення
Робоча напруга	3-12В
Рекомендована напруга	6-8В
Передатне число редуктора (без навантаження 3В)	1:48
Номінальний споживаний струм	70мА
Максимальний струм (при 6В)	250мА
Швидкість обертання (при 6В)	90 об/хв
Обертаючий момент	2 кг·см
Шум	65 dB

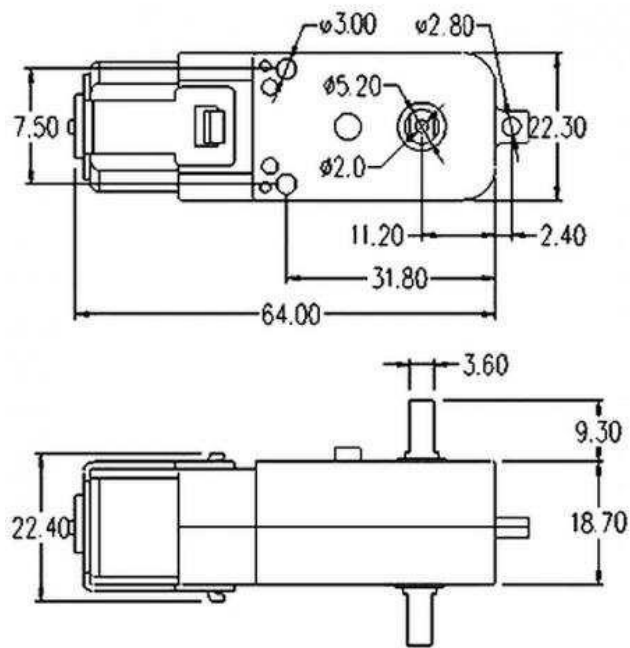


Рисунок 3.2 – Розміри мотор-редуктора DC-6v

3.2 Сервопривід MG996R

Привід – електромотор з редуктором. Щоб перетворити електрику на механічний поворот, необхідний електромотор. Однак найчастіше швидкість обертання мотора буває занадто великою для механізму з шестерень, що передає і перетворює крутний момент.

Вмикаючи та вимикаючи електромотор, можна обертати вихідний вал – кінцеву шестерню сервоприводу, до якої можна прикріпити щось, чим хочемо керувати. Однак, щоб положення контролювалося пристроєм, необхідний датчик зворотного зв'язку - енкодер, який буде перетворювати кут повороту назад в електричний сигнал. Для цього часто використовується потенціометр. При повороті бігунка потенціометра відбувається зміна опору, пропорційне куту повороту. Таким чином, за його допомогою можна встановити поточний стан механізму.

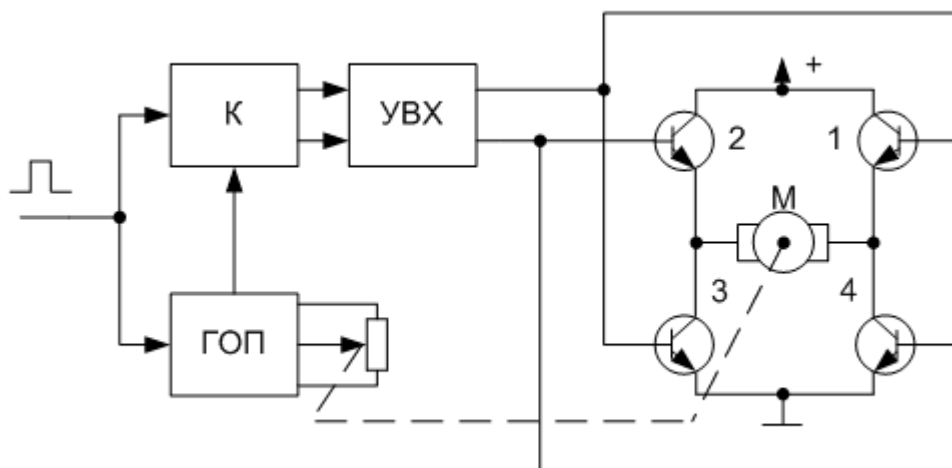


Рисунок 3.3 – Пристрій сервоприводу

Крім електромотора, редуктора та потенціометра в сервоприводі є електронна начинка, яка відповідає за прийом зовнішнього параметра, зчитування значень з потенціометра, їх порівняння та включення/вимикання мотора. Вона й відповідає за підтримку негативного зворотного зв'язку.



Рисунок 3.4 – Сервопривід MG996R

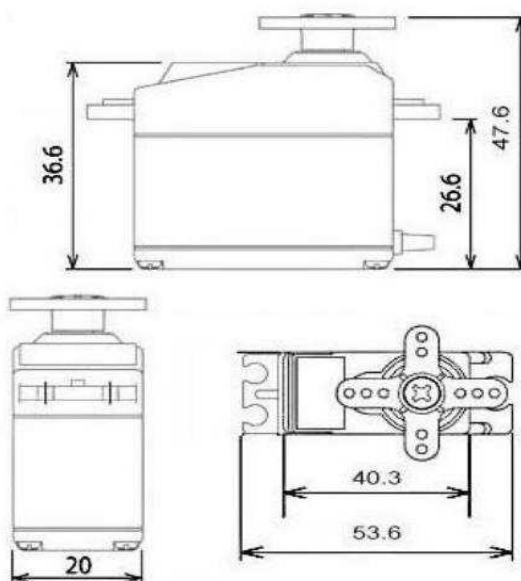


Рисунок 3.5 – Розміри сервоприводу MG996R

Таблиця 3.2 – Характеристики сервоприводу MG996R

Характеристики	Значення
Робоча напруга	4.8-7.2В
Матеріал редуктора	Металевий
Маса	55 г
Номінальний споживаний струм	500 – 900 мА
Пусковий струм	До 2.5А
Швидкість обертання (при 6В)	60°/0.14s
Крутний момент (при 4.8В)	9.4 кг·см
Крутний момент (при 6В)	11 кг·см

3.3 Сервопривід MG90S



Рисунок 3.6 – Сервопривід MG90S

Таблиця 3.3 – Характеристики сервоприводу MG90S

Характеристики	Значення
Робоча напруга	4.8-6В
Матеріал редуктора	Металевий
Маса	13.4 г
Номінальний споживаний струм	12 – 100 мА
Пусковий струм	До 1А
Швидкість обертання (при 6В)	60°/0.08s
Крутний момент (при 4.8В)	1.8 кг·см
Крутний момент (при 6В)	2.2 кг·см

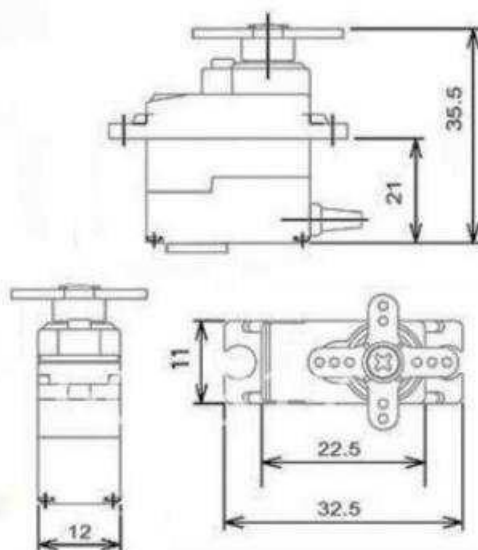


Рисунок 3.7 – Розміри сервоприводу MG90S

3.4 Реверсивний драйвер двигунів L293D

Мікросхема включає відразу два драйвера для управління слабострумними моторами.

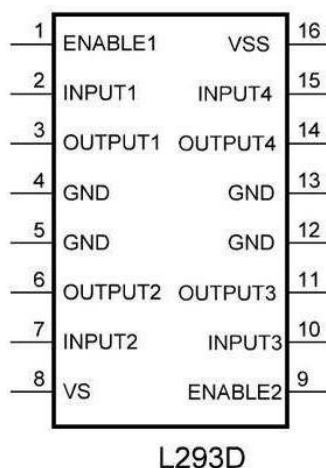


Рисунок 3.8 – Мікросхема L293D

Безперечним плюсом даної мікросхеми є роздільне живлення логічної частини мікросхеми, напруга живлення якої лежить у межах 4.5-5 вольт (VSS), і силової частини живлення двигунів (VS) .

Використовуючи цю мікросхему, ми можемо керувати двигунами з досить широким діапазоном напруги живлення від 4.5 до 36 вольт, але при

цьому, L293D може видати всього лише 600 mA тривалого струму навантаження на кожен канал. Піковий (максимальний) струм може швидко підскочити до 1.2A. Також з позитивних сторін даної мікросхеми слід відзначити її невибагливість до напруги вхідних сигналів, що подаються на виходи INPUT. Логічний "0" розпізнається мікросхемою, коли вхідна напруга <1.5 Вольт. Логічна "1" з'являється при вхідній напрузі, що лежить у межах від 2.3 до 7 Вольт.

Діапазон робочих температур від -40°C до +150°C.

3.5 Arduino Mega 2560

Arduino Mega 2560 – флагманська платформа для розробки на базі мікроконтролера ATmega2560. На платі передбачено все необхідне для зручної роботи з мікроконтролером: 54 цифрових входів/виходів (з яких 15 можуть використовуватися як ШІМ-виходи), 16 аналогових входів, 4 UART (апаратних передопередавача для реалізації послідовних інтерфейсів), кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для внутрішньосхемного програмування та кнопка скидання.

Серцем платформи Arduino Mega 2560 є 8-бітний мікроконтролер сімейства AVR — ATmega2560. Він надає 256КБ флеш-пам'яті для зберігання прошивки, 8КБ оперативної пам'яті SRAM та 4КБ енергонезалежної пам'яті EEPROM для зберігання даних.

Порти входу/виходу:

- цифрові входи/виходи: піни 0-53;
- логічний рівень одиниці – 5В, нуля – 0В;
- ШІМ: піни 2-13 та 44-46;
- АПЦ: піни А0-А16;
- TWI/I²C: піни 20(SDA) та 21(SCL);
- SPI: піни 50(MISO), 51(MOSI), 52(SCK) и 53(SS);
- UART: Serial: піни 0(RX) и 1(TX); Serial1: піни 19(RX) и 18(TX).

Ці виходи використовуються для отримання (RX) та передачі (TX) даних за послідовним інтерфейсом. Виходи 0(RX) і 1(TX) з'єднані з відповідними виходами мікросхеми ATmega16U2, яка виконує роль перетворювача USB-UART.

MEGA PINOUT

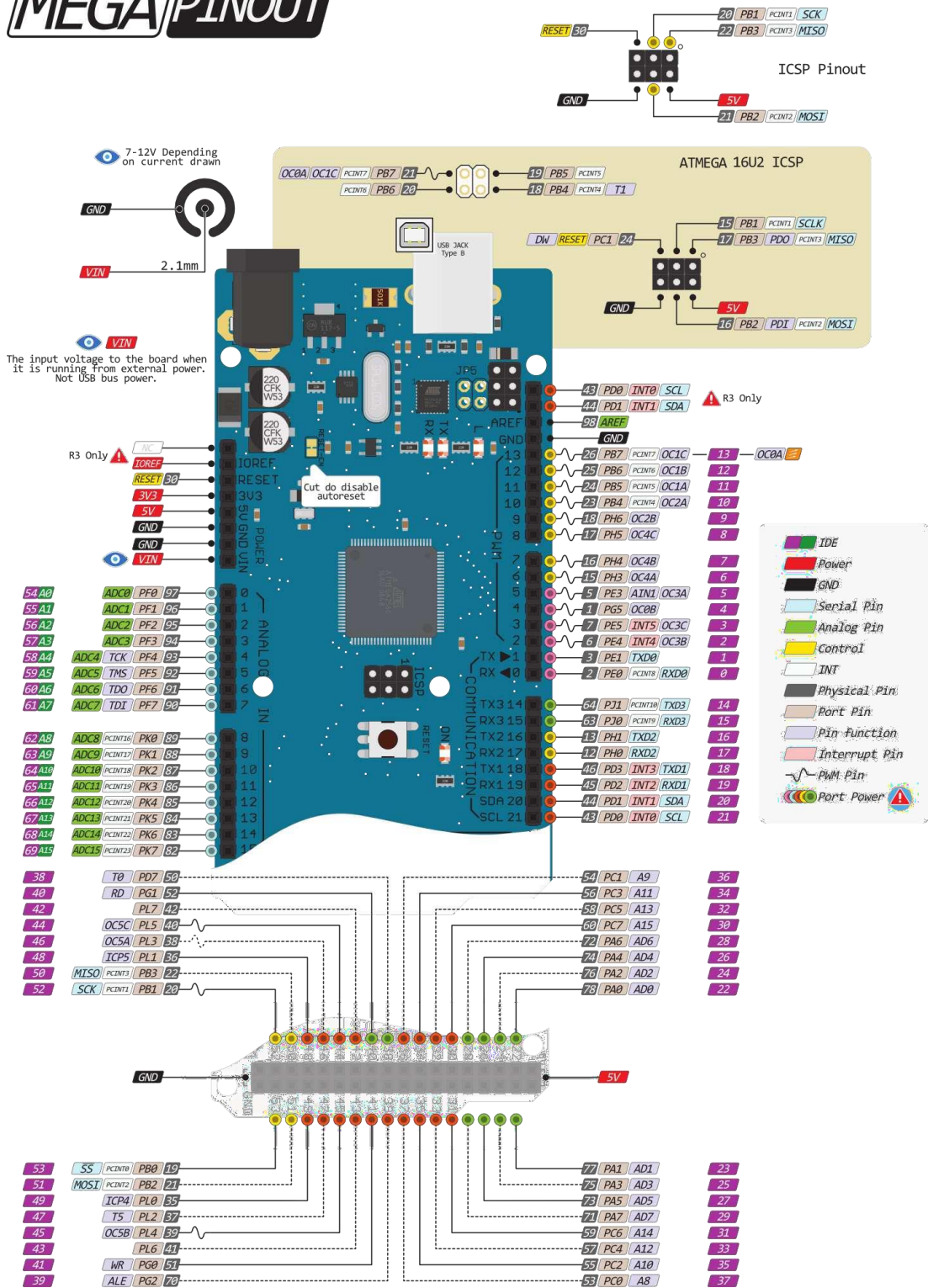


Рисунок 3.9 – Розпіновка плати

Таблиця 3.4 – Характеристика Arduino Mega 2560

Характеристика	Значення
Мікроконтролер	ATmega2560
Тактова частота	16 МГц
Напруга логічних рівнів	5 В
Вхідна напруга живлення	7–12 В
Максимальний струм із піна введення виводу	40 мА
Максимальний вихідний струм піна 3.3V	50 мА
Максимальний вихідний струм піна 5V	800 мА
Оперативна пам'ять	8 КБ
Розрядність АЦЦ	10 біт
Flash-пам'ять	256 КБ
EEPROM-пам'ять	4 КБ

3.6 Лінійний стабілізатор напруги L293D

Стабілізатор - це пристрій автоматичної підтримки сталості значення електричної напруги на входах приймачів електричної енергії.

7805 - стабілізатор, виконаний у корпусі, схожому на транзистор і має три виходи. Див. рис. 3.12 (+5V стабілізованої напруги і струм 1A). Так само в корпусі є отвір для кріплення стабілізатора напруги 7805 до радіатора охолодження. 7805 є стабілізатором позитивної напруги.

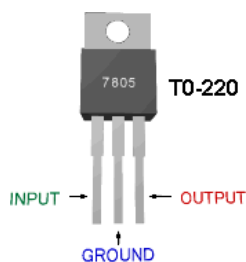


Рисунок 3.10 – Лінійний стабілізатор напруги L7805

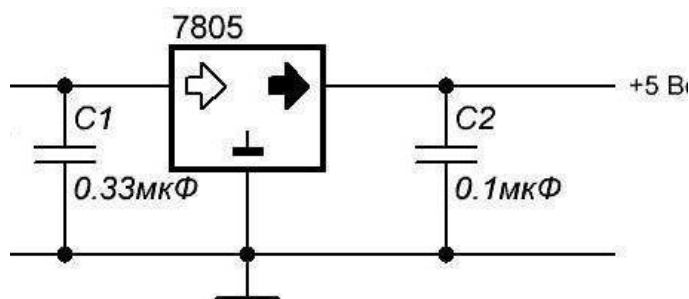


Рисунок 3.11 – Схема підключення стабілізатора

3.7 Блок живлення

Для живлення мобільного робота на початковому етапі використовуємо блок живлення «AC/DC adapter 12V 5A».

Таблиця 3.5 – Характеристика блока живлення

Характеристика	Значення
Вхідна напруга	AC 100-240 В
Вихідна напруга	DC 12 В
Вихідний струм	5 А
Робоча температура	0 ~ 40 С
Вбудований захист від перенапруги, надструму та короткого замикання	

3.8 Оптичний енкодер FC-03

Сприймаючий елемент модуля - оптична пара ITR9608. В одному виступі корпусу розташований інфрачервоний світлодіод, спрямований на фототранзистор, що знаходиться в іншому виступі. Якщо у щілину між виступами внести непрозору пластину, то ІЧ випромінювання від світлодіода перекривається і фототранзистор закривається. Такий компонент іноді називають фотопереривачем, але він лише фіксує переривання світлового потоку. Як показано на схемі з емітера транзистора, сигнал надходить у схему приладу.

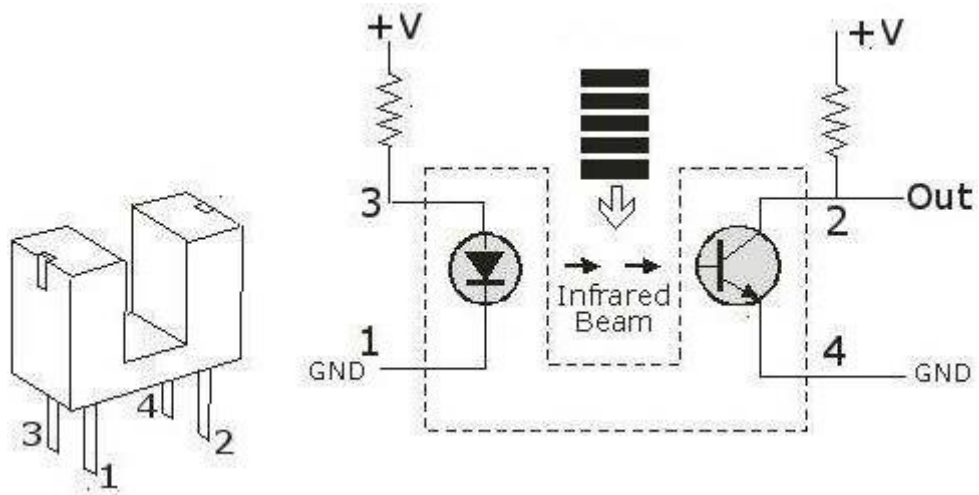


Рисунок 3.12 – Схема роботи датчика швидкості обертання

Світло перегорджує пластина, закріплена на рухомій частині механізму, що контролюється. За допомогою датчика визначають переміщення в крайні положення площини, що рухається, коли пов'язаний з нею елемент конструкції входить в датчик. Вимірюють параметри обертання різних механічних деталей. Так відбувається перетворення механічних параметрів в електричні величини і далі в програмні значення.

Датчик швидкості обертання FC-03 використовується для вимірювання частоти обертання. Визначити з його допомогою напрямок обертання не можна. На валу двигуна або шестірні редуктора встановлюється диск з отворами. Оптичара модуля має проріз шириною 5 мм. При обертанні диска постійно в прорізі чергуються отвори та ділянки пластини. Датчик перетворює чергування елементів диска електричні імпульси. Електроніка модуля робить сигнал датчика придатним сприйняття цифровими логічними мікросхемами чи мікроконтроллером.

На платі розташовано 2 світлодіоди. Один показує включення датчика швидкості обертання FC-03, інший – відсутність перешкоди датчика. При першому ознайомлювальному включенні обидва світлодіоди повинні світитися. При внесенні металевої пластини у вільний простір переривника один світлодіод повинен згаснути.



Рисунок 3.13 – Диск із прорізами із чорного оргскла для оптичного датчика

Таблиця 3.6 – Характеристика блока живлення

Характеристика	Значення
Живлення	3,3 – 5 В
Ширина просвіту для диска з прорізами	5 мм
Розміри модуля	38 x 14 x 7 мм
Кріпильний отвір під гвинт	3 мм



Рисунок 3.14 – Оптичний датчик швидкості

Контакти промарковані:

- VCC – живлення;
- GND – загальний провід, негативний полюс живлення;
- OUT – вихід імпульсного сигналу.

3.9 Wi-fi модуль NodeMCU LUA R2

NodeMCU – платформа на основі ESP8266 для створення різних пристроїв IoT. Модуль вміє відправляти та отримувати інформацію в локальну мережу або в інтернет за допомогою Wi-Fi. Недорогий модуль

часто використовується для створення систем розумного будинку або роботів Arduino, керованих на відстані.

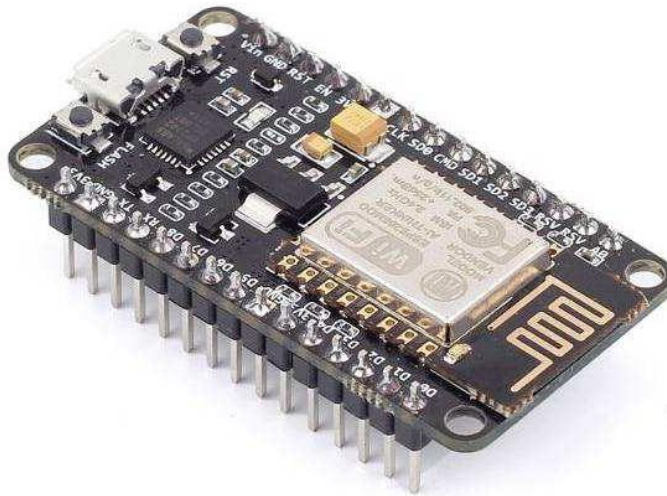


Рисунок 3.15 – Зовнішній вигляд модуля

Технічні характеристики модуля:

- підтримує Wi-Fi протокол 802.11 b/g/n;
- підтримувані режими Wi-Fi – точка доступу, клієнт;
- вхідна напруга 3.7В – 20В;
- робоча напруга 3В – 3.6В;
- максимальний струм 220мА;
- вбудований стек TCP/IP;
- діапазон робочих температур від -40°C до 125°C;
- 80МГц, 32-бітний процесор;
- час пробудження та відправки пакетів 22мс;
- вбудований TR перемикач і PLL;
- наявність підсилювачів потужності, регуляторів, систем управління живленням.

Модуль V3 має 11 контактів входу/виходу загального призначення. Крім того, деякі виходи мають додаткові функції:

- D1-D10 – виходи із широтно-імпульсною модуляцією;
- D1, D2 – виходи для інтерфейсу I²C/TWI;

- D5–D8 – виходи для інтерфейсу SPI;
- D9, D10 – UART;
- A0 – вхід с АЦП.

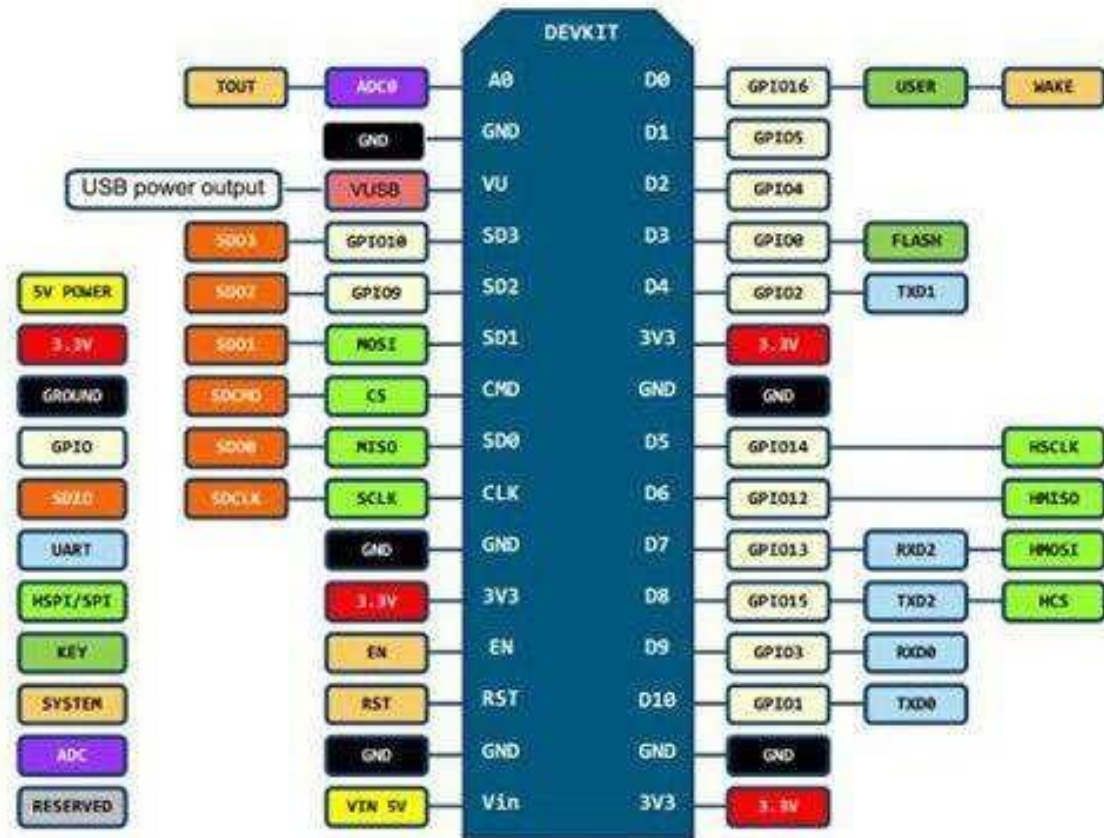


Рисунок 3.16 – Порти модуля

3.10 Акселерометр і гіроскоп MPU6050 54

Гіроскоп - це пристрій, який реагує на зміну кутів орієнтації тіла, що контролюється. У класичному виконанні це якийсь інерційний об'єкт, який обертається на суспензіях. Як результат, обертовий об'єкт завжди буде підтримувати свій напрямок, а за положенням підвіски ви можете визначити кут відхилення.

Акселерометр - це пристрій, який вимірює проєкцію очевидного прискорення, тобто різницю між справжнім прискоренням об'єкта та гравітаційним прискоренням. У простому прикладі така система - це маса, закріплена на підвісці з еластичністю (пружина для хорошого прикладу). Отже, якщо така система буде повернена під деяким кутом, або кинута, або

задана лінійному прискоренню, то еластична підвіска реагуватиме на рух під впливом маси та відхилення, а прискорення визначатиметься за цим відхиленням. Таким чином, гіроскоп реагує на зміну у просторі, незалежно від напрямку руху, за допомогою акселерометра може вимірювати лінійне прискорення об'єкта, а також і штучно обчислене розташування об'єкта в просторі. Кожен пристрій має власні переваги та недоліки.

Мікросхема MPU6050 містить як акселерометр, так і гіроскоп на борту, а крім цього ще й датчик температури. MPU6050 - основний елемент модуля GY-531. На додаток до цієї мікросхеми, на платі модуля є необхідна обв'язка MPU6050, в тому числі підтягуючі резистори інтерфейсу I2C, а також стабілізатор напруги на 3.3В з низьким падінням напруги (при нарузі уже в 3,3 В на виході стабілізатора буде рівно 3В) з фільтрувальними конденсаторами. Ну і бонус, на платі є SMD світлодіод з обмежувальним резистором як індикатор напруги живлення. Розмір плати модуля GY-521 10 x 20 мм.

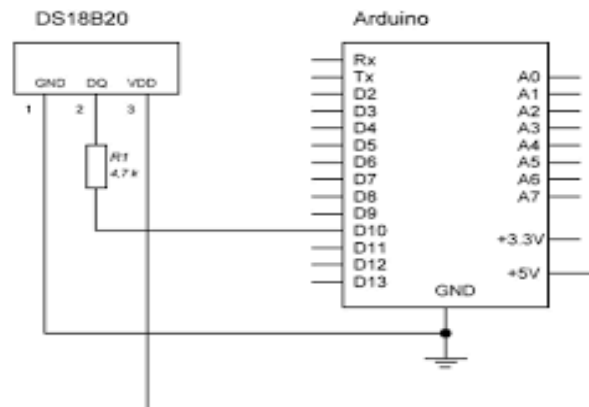


Рисунок 3.17 – Схема підключення до Arduino

Характеристики MPU6050:

- напруги потужності 2,375 - 3,46В;
- струм, що споживається, до 4мА;
- інтерфейс передачі даних - I²C;
- максимальна швидкість I²C – 400кГц;
- вхід для інших датчиків I²C;

- внутрішній генератор на 8 МГц (поза модулем є здатність підключити зовнішній кварцовий резонатор на 32,768 кГц або 19,2 МГц).

Функції MPU6050:

- три осьовий MEMS гіроскоп з 16-бітним АЦП;
- три осьовий MEMS акселерометр з 16-бітним АЦП;
- digital Motion Processor (DMP);
- slave I2C для підключення до мікроконтролера;
- master I2C для підключення до мікросхеми додаткового датчика;
- реєстри даних датчиків;
- FIFO;
- переривання;
- температурний сенсор;
- самоперевірка гіроскопа та акселерометра;
- реєстр ідентифікації приладу.

3.11 Система підключення живлення для всіх елементів

Для живлення основних елементів мобільного робота використовується блок живлення на 12 Вольт. Це суперечить визначенню мобільного робота, він не повинен бути прив'язаний його до проводів, він має бути повністю автономним. Але у зв'язку з бюджетом проєкту, на етапі розробки, було вирішено використовувати живлення.

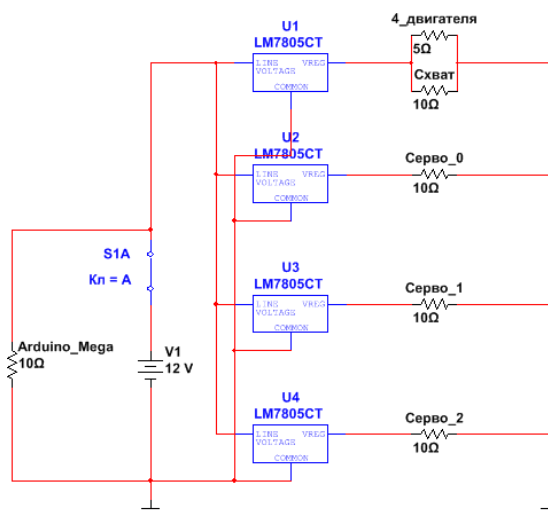


Рисунок 3.18 – Схема живлення елементів мобільного робота

Штекер від блока живлення підключений у гніздо 2.1 мм, далі встановлена кнопка, що відповідає за подачу живлення на стабілізатори. На схемі встановлені 4 стабілізатори напруги L7805 на 5 Вольт.

Розрахунок споживання струму всім елементів (до розрахунку приймаємо середні значення споживання струму):

- приводна система: 4 мотор-мотор редуктора із середнім споживанням струму 150мА, отже, знадобиться 600мА;
- сервопривід MG996R: у роботі 3 таких елементи, із середнім споживанням 700мА, отже, знадобиться 2.1А;
- сервопривід MG90S: призначений для схвату із середнім споживанням струму 400мА.

Якщо встановити один стабілізатор, струму не вистачає для нормального функціонування всіх елементів (1.5А - максимальна сила струму, яку видає стабілізатор), тому доцільно встановити паралельно 4 стабілізатори, і рівномірно розподілити навантаження.

3.12 Схема управління кожним елементом

На рис. 3.19 показано тільки підключення сигналу, що управляє, а живлення реалізовано в розділі вище. Сервоприводи підключені до входів 10-13, які дозволяють виводити аналогові 8-бітові значення у вигляді ШІМ-сигналу.

Мотор-редуктори розділені на праву вісь та ліву. Два двигуни підключаються до однієї сторони каналу драйвера L293D, і два з іншого боку. Виходи мікросхеми це: IN1, IN2, IN3, IN4, EN1, EN2.

Для бездротової передачі даних використовується wi-fi модуль, підключений до TX3, RX3.

На рис. 3.20 зображено зібраний робот.

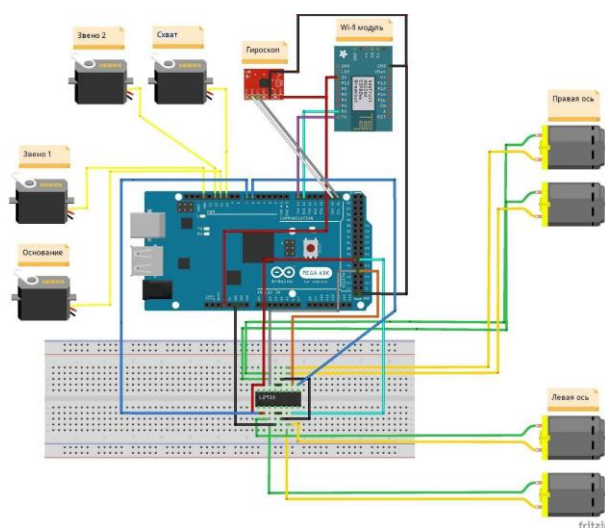


Рисунок 3.19 – Схема підключення всіх компонентів робота

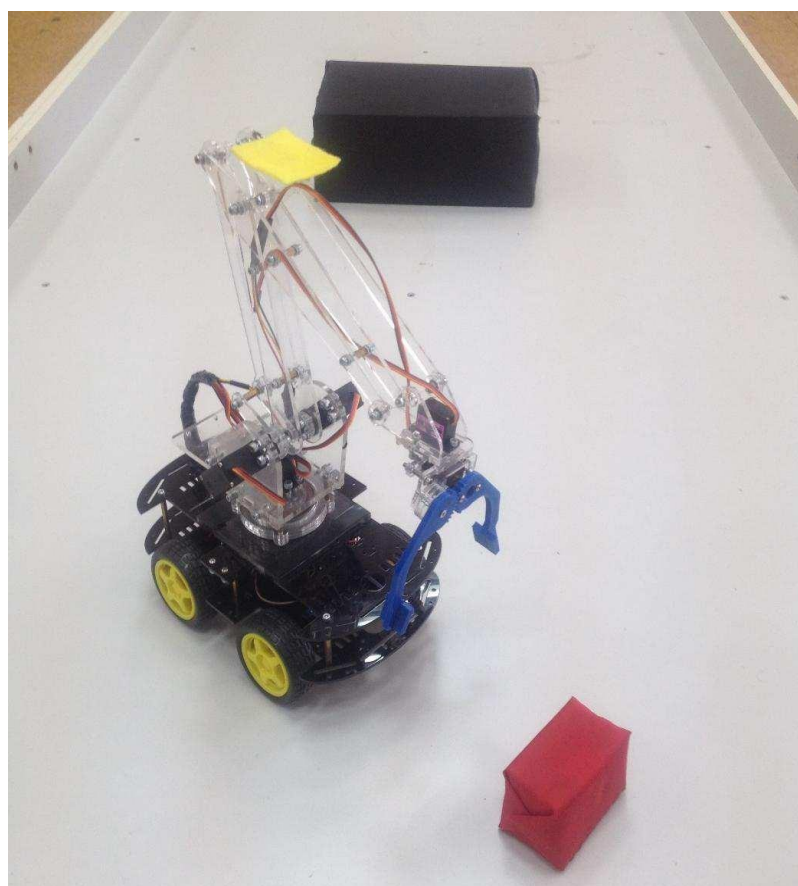


Рисунок 3.20 – Мобільний маніпуляційний робот

3.13 Висновки за розділом

В розділі були описані всі елементи, які були використані при виготовленні інтелектуального мобільного робота.

4. ДОСЛІДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ МОБІЛЬНИМ РОБОТОМ

4.1 Постановка завдання

Обробка зображень проводиться у бібліотеці OpenCV. Пакет інструментів надає великий набір еталонних алгоритмів та графічних інструментів для обробки, аналізу, візуалізації зображень та розробки алгоритмів. Він дозволяє здійснювати покращення якості зображення, усунення розмитості зображення, виділення особливостей, шумозаглушення, сегментацію зображення, геометричні трансформації, суміщення зображень. Багато функцій набору підтримують багатоточне виконання сучасних багатоядерних та багатопроцесорних комп'ютерів.

Завдання управління мобільним роботом у середовищі з перешкодами формулюється наступним чином: скласти список команд керування роботом, який забезпечує його переміщення до заданої мети за наявності цифрового зображення робочої зони робота при її виді зверху (приклад показаний на рис. 4.1) та значення масштабу зображення.

Розв'язання задачі складається з наступних етапів:

- 1) отримання знімка з камери;
- 2) виділення основних кольорів на фоні білої підлоги;
- 3) зменшення обсягу матриці та пошук початкової та кінцевої координати та перешкод;
- 4) апроксимація перешкоди прямокутниками;
- 5) заповнення матриці за допомогою хвильового алгоритму Лі;
- 6) пошук найкоротшого шляху;
- 7) вирішення зворотної задачі кінематики;
- 8) формування списку керуючих команд керування роботом.

4.2 Отримання знімка з камери

На відстані 3.5м від підлоги встановлена веб-камера Sven IC-975 HD. Камера підключена до ПК через USB 2.0 і передає дані до пакета матлаб для використання веб-камер.

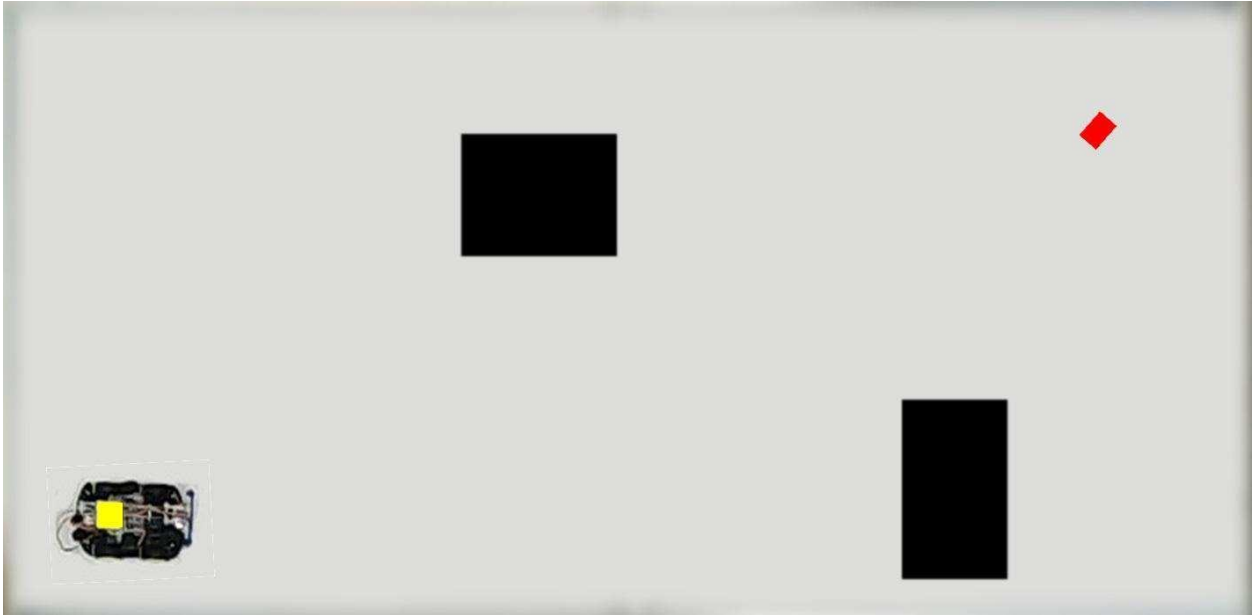


Рисунок 4.1 – Робоча зона робота

4.2.1 Виділення основних кольорів

Як можна побачити на малюнку 5.1, є чотири основні кольори: жовтий, чорний, червоний і колір покриття підлоги.

Тепер зображення дорівнює матриці $700 \times 1420 \times 3$. Кожен осередок складається з трьох значень від 0 до 255. Це адитивна кольорова модель RGB. Застосуємо функцію `rgb2ind`, яка створює палітрове зображення X із повнокольорового RGB, складаючи палітру `map` із усіх унікальних кольорів, представлених у вихідному зображенні.



Рисунок 4.2 – Індекс кінцевої точки

Після застосування цієї функції складаємо матрицю з елементів: 0,1,2 - 1.0 - це вільний прохідний осередок; 2 - це початкове положення робота; 1 – положення об'єкта для маніпулювання.

4.2.2 Зменшення обсягу матриці і пошук початкової і кінцевої координати

Зараз зображення дорівнює матриці 700x1420. Але це великий обсяг інформації, що можна зменшити зображення в 10 разів, розбивши на прямокутники.

Застосовуючи функцію "mat2cell", задавши потрібні параметри, отримуємо матрицю 70x142.

Пошук координат відбувається з пошуку індексу кольору в матриці.

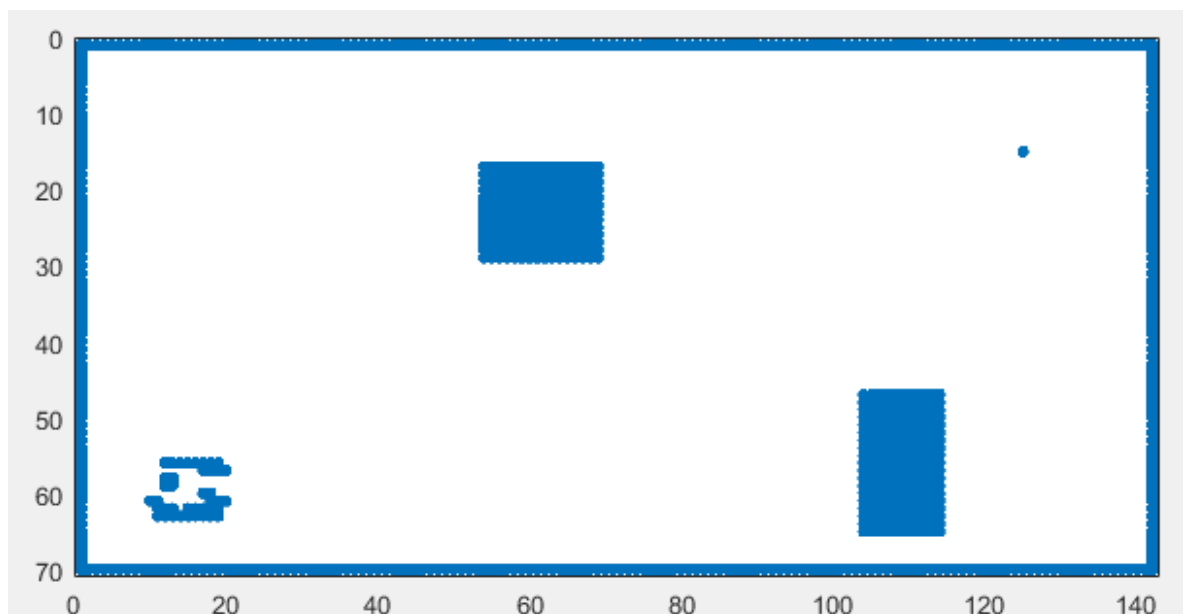


Рисунок 4.3 – Матриця після перетворень

Як видно з рис. 4.3 початкове положення робота визначається ще й іншими значеннями, оскільки мобільний робот має забарвлення чорного кольору. Програма вирішує, що початкове становище оточене перешкодами, і шляхів немає. Тому потрібно всі елементи навколо прирівняти до індексу покриття.

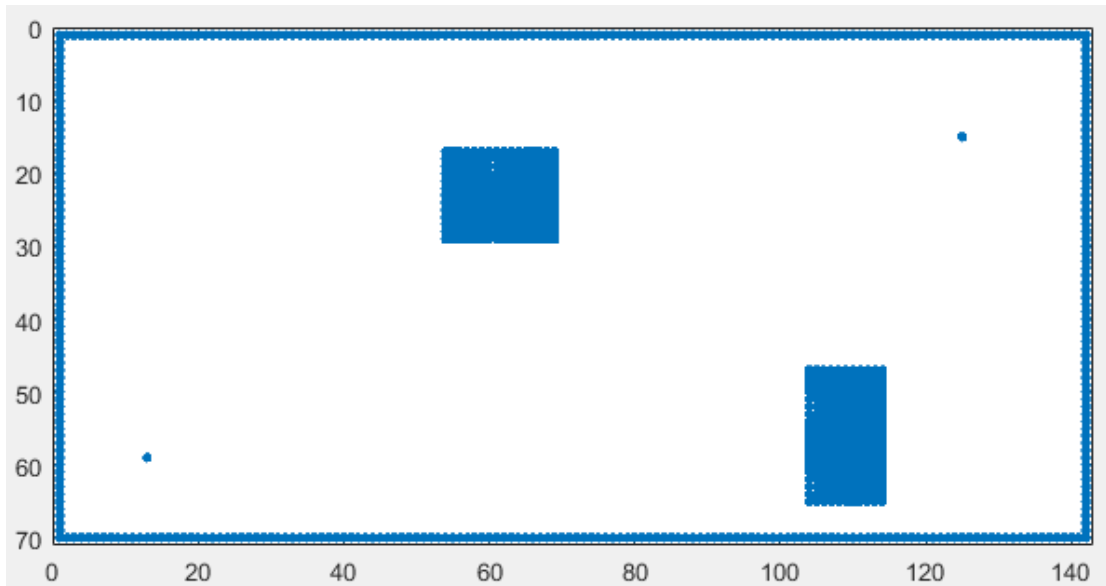


Рисунок 4.4 – Готова матриця з початковою і кінцевою координатою в середовищі з перешкодами

4.2.3 Апроксимація перешкод прямокутниками

Цей етап необхідний, щоб був запас ходу у робота поблизу перешкод. Робот починає поворот тільки коли зустрівся з перешкодою, але на практиці було виявлено, що простору бракує для повороту, і якщо робот буде розуміти, що перешкода знаходиться трохи ближче, то він заздалегідь почне виконувати поворот.

Алгоритм роботи:

- 1) перебираємо масив зображення A за рядками та стовпцями. При $A(i,j) = -1$ запам'ятовуємо індекс першого елемента;
- 2) по кроці спускаємося вниз і рахуємо кількість $A(i,j) = -1$ поки $A(i,j)$ не стане 0;
- 3) по кроці рухаємося вправо і вважаємо кількість $A(i,j) = -1$ поки $A(i,j)$ не дорівнюватиме 0;
- 4) запам'ятовуємо індекси початкової точки та кінцевої точки по діагоналі «nash1» та «kon1»;
- 5) створюємо цикл збільшуючи розміри перешкод;

б) якщо більше немає перешкод, то продовжується виконання програми, якщо є перехід до (1), але цикл починається з «nash1» і «kon1».

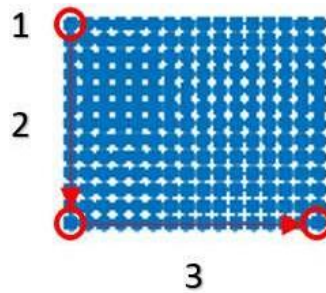


Рисунок 4.5 – Алгоритм збільшення перешкоди

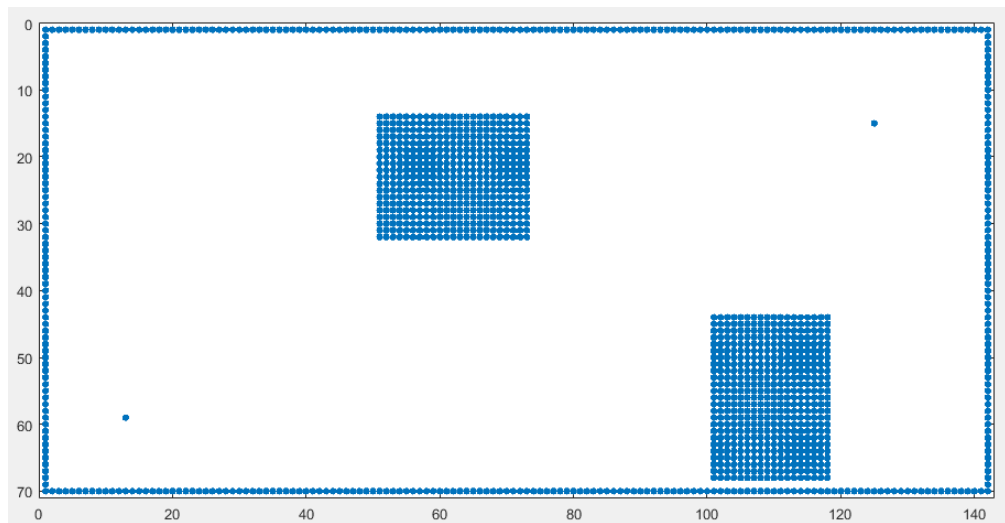


Рисунок 4.6 – Перешкоди, збільшені на 3 комірки з кожної сторони

4.2.4 Хвильовий алгоритм Лі

Мета моделювання маршрутів (трасування) – одна з найбільш складних проблем у загальній проблемі автоматизації проектування. Це з деякими умовами, зокрема з різноманітністю способів конструктивно-технологічної реалізації різних трас, кожному з яких при алгоритмічному постановленню завдання використовуються специфічні аспекти оптимізації та обмеження. З математичної точки зору трасування - це найскладніше завдання підбору оптимального рішення з великою кількістю варіантів.

У двовимірній картатій карті (матриці), що складається з "прохідних" і "непрохідних" клітин, зазначена клітина старту та клітина фінішу. Завдання

методу - продовжити найкоротший шлях з клітини старту до клітини фінішу, якщо це допустимо. Від старту уві сні без винятку напрямку поширюється хвиля, при цьому будь-яка пройдена хвилею клітина позначається як "пройдена". Хвиля, у свою чергу, ніяк не може протікати через клітини, позначені як "пройдені" або "непрохідні".

Хвиля рухається доти, доки дійде місця фінішу чи до того часу, доки не залишиться клітин, яких хвиля не пройшла. Якщо хвиля пройшла всі без винятку доступні клітини, але так і не досягла клітини фінішу, отже, шлях від старту до фінішу провести не можна. Вже після досягнення хвилею фінішу прокладається шлях і зберігається в масиві (згідно з координатами пройдених клітин, у площині).

Введемо припущення - клітини вихідної матриці поставимо квадратними і шлях повинен бути суворо ортогональним.

Опис алгоритму

В нашому випадку із кінцевого елемента хвиля розходиться в 4-х напрямках.

1. Створюємо масив елементів A , що дорівнює розміру поля.
2. Кожному елементу масиву надається деяке значення, залежно від властивостей комірки:
 - A. якщо комірка прохідна, то $A(i,j)=0$;
 - B. якщо комірка непрохідна, то $A(i,j)=-1$;
 - C. якщо комірка - це стартова точка, то $A(i,j) = 1$;
 - D. якщо комірка - це кінцева точка, то $A(i,j) = 2$;
3. Етап поширення хвилі. Вводимо змінну N_i , лічильник ітерацій, і присвоюємо початкове значення 0.
4. Вводимо константу N_k , яку встановлюємо рівною максимально можливому числу ітерацій.
5. Задаємо межі масиву як непрохідні комірки.
6. Рядково переглядаємо робочий масив A .

7. Якщо $A(\text{indstart}(1)-i, \text{indstart}(2)) = 0$, то цій комірці присвоюємо i (крок вгору). Де indstart - це координати стартової точки.

8. Якщо $A(\text{indstart}(1)-i, \text{indstart}(2)+j) = 0$, то цій комірці присвоюємо $i+j$ (крок праворуч).

9. 9. Якщо $A((\text{indstart}(1)-h)-i, \text{indstart}(2)+j) = 0$, то цій комірці присвоюємо $i+j+h$. Де h - це змінна циклу від 1 до вертикального розміру поля (крок вправо і вгору)

10. Якщо $A((\text{indstart}(1)+h)-i, \text{indstart}(2)+j) = 0$, то цій комірці присвоюємо $i+j+h$ (крок праворуч і донизу).

11. Якщо осередок дорівнює -1 , то виходимо з циклу h , і продовжуємо поширення далі.

На рис. 4.7 можна побачити етапи поширення хвилі. Де 1 - крок праворуч і відразу йдуть 2 і 3 - вгору і вниз хвилі. Аналогічно це працює і в інший бік від стартової точки.

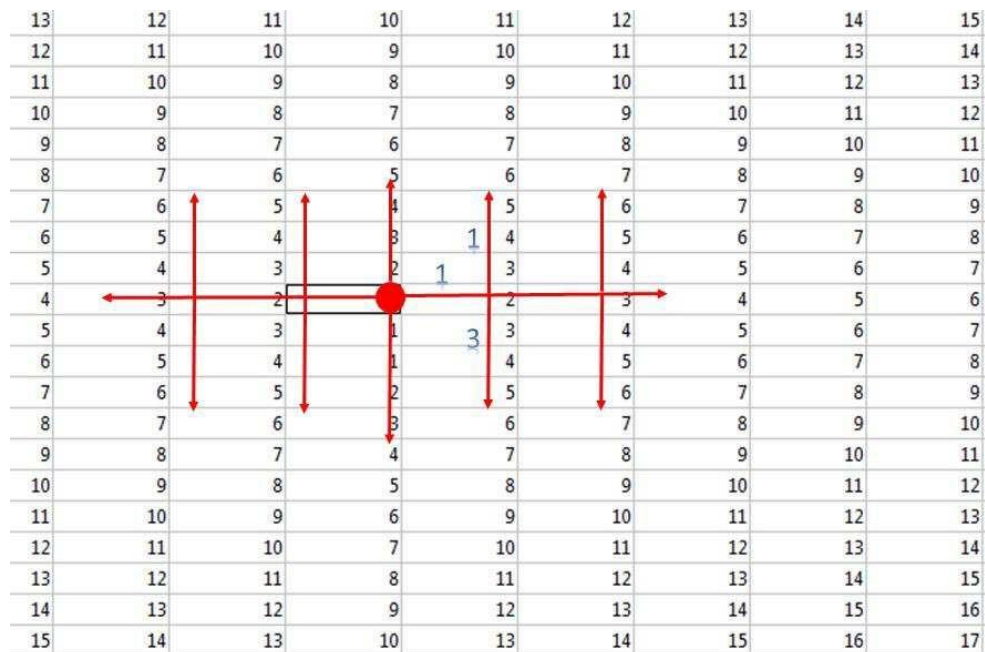


Рисунок 4.7 – Масив А при поширенні хвилі

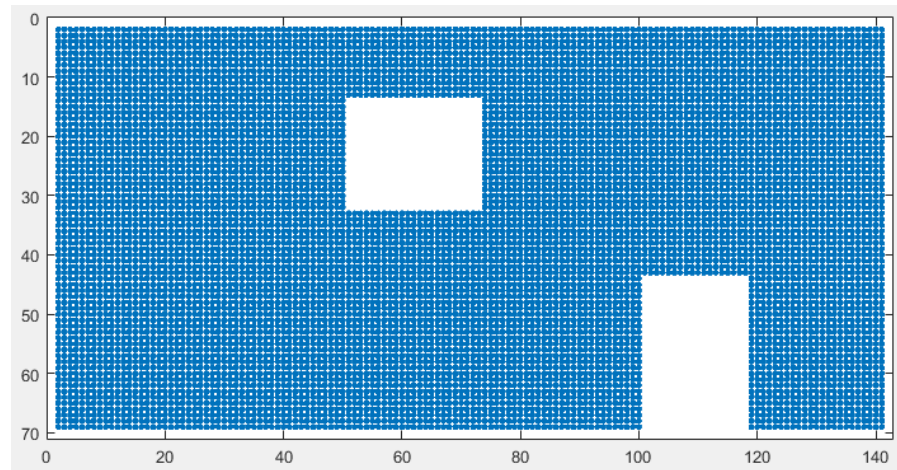


Рисунок 4.8 – Етап, коли хвиля повністю заповнила масив, огинаючи перешкоди

Алгоритм пошуку шляху в масиві А. Побудова траси ведеться від кінцевої до початкової координати.

1. Привласнюємо змінним X та Y значення координат стартової позиції.
2. В околиці позиції A(X,Y) шукаємо елемент із найменшим значенням, тобто переглядаємо значення з 4-х сторін (a,b,c,d).
3. Записуємо значення масив D=[a b c d].
4. Знаходимо найменше значення.
5. Якщо комірка дорівнює 0 або -1, то привласнюємо їй значення 999, щоб це число не виявилось найменшим.
6. Робимо крок у бік найменшого значення і зберігає координати комірок у масив. Так само для наочності присвоюємо цій комірці 0.
7. Якщо число ітерацій більше 800, відбувається вихід із підпрограми, оскільки шлях продовжити виявилось неможливо.

158	159	160	161	162
157	158	159	160	161
156	157	158	159	160
155	156	157	158	159
154	155	155	157	158
153	154	155	156	157
152	153	154	155	156
151	152	153	154	155
150	151	152	153	154
149	150	151	152	153

Рисунок 4.9 – Процес пошуку напрямку шляху із кожним кроком

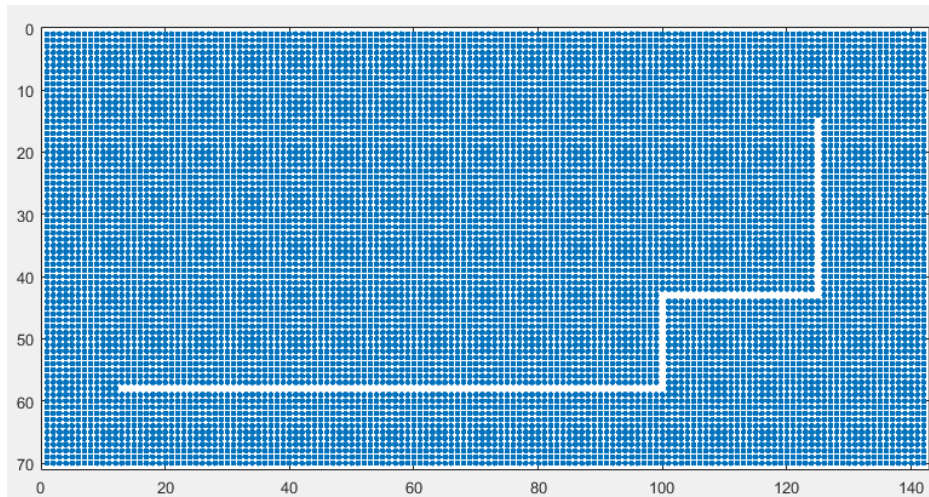


Рисунок 4.10 – Шлях в середовищі із перешкодами

4.3 Керуючі команди для мобільного робота

За знайденим шляхом формується список керуючих команд. Цей список складається з команд, що чергуються "їхати прямо x обертів" і "повернути ліворуч/праворуч".

Алгоритм.

1. Маючи матрицю з індексами переміщень визначаємо напрямок руху (вгору, вниз, вправо, вліво). Якщо поточний індекс комірки дорівнює наступному, то записуємо значення в нову матрицю way:

- A. індекс 8 - якщо рух нагору;
- B. індекс 5 – якщо рух вниз;
- C. індекс 4 - якщо рух ліворуч;
- D. індекс 6 - якщо рух праворуч.

2. Знаходимо точки поворотів, перебираючи всі значення матриці way. Якщо поточний елемент не дорівнює наступному, то, отже, стався поворот. Вираховуємо з поточного значення наступні в точці переривання послідовності і отримуємо одне з 4-х значень, і робимо висновок у який бік відбувся поворот.

3. Записуємо значення для керуючих команд у матрицю W. Наприклад: [153 4 67 6 52 4 27]. Це говорить про те, що рух прямо 153 комірки, поворот вліво, рух прямо 67 комірок і так далі.

4. Так як кількість осередків не дорівнює відстані на підлозі, знаходимо коефіцієнт для імпульсів енкодера практичним шляхом, методом спроб і помилок.

4.4 Рішення оберненої задачі кінематики маніпулятора

Практичний інтерес має обернена задача кінематики (ОЗК), що полягає у визначенні необхідних узагальнених координат по заданому бажаному положенню схвату. Очевидно, що це завдання не має однозначного рішення, оскільки може існувати безліч комбінацій узагальнених координат, що забезпечують задане положення схвату. Існують різні методи рішення ОЗК, до яких належать методи обернених перетворень, гвинтової алгебри, двоїстих матриць, двоїстих кватерніонів, ітерацій, геометричний підхід.

Рішення ОЗК відбувається у пакеті Matlab – Robotics Toolbox за допомогою пакету функції *ikine*, заснованої на представленні Денавіта-Хартенберга.

Для опису обертальних та поступальних зв'язків між сусідніми ланками Денавіт та Хартенберг запропонували матричний метод послідовної побудови систем координат, пов'язаних з кожною ланкою кінематичного ланцюга. Сенс представлення Денавіта-Хартенберга (ДХ-представлення) полягає у формуванні однорідної матриці перетворення, що має розмірність 4×4 і описує положення системи координат кожної ланки щодо системи координат попередньої ланки. Це дає можливість послідовно перетворити координати схвата маніпулятора з системи відліку, пов'язаної з останньою ланкою, у базову систему відліку, що є інерційною системою координат для аналізованої динамічної системи.

Кожна система координат формується на основі наступних трьох правил:

1. Вісь z_{i-1} спрямована вздовж осі i -го зчленування.
2. Вісь x_i перпендикулярна до осі z_{i-1} і направлена від неї.
3. Вісь y_i доповнює осі x_i , z_i до правої декартової системи координат.

ДХ-представлення твердих ланок залежить від чотирьох геометричних параметрів, що відповідають кожній ланці. Ці чотири параметри повністю описують будь-який обертальний чи поступальний рух і визначаються відповідно до рис. 4.4 наступним чином:

Θ_i - приєднаний кут, на який треба повернути вісь x_{i-1} навколо осі z_{i-1} , щоб вона стала сонапрявлена з віссю x_i (знак визначається відповідно до правила правої руки);

d_i - відстань між перетином осі z_{i-1} з віссю x_i і початком $(i-1)$ -ї системи координат, що відраховується вздовж осі z_{i-1} ;

α_i - лінійне зміщення - відстань між перетином осі z_{i-1} з віссю x_i і початком i -ї системи координат, що відраховується вздовж осі x_i , тобто найкоротша відстань між осями z_{i-1} і z_i ;

a_i - кутове зміщення - кут, на який треба повернути вісь z_{i-1} навколо осі x_i , щоб вона стала сонапрявленою з віссю z_i (знак визначається відповідно до правила правої руки).

Для обертальних зчленувань параметри d_i , α_1 та α_i є характеристиками зчленування, постійними даного типу робота. У той же час, Θ_i є змінною величиною, що змінюється під час руху (обертання) i -ї ланки щодо $(i-1)$ -ї.

Після побудови ДХ-координат для всіх ланок можна побудувати однорідні матриці перетворення, що зв'язують i -ю і $(i-1)$ -ю систему координат:

$${}^{i-1}\mathbf{A}_i = \mathbf{T}_{z,d} \mathbf{T}_{z,\Theta} \mathbf{T}_{x,\alpha} \mathbf{T}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos Q_i & -\sin Q_i & 0 & 0 \\ \sin Q_i & \cos Q_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \quad (4.1)$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos Q_i & -\cos \alpha_i \sin Q_i & \sin \alpha_i \sin Q_i & a_i \cos Q_i \\ \sin Q_i & \cos \alpha_i \cos Q_i & -\sin \alpha_i \cos Q_i & a_i \sin Q_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Перетворюючи (4.1), знайдемо, що матриця, обернена до ${}^{i-1}\mathbf{A}_i$, має вигляд:

$$[{}^{i-1}\mathbf{A}_i]^{-1} = {}^i\mathbf{A}_{i-1} = \begin{bmatrix} \cos \theta_i & \sin \theta_i & 0 & -a_i \\ -\cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & \sin \alpha_i & -d_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & -\sin \alpha_i \cos \theta_i & \cos \alpha_i & -d_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.2)$$

де α_i , a_i , d_i – константи, а θ_i – приєднувальна змінна, якщо розглянуте зчленування – обертальне.

Використовуючи матрицю ${}^{i-1}\mathbf{A}_i$, можна зв'язати однорідні координати \mathbf{p}_i точки \mathbf{p} щодо i -ї системи координат (точка \mathbf{p} лежить в i -й системі координат) з односторонніми координатами цієї точки щодо $(i-1)$ -ї системи відліку, пов'язаної з $(i-1)$ -ю ланкою. Цей зв'язок встановлюється рівністю:

$$\mathbf{p}_{i-1} = \mathbf{A}_i \mathbf{p}_i^{i-1}, \quad (4.3)$$

где $\mathbf{p}_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1}, 1)^T$ и $\mathbf{p}_i = (x_i, y_i, z_i, 1)^T$.

В Robotic toolbox створюємо модель, пропорційну реальному об'єкту:

L(1) = Link('d',0, 'a',4, 'alpha',-pi/2);

L(2) = Link('d',0, 'a',16, 'alpha',0);

L(3) = Link('d',0, 'a',20, 'alpha',0);

L(4) = Link('d',0, 'a',4, 'alpha',0);

L(5) = Link('d',0, 'a',0, 'alpha',0).

Задаємо початкове положення ланок:

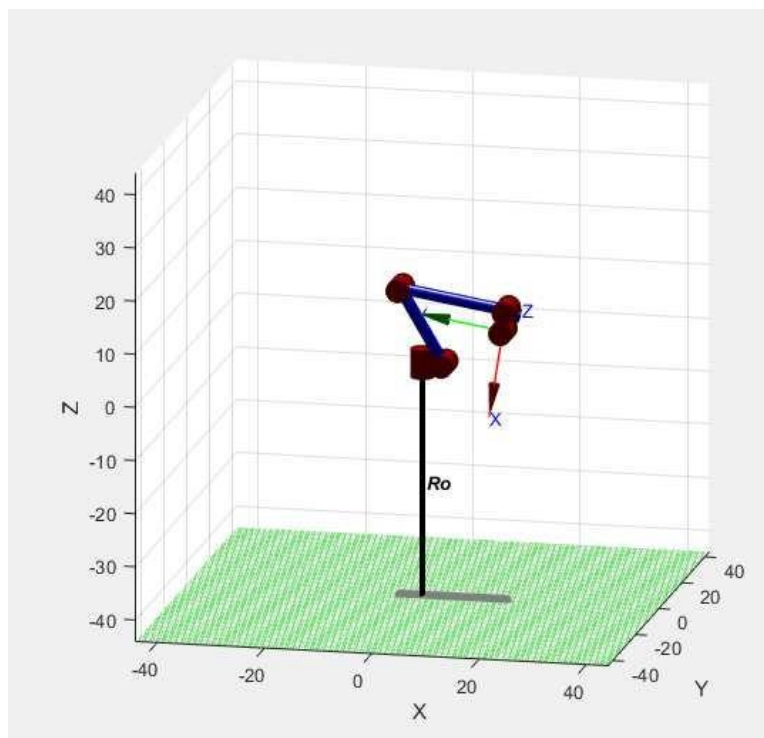


Рисунок 4.11 – Початкове положення ланок

Використовуючи функцію `ikine`, знайдемо нове положення ланок, знаючи координати положення об'єкта.

Необхідно перемістити схват в точку $[21 \ -11 \ -12]$.

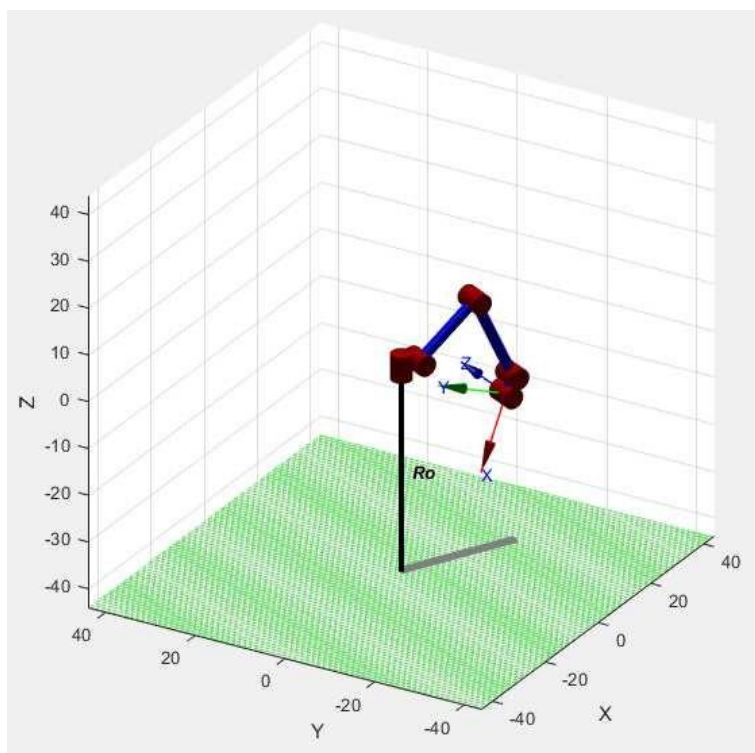


Рисунок 4.12 – Кінцеве положення ланок маніпулятора

Отримуємо кути: $f_i = [118\ 98\ 123]$.

Тепер дані про кути можна відправити на мікроконтролер для виконання переміщення.

4.5 Висновки за розділом

Відповідно до поставленого завдання управління мобільним роботом у середовищі з перешкодами були визначені алгоритми, що здійснюють аналіз зображення робочої зони робота, пошук найкоротшого шляху від положення робота до заданої мети і управління роботом на горизонтальній місцевості з перешкодами.

5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА – РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Опис функціоналу програмного забезпечення

Мобільний застосунок було розроблено у середовищі програмування Xamarin мовою програмування C# і розрахована для роботи з бездротовим інтерфейсом мобільного робота.

На основі алгоритмів, визначених у розділах 2-4, було складено такі функції (програми):

- функція отримання зображення із камери;
- функція пошуку координати робота;
- функція збільшення розміру перешкоди у матриці;
- функція поширення хвилі у матриці з перешкодами;
- функція пошуку найкоротшого шляху в середовищі з перешкодами;
- функція підрахунку кількості кроків та напрямки мобільного робота;
- функція формування керуючих команд та передача бездротового зв'язку;
- функція пошуку об'єкта для маніпулювання та рішення ОЗК маніпулятора.

Більш детально розглянемо лише деякі з них, тому що у розділі 4 були описані основні етапи та функції для обробки матриці та пошуку шляху середовищі з перешкодами.

5.2 Функція отримання зображення з камери

Програма захоплення зображення з камери міститься у дод.А. При запуску їй не потрібно прочитувати файли, тому що з цієї функції починається вся програма управління роботом.

Для використання OpenCV для захоплення зображень потрібно реалізувати кроки, описані в таблиці 5.1.

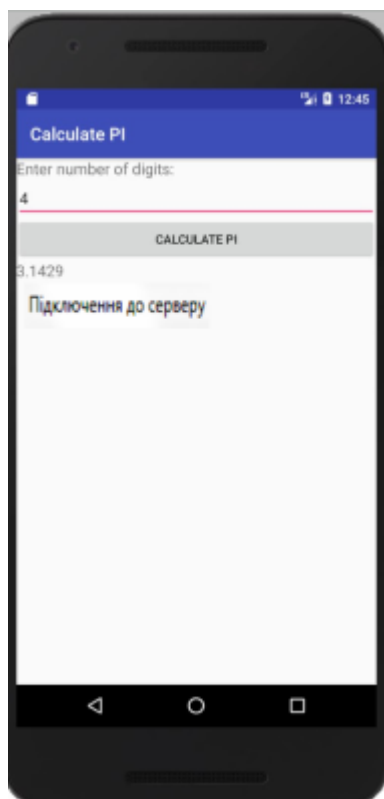


Рисунок 5.1- Вікно початкового додатка

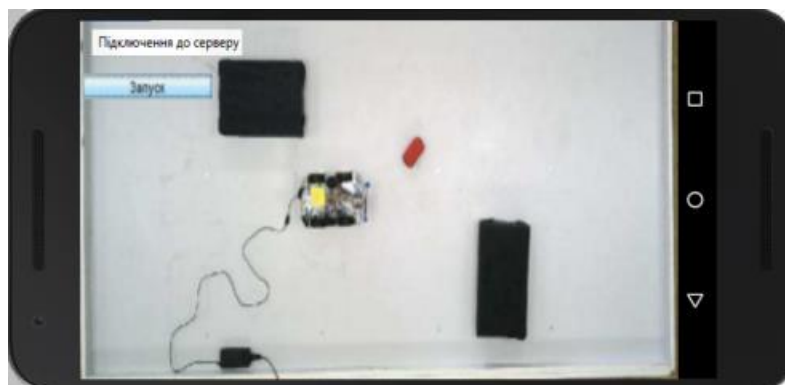


Рисунок 5.2 – Вікно після натискання кнопки «підключення до сервера»

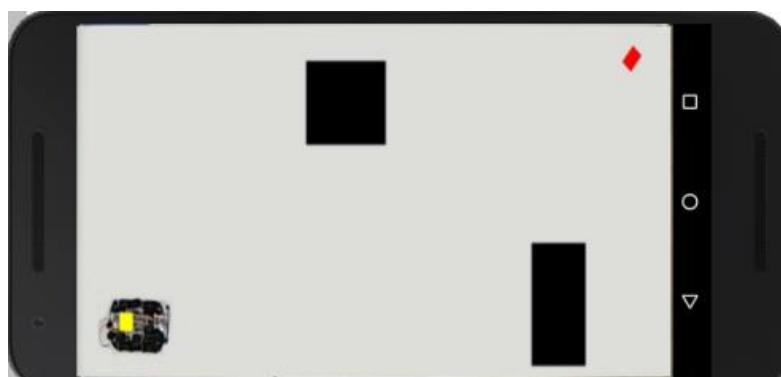


Рисунок 5.3 – Робоча зона

Таблиця 5.1 – Кроки для захоплення зображень

Крок 1	Інсталяція та конфігурація пристроїв захоплення зображень
Крок 2	Отримання інформації, яка однозначно ідентифікує пристрій захоплення зображення в OpenCV
Крок 3	Створення вихідних об'єктів відеопослідовності
Крок 4	Попередній перегляд відеопослідовності (необов'язково)
Крок 5	Формування властивостей об'єктів захопленого зображення (необов'язково)
Крок 6	Захват даних зображення
Крок 7	Очищення даних

Далі розглянемо кожен крок більш детально.

Крок 1. Інсталяція та конфігурація пристроїв захоплення зображень

Для встановлення пристрою захоплення зображень слід користуватися інструкцією установки, яка містить такі типові кроки:

- встановлення плати захоплення зображень на комп'ютер;
- встановлення необхідних програмних драйверів для цього пристрою. Вони поставляються разом із пристроєм захоплення зображень;
- приєднання камери до плати захвату зображень;
- перевірка роботи камери, її характеристик, запуск відповідного програмного забезпечення та перегляд отриманої відеопослідовності.

Типовий пристрій захоплення зображень, такі як web-камери або портативні камери не вимагають встановлення плати захоплення зображень. Ці пристрої підключаються безпосередньо до комп'ютера через USB або FireWire-порт.

Крок 2. Отримання інформації, яка однозначно ідентифікує пристрій захоплення зображення в OpenCV

На цьому кроці необхідно обрати камеру із доступних та, за необхідності, налаштувати її параметри.

Крок 3. Створення вихідних об'єктів відеопослідовності

На цьому кроці створимо вихідні об'єкти відеопослідовності, які використовуються програмою для подання зв'язку між мобільним додатком та пристроєм захоплення зображення. Використовуючи властивості вихідних об'єктів відеопослідовності можна контролювати багато аспектів процесу захоплення зображень.

Крок 4. Попередній перегляд відеопослідовності (необов'язково)

Після створення вихідних об'єктів відеопослідовності мобільний застосунок має доступ до пристроїв захоплення зображення та може готувати захоплені дані. Проте, спочатку можна провести попередній перегляд вихідних об'єктів відеопослідовності та переконатися у прийнятності результату. Наприклад, можна змінити розташування камери, змінити освітлення тощо.

Крок 5. Формування властивостей об'єктів захопленого зображення (необов'язково)

Після створення вихідних об'єктів відеопослідовності та попереднього перегляду потокового відео існує можливість модифікації характеристик зображення або інших аспектів процесу захоплення. Це можна досягти, змінюючи властивості об'єктів захоплення зображень. У цьому розділі наведено:

- опис типів об'єктів захоплення зображень, що використовуються у додатку;
- опис того, як переглядати всі характеристики цих об'єктів та їх поточні значення;
- опис того, як встановити значення властивостей об'єктів.

Крок 6. Захват даних зображення

Після створення вихідних об'єктів відеопослідовності та формування їх властивостей можна розпочинати захоплення даних. Типові етапи захоплення зображень такі:

- 1) запуск відеопослідовності;
- 2) запуск захоплення;
- 3) передача даних у робочий простір мобільного застосунку.

5.3 Функція підрахунку кількості кроків і напрямки мобільного робота

При пошуку шляху програма запам'ятовувала індекс кожного пройденого осередку в масиви S - значення рядка і Sh - значення стовпця. Далі представлений алгоритм підрахунку кількості кроків у матриці та напрямки.

Алгоритм.

1. Задаємо цикл від 1 до довжини масиву $S-1$.
2. Якщо $Sh(i)$ дорівнює $Sh(i+1)$ і $S(i)$ менші за $S(i+1)$, значить відбувається рух вгору по матриці та i -ому елементу масиву way присвоюємо значення 8.
3. Якщо $Sh(i)$ дорівнює $Sh(i+1)$ і $S(i)$ більше за $S(i+1)$, значить відбувається рух вниз по матриці і i -ому елементу масиву way присвоюємо значення 5.
4. Якщо $S(i)$ дорівнює $S(i+1)$ і $Sh(i)$ менше $Sh(i+1)$, значить відбувається рух ліворуч по матриці та i -ому елементу масиву way присвоюємо значення 4.
5. Якщо $S(i)$ дорівнює $S(i+1)$ і $Sh(i)$ більше $Sh(i+1)$, значить відбувається рух вправо по матриці та i -ому елементу масиву way присвоюємо значення 6.

У такий спосіб ми отримали масив way з індексами, що показують у який бік по матриці потрібно рухатись роботу, щоб досягти ділі.

Кількість кроків визначається довгою масивом way .

5.4 Функція формування керуючих команд і передача по бездротовому зв'язку

Після підрахунку кількості кроків необхідно знайти точки для робота в реальному світі, тому що робот не має команди "рух ліворуч", він повинен отримати команду "поворот ліворуч" та "рух прямо n-імпульсів". Для реалізації цього завдання наведено нижче алгоритм.

1. Задаємо цикл від 1 до довжини масиву way-1.
2. Якщо $(way(i) \neq way(i+1))$ і $((way(i+1)) - (way(i)))$ дорівнює -1 (або -2, або -4), то це поворот направо і задаємо індекс 6 у точці i. Наприклад: значення матриці way наступні "6666655555" це говорить про те, що робот рухався вправо, потім змінив свій рух і продовжив рух вниз. Якщо з цифри 5 відняти 6, отримуємо -1, що говорить про поворот праворуч.
3. Якщо $(way(i) \neq way(i+1))$ і $((way(i+1)) - (way(i)))$ дорівнює 2 (або -4, або 1), то це поворот наліво і задаємо індекс 4 у точці i.

Перейдемо до складання масиву в якому міститься інформація про кількість імпульсів та повороти.

Масив із даними виглядає так: $[n \ p \ n \ p \dots]$,

де n - число імпульсів, помножене на коефіцієнт 1.1, p - поворот праворуч або ліворуч.

Тепер у нас є команди, що управляють, але може статися таке, що число непотрібних поворотів або руху буде неефективним, тобто число імпульсів менше 11 (на практиці це 2см). Відкидаємо ці значення та отримуємо масив S1 з готовими до відправлення керуючими командами.

Алгоритм передачі даних на сервер.

1. Створення циклу i від 1 до кількості елементів в масиві S1.
2. Використання функції rem - для дійсних чисел функція $rem(i,2)$ обчислює залишок від поділу i на 2. На виході виходить масив hd, що з 0 і 1.
3. Створення TCP об'єкта та встановлення номера порту сервера $t=tcip('192.168.4.1', 80, 'NetworkRole', 'client')$.
4. Активація з'єднання з портом.

5. Якщо $hd(i)$ дорівнює 1, то йде перевірка: якщо кількість імпульсів більше 100, то присвоюємо значення "1". Далі створюється команда передачі у порт ($f=[8 \text{ o } ((s1(i))-(o*100))]$).
6. Переклад із типу даних `integer` в `char`.
7. Передача даних на сервер.
8. Якщо $hd(i)$ дорівнює 0, ($f=[s1(i) \text{ 0 } 30]$). На цьому етапі відбувається передача сигналу про напрямок повороту.
9. Переклад із типу даних `integer` в `char`.
10. Передача даних на сервер.
11. Пауза
12. Зупинка сервера та від'єднання TCP об'єкта від сервера.

5.5 Функція пошуку об'єкта для маніпулювання і рішення оборотної задачі кінематики маніпулятора

Після того, як робот виконав керуючі команди, він прибув до точки, що знаходиться поряд з об'єктом для маніпулювання. Далі необхідно обчислити координати x , y , z на вирішення ОЗК.

Алгоритм пошуку координат об'єкта для маніпулювання.

1. З масиву `way` виділяємо останній елемент Hg і визначаємо, з якого боку робот під'їхав до об'єкта.
2. Викликаємо функцію `myfun.m` для повторного знімка. Дізнаємося координати робота - , і координати об'єкта.
3. Для пошуку координат об'єкта щодо робота використовуємо
4. Якщо Hg дорівнює 6 або 4:
 - $x = end0(2) - indend(2) + 5$;
 - якщо $end0(1) > indend(1)$, то $y = end0(1) - indend(1) + 5$;
 - якщо $indend(1) > end0(1)$, то $y = end0(1) - indend(1) - 5$;
 - якщо $(-10 < (indend(1) - end0(1)) < 10)$, то $y = 0$;
 - $z = -10$, це пов'язано із конструкцією робота.
5. Якщо Hg дорівнює 6 або 4:

- $x = \text{indend}(1) - \text{end0}(1) + 5$;
 - якщо $\text{end0}(2) > \text{indend}(2)$, то $y = \text{end0}(2) - \text{indend}(2) + 5$;
 - якщо $\text{indend}(2) > \text{end0}(2)$, то $y = \text{end0}(2) - \text{indend}(2) - 5$;
 - якщо $(-10 < (\text{indend}(1) - \text{end0}(1)) < 10)$, то $y = 0$;
 - $z = -10$, це пов'язано із конструкцією робота.
6. Якщо $x < 20$, необхідно від'їхати назад 15 імпульсів.
 7. Якщо $y < 20$, необхідно виконати поворот ліворуч і проїхати вперед 11 імпульсів.
 8. Якщо $y > 20$, необхідно виконати поворот праворуч і проїхати вперед 11 імпульсів.

5.6 Висновки за розділом

Програми, засновані на алгоритмах, визначених у розділах 2, 4, показали відповідність алгоритмів поставленим завданням, однак експериментальні дослідження вказали на необхідність урахування помилки позиціонування при пошуку координати об'єкта маніпулювання. Необхідні додаткові датчики, оскільки зображення з камери не може дати чіткого та точного зображення (похибка залежить від кута між перпендикуляром від камери та підлогою, та розташування робота в даний момент часу).

6 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Відповідно до технічного завдання на дипломне проєктування для дослідницьких робіт було придбано необхідне обладнання та програмне забезпечення. У цьому розділі проводиться розрахунок кошторису витрат, пов'язаних із їх придбанням та обслуговуванням.

Кошторис витрат - це вираження у грошовій формі витрат підприємства на розробку та виконання господарсько-договірної роботи.

Кошторис витрат складатиметься з витрат на:

- основне обладнання та програмне забезпечення;
- заробітну плату;
- амортизацію обладнання;
- оренду приміщення.

Далі подано розрахунок всіх складових кошторису витрат.

6.1 Розрахунок витрат на обладнання та програмне забезпечення

Вартість витрат на придбання компонентів та програмного забезпечення обчислимо за формулою:

$$S_z = S_{об} + S_{npos}, \quad (6.1)$$

де $S_{об}$ – вартість витрат на придбання компонентів;

S_{npos} – витрати на придбання програмного забезпечення.

Витрати на придбання програмного забезпечення наведені в таблиці 6.1, витрати на придбання обладнання – в таблиці 6.2.

Таблиця 6.1 – Вартість програмного забезпечення

Назва	Вартість, грн
Microsoft Windows 7 Home Premium	1746
Microsoft Office 2016	5240
MATLAB and Simulink Student Suite	900
SolidWorks 2018 Standard	90000
Всього:	97886

Таблиця 6.1 – Вартість компонентів

Назва	К-ть, шт.	Ціна за шт., грн	Сума, грн
Платформа для збору роботів, , 4WD, 2 рівні	1	50	50
Сервопривід TowerPRO MG995	3	95	465
Сервопривід MG90S	1	45	45
Wi-fi модуль NodeMcu v3Lua	1	45	45
Arduino Mega 2560	1	85	85
Три осьовий гіроскоп GY-521	1	66	66
Лінійний стабілізатор напруги L7805	1	71	71
Диск із прорізами для оптичного енкодера й енкодер	1	48	48
Блок живлення 12 Вольт	1	87	87
Веб-камера Sven IC-975HD	1	98	98
ПК HP Pavilion g-6 /6.0Gb/500Gb/1024Mb	5	90	450
Всього:	17	780	1412

Загальна вартість витрат на придбання компонентів та програмного забезпечення дорівнює:

$$\underline{S_3 = 780 + 1412 = 2192} \quad (6.2)$$

6.2 Розрахунок фонду заробітної плати розробникам

У період розробки установки оклад ведучого інженера $S_{інж}$ складає 1687 грн. в місяць.

Визначимо денну заробітну платну ведучого інженера:

$$S_{інж.} = S_{інж.} / N, \quad (6.3)$$

де $N = 21$ – кількість робочих днів у місяці. Тоді:

$$S_{інж.} = \underline{1687/21=80,3}, \quad (6.4)$$

Визначимо основну заробітну плату керівника дипломного проекту:

$$S_{кер.} = S_{кер.} \cdot n, \quad (6.5)$$

де $n = 84$ – кількість робочих днів за 4 місяці роботи. Тоді:

$$S_{кер.} = \underline{80,3 \cdot 84 = 6745,2}, \quad (6.6)$$

Додаткова заробітна плата:

$$S_{дод.з.п.} = S_{дод.з.п.} \cdot k_{дзп}, \quad (6.7)$$

де $k_{дзп} = 0,2$ – коефіцієнт додаткової заробітної плати. Тоді:

$$S_{дод.з.п.} = \underline{6745,2 \cdot 0,2 = 1349,04}, \quad (6.8)$$

Розмір надбавки до заробітної плати з урахуванням районного коефіцієнта:

$$S_{з.п.над.} = (S_{кер.} + S_{дод.з.п.}) \cdot k_{рк}, \quad (6.9)$$

де $k_{рк} = 0,3$ – районний коефіцієнт для місто. Тоді:

$$S_{з.п.над.} = \underline{(6745,2 + 1349,04) \cdot 0,3 = 2428,272} \quad (6.10)$$

Повністю фонд заробітної плати всіх робітників без податків:

$$S_{заг.} = (S_{кер.} + S_{дод.з.п.} + S_{з.п.над.}) \cdot a_p, \quad (6.11)$$

де a_p – кількість робітників. Тоді:

$$S_{заг.} = \underline{(6745,2 + 1349,04) \cdot 0,3 = 2428,272}, \quad (6.12)$$

Після визначення загального розміру заробітної плати здійснимо розрахунок відрахувань на податки. Вони беруться від фонду оплати праці (суми основної та додаткової заробітної плати) у розмірі 19,5%, з них:

- 18% ПДФО;
- 1,5% військовий збір;

Таким чином,

$$S_{соц} = R_{соц} \cdot (S_{кер.} + S_{дод.з.п.}), \quad (6.13)$$

де $R_{\text{соц}} = k_{\text{ПДФО}} + k_{\text{ВЗ}}$

$k_{\text{ПДФО}} = 0.18$ – ПДФО:

$k_{\text{ВЗ}} = 0.015$ – військовий збір. Тоді:

$$S_{\text{соц}} = (0,18+0,015) \quad (6.14)$$

Всього фонд оплати праці з податками:

$$S_{\text{под.}} = S_{\text{соц}} + S_{\text{заг}} \quad (6.15)$$

$$S_{\text{соц}} = (0,18+0,015) \quad (6.15)$$

6.3 Витрати на електроенергію

Вихідні дані для розрахунку вартості електроенергії:

- потужність, що споживається ПК з монітором – 200вт
- потужність, що споживається лампою денного світла -18вт
- тариф на електроенергію -2,64 грн/кВт-год
- кількість ламп у приміщенні –92шт

Витрати на електроенергію в день:

$$S_{\text{ел.д.}} = (P_{\text{ПК}} + P_{\text{л}} \cdot n) \cdot N_{\text{д}} \cdot T_{\text{ел.}} \cdot N_{\text{г}} \quad (6.16)$$

де $P_{\text{ПК}}$ – потужність, що споживається ПК з монітором;

$P_{\text{л}}$ – потужність, що споживається лампою денного світла;

n – кількість ламп денного світла;

$T_{\text{ел.}}$ – тариф на електроенергію;

$N_{\text{д}}$ – кількість робочих днів;

$N_{\text{г}}$ – години робочого дня.

$$S_{\text{ел.д.}} = S_{\text{соц}} = (0,18+0,015)+86+967=1230 \quad (6.16)$$

6.4 Витрати на амортизацію обладнання

Для розрахунку витрат за амортизацію обладнання необхідно визначити час експлуатації кожного з елементів обладнання. Для роботи з налаштування та програмування робота, а також для складання додаткового програмного забезпечення використовувався ПК.

Таким чином, сумарні витрати на амортизацію визначаються за такою формулою:

$$Z_A = Z_{AK}, \quad (6.17)$$

де Z_{AK} – витрати на амортизацію комп'ютера.

Визначимо витрати на амортизацію комп'ютера за формулою:

$$Z_{AK} = C_K \cdot n_T / n_i \cdot 12 \quad (6.18)$$

де C_K – початкова вартість обладнання;

$n_T = 4$ місяці – час використання;

$n_i = 5$ – строк служби. Тоді:

$$Z_{AK} = 500 \quad (6.19)$$

6.5 Розрахунки на оренду приміщення

Витрати на оренду приміщення за час розробки системи склали:

$$Z_{OP} = Z_{O1} \cdot P \cdot T_p, \quad (6.24)$$

де Z_{O1} – вартість оренди за 1м² площі;

P – площа приміщення;

$T_p = 4$ місяці – час етапу розробки. Тоді:

$$Z_{OP} = 50 \cdot 51,6 \cdot 4 = 20320 \quad (6.25)$$

6.6 Кошторис витрат

Усі витрати, пов'язані з придбанням та використанням обладнання та програмного забезпечення, амортизацію, оплату праці та оренду приміщення, наведено у таблиці 6.3 та показано у співвідношенні на малюнку 6.1.

Таблиця 6.3 – Кошторис витрат

Назва витрат	Сума витрат, грн
Витрати на програмне забезпечення та обладнання	42
Фонд оплати праці	50
Витрати на амортизацію обладнання	70

Розрахунки на оренду приміщення	81
Витрати на електроенергію	225
Всього:	

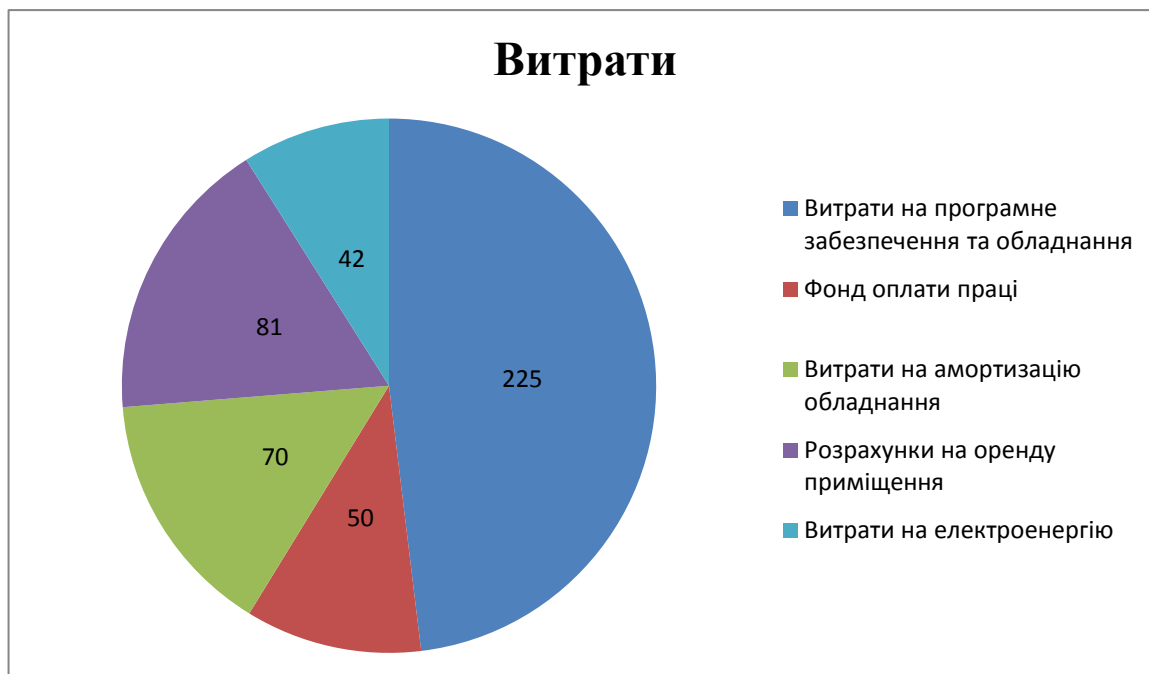


Рисунок 6.1 – Діаграма витрат

6.7 Висновки за розділом

В даному розділі визначено кошторис витрат на дипломне проектування для дослідницьких робіт. Як видно із таблиці 7.3, більша частина витрат зумовлена

ВИСНОВОК

У ході дипломного проекту було спроектовано та розроблено мобільний маніпуляційний робот. Була побудована 3Д модель маніпулятора, з урахуванням усіх довжин і розмірів деталей, визначення її конструктивних обмежень, саме діапазонів кутів поворотів ланок.

Всі датчики підключені до мікроконтролера, пов'язаного бездротовим зв'язком з ПК, напруга подається від блока живлення на спроектовану плату, потужності достатньо нормального функціонування всіх пристроїв і датчиків.

Отримуючи зображення з камери, програма, написана мовою програмування C#, розпізнає образи, знаходить координати робота, об'єкта для маніпулювання та перешкод. Далі складаються керуючі команди для управління роботом і передаються бездротовим зв'язком на мобільний робот. Робот, отримавши команди, починає послідовно їх виконувати. Під'їхавши до об'єкта відбувається повторний знімок, також розпізнання координат і далі маніпулятор, переміщуючи свої ланки намагається взяти об'єкт.

Експерименти підтвердили адекватність та ефективність обраних методів. Поставлені цілі та завдання були виконані, але є похибка у вирішенні ОЗК маніпулятора (не кожного разу схват точно бере об'єкт).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кулік, А. С. Методи моделювання об'єктів автоматичного управління [Текст]: навч. посібник / А. С. Кулік, С. Н. Пасичник. – Харків: Нац. аерокосм. ун-т ім. М.Є. Жуковського «Харків. авіац. інт-т», 2018. – 168 с.
2. Надійність автоматизованих систем керування. Навчальний посібник для вузів. Под ред. Хетагурова Я.А
3. An indoor mobile robot navigation technique using odometry and electronic compass [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://journals.sagepub.com/doi/full/10.1177/1729881417711643>
4. Building a Magnetic Track Guided AGV [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://www.robotiq.com/applications/allblogs/18-building-a-magnetic-track-guided-agv>.
5. Kam M. Sensor Fusion for Mobile Robot Navigation / M. Kam, X. Zhu, P. Kalata // Proceedings of the IEEE Vol. 85, Issue: 1 / M. Kam, X. Zhu, P. Kalata., 1997. – С. 108–119.
6. SIEGWART R. Mobile Robot Kinematics / R. SIEGWART, I. NOURBAKHSI // Introduction to Autonomous Mobile Robots / R. SIEGWART, I. NOURBAKHSI., 2004. – (Massachusetts Institute of Technology). – С. 47–82
7. Dudek G. Computational Principles of Mobile Robotics / G. Dudek, M. Jenkin. – London: Cambridge University Press, 2000. – 406 с.
8. Gonzalez-Banos H. Robot Navigation for Automatic Model Construction Using Safe Regions / H. Gonzalez-Banos, J.-C. Latombe // Lecture Notes in Control and Information Sciences / H. Gonzalez-Banos, J.-C. Latombe. – Berlin: Springer, 2001. – С. 405–415
9. Tikhonov A. N. Solutions of Ill-posed Problems / A. N. Tikhonov, V. Y. Arsenin., 1977. – 258 с. – (the University of Michigan).
10. Thielscher M. Reasoning Robots The Art and Science of Programming Robotic Agents / Thielscher. – Netherlands: Springer, 2005.
11. Joint element description [Електронний ресурс]. – 2014. – Режим

доступу до ресурсу: <http://wiki.ros.org/urdf/XML/joint>.

12. Розробка Android-додатків на C# з використанням Xamarin з нуля" - Євген Умріхін

13. Об'єктно-орієнтоване проектування систем керування [Текст] : навч. посіб. / Л. О. Краснов, О. В. Гавриленко. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2020. – 184 с.

ДОДАТОК А

```

#include <Servo.h>

Servo servo;

// Ultrasonic Module pins

const int trigPin = 13; // 10 microsecond high pulse causes chirp , wait 50 us

const int echoPin = 12; // Width of high pulse indicates distance

// Servo motor that aims ultrasonic sensor .

const int servoPin = 11; // PWM output for hobby servo

// Motor control pins : L298N H bridge

const int enAPin = 6; // Left motor PWM speed control

const int in1Pin = 7; // Left motor Direction 1

const int in2Pin = 5; // Left motor Direction 2

const int in3Pin = 4; // Right motor Direction 1

const int in4Pin = 2; // Right motor Direction 2

const int enBPin = 3; // Right motor PWM speed control

enum Motor { LEFT, RIGHT };

// Set motor speed: 255 full ahead, -255 full reverse , 0 stop

void go( enum Motor m, int speed)

{

digitalWrite (m == LEFT ? in1Pin : in3Pin , speed > 0 ? HIGH : LOW );

digitalWrite (m == LEFT ? in2Pin : in4Pin , speed <= 0 ? HIGH : LOW );

analogWrite(m == LEFT ? enAPin : enBPin, speed < 0 ? -speed : speed );

} // Initial motor test :

// left motor forward then back

// right motor forward then back

/*void testMotors ()

{

static int speed[8] = { 128, 255, 128, 0 ,

-128, -255, -128, 0};

go(RIGHT, 0);

```

```

for (unsigned char i = 0 ; i < 8 ; i++)

go(LEFT, speed[8]), delay (200);

for (unsigned char i = 0 ; i < 8 ; i++)

go(RIGHT, speed[1]), delay (200);

}

*/

unsigned int readDistance ()

{

digitalWrite ( trigPin , HIGH );

delayMicroseconds (10);

digitalWrite ( trigPin , LOW );

unsigned long period = pulseIn ( echoPin, HIGH );

return period * 343 / 2000;

}

#define NUM_ANGLES 14

unsigned char sensorAngle[NUM_ANGLES] = { 20,30, 40, 50, 60, 70, 80, 90, 100, 110, 120,130,140,
150 };

unsigned int distance [NUM_ANGLES];

// Scan the area ahead by sweeping the ultrasonic sensor left and right

// and recording the distance observed. This takes a reading , then

// sends the servo to the next angle. Call repeatedly once every 50 ms or so.

void readNextDistance ()

{

static unsigned char angleIndex = 0;

static signed char step = 1;

distance [angleIndex ] = readDistance ();

angleIndex += step ;

if (angleIndex == NUM_ANGLES - 1) step = -1;

else if (angleIndex == 0) step = 1;

servo . write ( sensorAngle[angleIndex ] );

```



```
}

// Initial configuration

//

// Configure the input and output pins

// Center the servo

// Turn off the motors

// Test the motors

// Scan the surroundings once

//

void setup () {

pinMode(trigPin , OUTPUT);

pinMode(echoPin, INPUT);

digitalWrite ( trigPin , LOW);

pinMode(enAPin, OUTPUT);

pinMode(in1Pin, OUTPUT);

pinMode(in2Pin, OUTPUT);

pinMode(in3Pin, OUTPUT);

pinMode(in4Pin, OUTPUT);

pinMode(enBPin, OUTPUT);

servo . attach ( servoPin );

//servo . write (0);

//testMotors ();

// Scan the surroundings before starting

//servo . write ( sensorAngle[0] );

delay (200);

for (unsigned char i = 0 ; i < NUM_ANGLES ; i ++ )

readNextDistance (), delay (200);

}
```

```
// Main loop:

//

// Get the next sensor reading

// If anything appears to be too close , back up

// Otherwise, go forward

//

void loop () {

  readNextDistance ();

  // See if something is too close at any angle

  unsigned char tooClose = 0;

  for (unsigned char i = 0 ; i < NUM_ANGLES ; i++)

    if ( distance [ i ] < 250)

      tooClose = 1;

  if (tooClose) {

    // Something's nearby: back up left

    go(LEFT, -255);

    go(RIGHT, 0);

  } else {

    // Nothing in our way: go forward

    go(LEFT, 255);

    go(RIGHT, 255);

  }

  // Check the next direction in 50 ms

  delay (50);

}
```

ДОДАТОК Б

```

using System; using
System.Linq;using
System.Text;using
System.Data;

using System.Drawing;

using System.Windows.Forms;
using System.ComponentModel;

using System.Collections.Generic;using
System.IO;

namespace WinFormCharpWebCam

{

    //Design by Pongsakorn Poosankam 14.07.17public
    partial class mainWinForm : Form

    {

        public int photo = 0, zf = 0;
        public long nd,nn,kd;

        public int nx, mx, ny, my, ix, iy,fx;Timer
        timer;

        public String s1, s3,ss,sk="No"; public int
        c0, c1, u, v, j, j1, j2, v1;public byte[] v2 =
        new byte[25];

        public char[] a = new char[25],a1=new char[25];public
        char ca;

        public byte[] w1 = new byte[307254];
        System.Drawing.Bitmap sky, rbt; Graphics
        g;

        Rectangle rct; public
        mainWinForm()

        {

            InitializeComponent();

        }

        WebCam webcam;

        private void mainWinForm_Load(object sender, EventArgs e)

        {

            sky = Properties.Resources.sky;rbt =
            Properties.Resources.rbt; webcam =
            new WebCam();

            webcam.InitializeWebCam(ref imgVideo);
            serialPort1.Open();

            this.BackgroundImage = Properties.Resources.sky;
            rbt.MakeTransparent();

            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;

            this.MaximizeBox = false;

```

```
        g = Graphics.FromImage(BackgroundImage);
        rct.Width = rbt.Width;

        rct.Height = rbt.Height;
        rct.X = 700; rct.Y = 200;

        g.DrawImage(rbt, rct.X, rct.Y);
    }

    private void bntStart_Click(object sender, EventArgs e)
    {
        webcam.Start();
    }

    private void bntStop_Click(object sender, EventArgs e)
    {
        webcam.Stop();
    }

    private void bntContinue_Click(object sender, EventArgs e)
    {
        webcam.Continue();
    }

    private void bntCapture_Click(object sender, EventArgs e)
    {
        imgCapture.Image = imgVideo.Image;
    }

    private void bntSave_Click(object sender, EventArgs e)
    {
        Helper.SaveImageCapture(imgCapture.Image);
    }

    private void bntVideoFormat_Click(object sender, EventArgs e)
    {
        webcam.ResolutionSetting();
    }

    private void bntVideoSource_Click(object sender, EventArgs e)
    {
        webcam.AdvanceSetting();
    }

    private void timer1_Tick(object sender, EventArgs e)
```

```

    {
        if (zf == 1)
        {
            imgCapture.Image = imgVideo.Image;
            Helper.Save(imgCapture.Image, "Imagr.bmp");photo++;

FileStream fr = new FileStream("Imagr.bmp", FileMode.Open, FileAccess.Read);

//FileInfo mf = new FileInfo("\\\\Win- cgorfim3mg8\\users\\Public\\obmen00\\coord.txt");

//FileStream fs = mf.Open(FileMode.OpenOrCreate,
FileAccess.Write,FileShare.ReadWrite);

BinaryReader br = new BinaryReader(fr); nd = fr.Length;if (nd ==
307254)
    {
for (long n1 = 0; n1 < nd; n1++) w1[n1] = br.ReadByte();

        nx = -1; mx = -1; ny = -1; my = -1; ;for (ix
= 0; ix < 240; ix++)
            {
                for (iy = 0; iy < 320; iy++)
                    {
                        nn = (ix * 320 + iy) * 4+54;

if ((w1[nn + 1] > 0xCF) && (w1[nn + 2] > 0xCF) && (w1[nn + 3] > 0xCF))
                    {
if (nx == -1) nx = ix; if (ny == -1) ny = iy; fx =ix;
                    }
                    else
                    {
                        if (nx != -1)
                            {
                                if (my == -1) my = iy;mx
                                = fx;
                            }
                    }
            }
    }
}

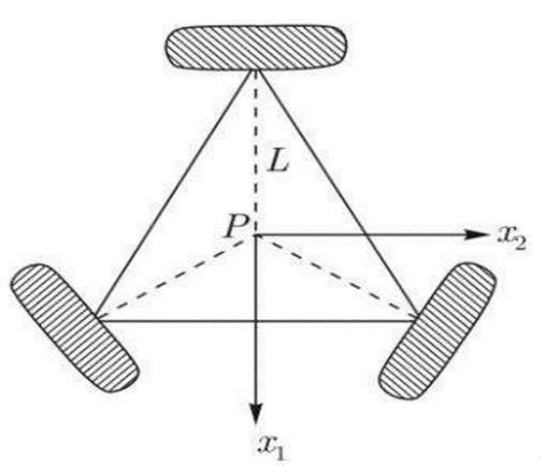
```

$$J_1 = \begin{bmatrix} J_{2f} \\ J_{2oc} \end{bmatrix} = \begin{bmatrix} 0 & 1 & L \\ 0 & -1 & L \\ \cos(\beta_{oc3}) & \sin(\beta_{oc3}) & L\cos(\beta_{oc3}) \end{bmatrix}$$

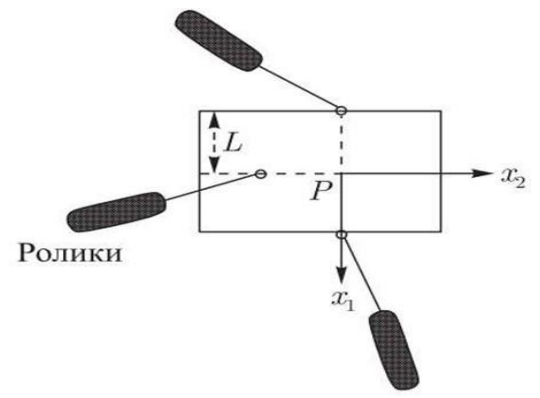
$$J_2 = \text{diag}(r)$$

$$C_2 = \begin{bmatrix} C_{2f} \\ C_{2oc} \end{bmatrix} = \begin{bmatrix} 0 & 1 & L \\ 0 & -1 & L \\ \sin(\beta_{oc3}) & -\cos(\beta_{oc3}) & L\cos(\beta_{oc3}) \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}$$



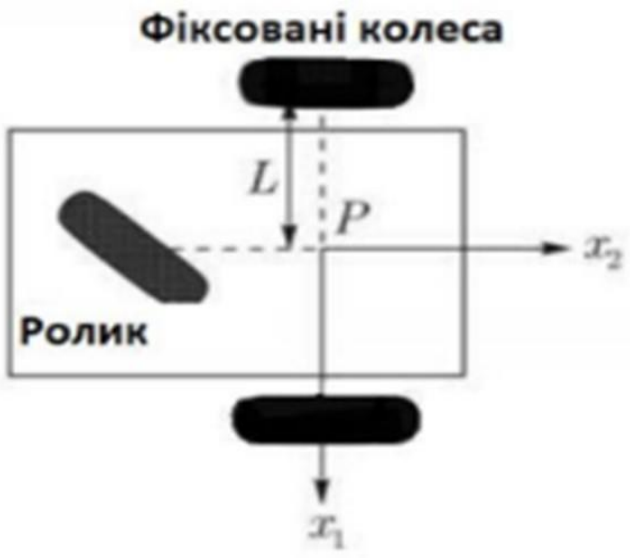
Робот з омніколесами, розташованими у вершинах корпусу



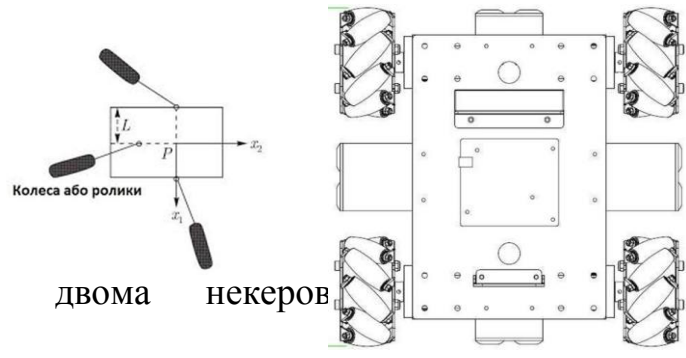
Омнімобільний робот із нецентральними орієнтованими колесами



Двоколісний робот із некерованим третім колесом



Модель типу триколісного дитячого велосипеда



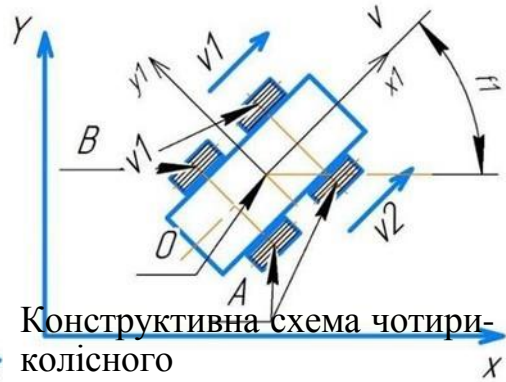
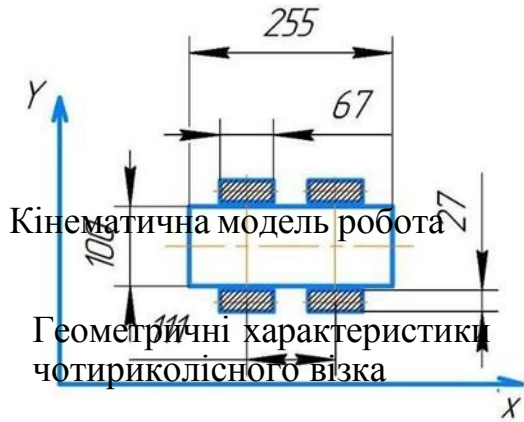
Робот із двома некерованими колесами

Модель робот із чотирма провідними колесами

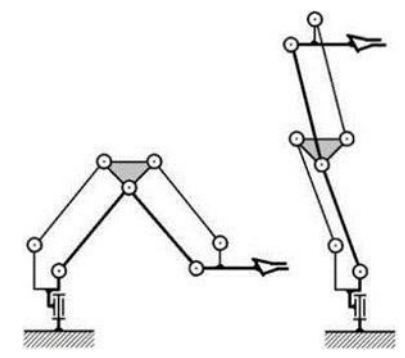
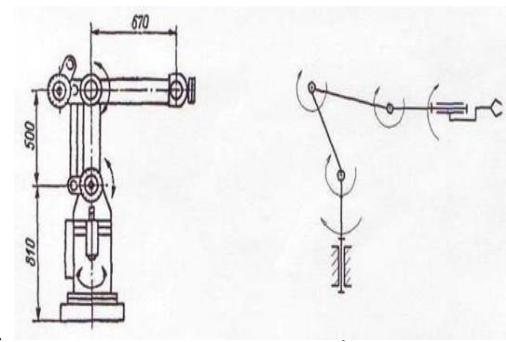
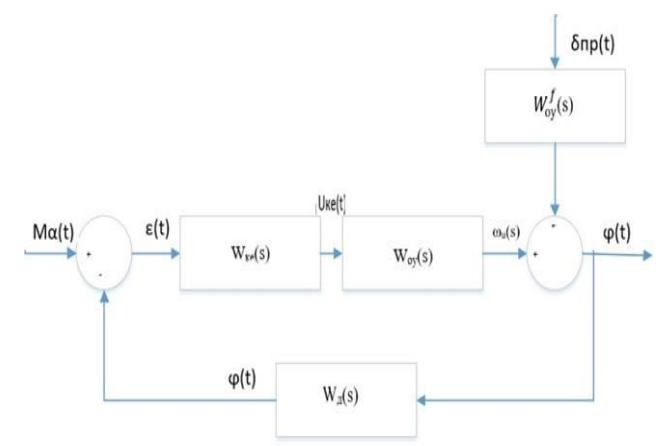
Зв'язки, накладені на систему:

$$\left. \begin{aligned} \dot{X} &= v1 \cdot \cos(f1) \\ \dot{Y} &= v2 \cdot \sin(f1) \\ f1 &= \omega \end{aligned} \right\}$$

					362.1.00			
Изм.	Лист		Подп.	Дата	Стан проблеми розробки та аналіз існуючих мобільних платформ мані-пуляційних роботів	Лит.	Лист	Листов
Разраб.		Гаркуша В.М				у		
Пров.		доц..к.т.н Гавриленко О.В						
Н.								



$$\begin{cases} \dot{X} = v \cdot \cos(f1) \\ \dot{Y} = v \cdot \sin(f1) \\ f1 = \omega \end{cases}$$



Передаюча функція коригувального пристрою:

$$W_{ke}(s) = \frac{U_{ke}(s)}{\varepsilon(s)} = K_p$$

$$W_{oy}(s) = \frac{\varphi(s)}{\psi(s)} = \frac{0.12844 s^3 + 11.34564 s^2 + 3s + 64.955}{0.12844 s^3 + 11.34564 s^2}$$

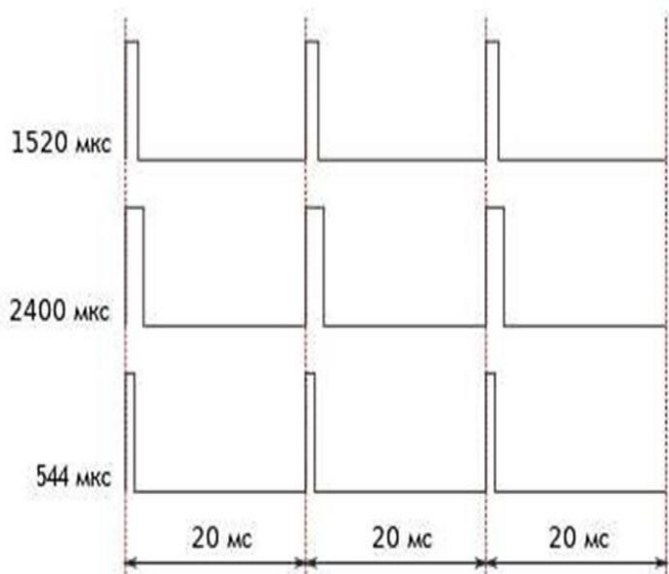
$$W_{oy}^f(s) = \frac{\varphi(s)}{\delta_{np}(s)} = \frac{-3.77}{0.12844 s^3 + 11.34564 s^2}$$

$K_M = 1;$

Передаюча функція розімкнутої системи за керуючим впливом:

$$W_u^u(s) = \frac{\varphi(s)}{M_\alpha(s)} = W_{ke}(s) W_{oy}^u(s) \cdot \frac{1}{s} = \frac{K_p(0.12844 s^3 + 11.34564 s^2 + 3s + 64.955)}{0.12844 s^3 + 11.34564 s^2}$$

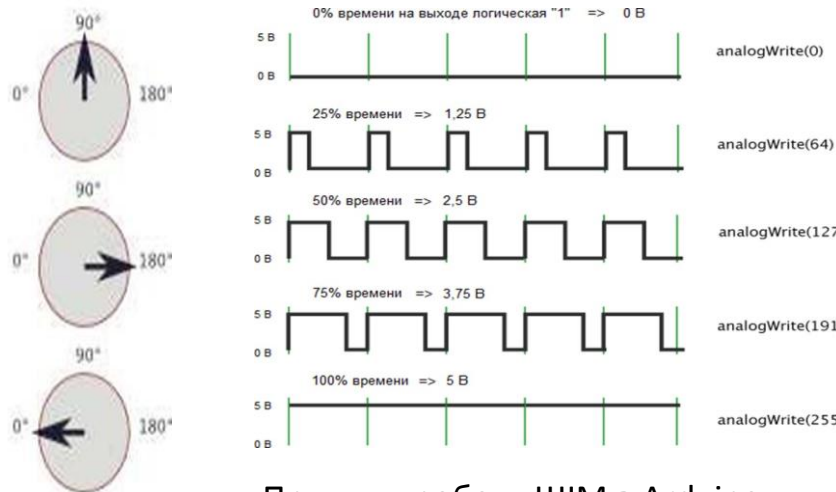
					362.2.00				
Изм.	Лист	Подп.	Дата	Кінематична модель мобільного робота			Лит.	Лист	Листов
Разраб.	Гаркуша В.М						у		
Пров.	Доц..к.т.н Гавриленко О.В								
конт.									
Н.									



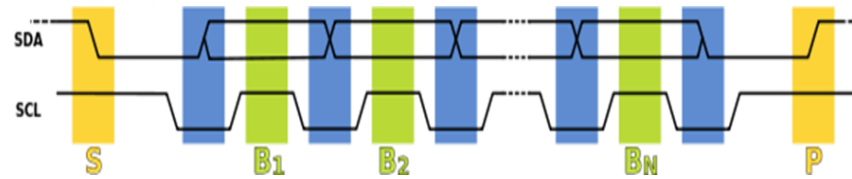
Управління сервоприводом. Форми керуючих сигналів

До сервоприводу тягнеться три дроти. Два з них відповідають за живлення мотора, третій доставляє керуючий сигнал, який використовується для виставлення положення пристрою.

Широтно-імпульсно модульований сигнал - це імпульсний сигнал постійної частоти, але змінної шпаруватості (співвідношення тривалості імпульсу та періоду його прямування).



Принцип роботи ШІМ в Arduino



Тактування послідовності передачі даних

I²C (ІІС, англ. Inter-Integrated Circuit) – послідовна асиметрична шина для зв'язку між інтегральними схемами всередині електронних приладів.

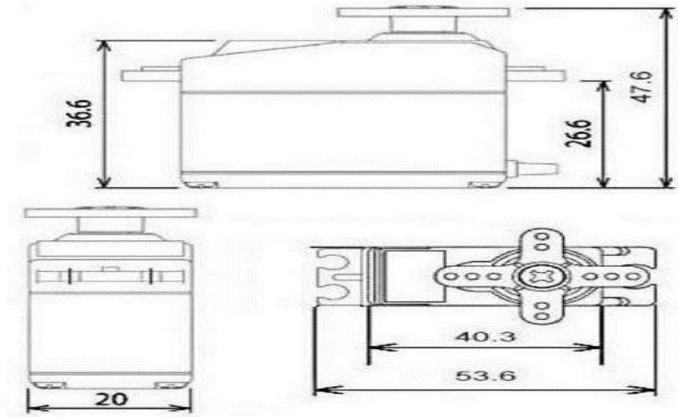
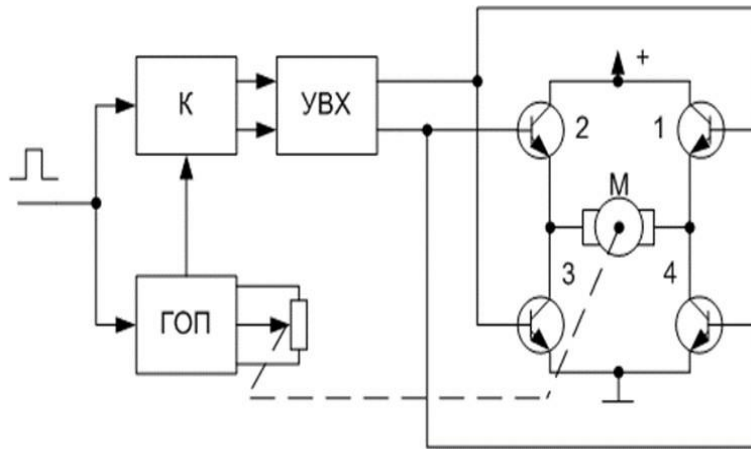
Управління двигуном постійного струму найпростіше організувати за допомогою ШІМ-регулятора. ШІМ – це широтно-імпульсна модуляція, в англійській мові це називається PWM — Pulse

Width Modulation. Використовуючи L293D ми можемо управляти двигунами з досить широким діапазоном напруги живлення від 4.5 до 36 Вольт, але при цьому, L293D може видати всього лише 600mA тривалого струму навантаження на кожен канал.

					362.3.00				
Изм.	Лист	Подп.	Дата	Принципи управління сервоприводами та постійним струм-мом			Лит.	Лист	Листов
Разраб.	Гаркуша В.М						у		
Пров.	доц..к.т.н Гавриленко О.В								
Н.									



Сервопривід MG996R



Характеристики сервоприводу MG996R

Характеристики	Значення
Робоча напруга	4.8-7.2В
Матеріал редуктора	Металевий
Маса	55 г
Номінальний споживаний струм	500 – 900 мА
Пусковий струм	До 2.5А
Швидкість обертання (при 6В)	60°/0.14s
Крутний момент (при 4.8В)	9.4 кг·см
Крутний момент (при 6В)	11 кг·см

Пристрій сервоприводу

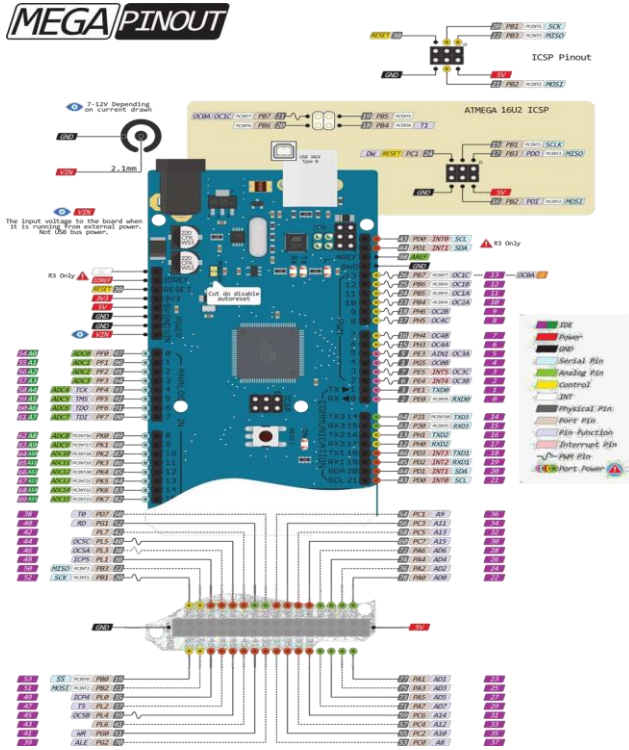
Характеристики сервоприводу MG996R

Характеристики	Значення
Робоча напруга	4.8-6В
Матеріал редуктора	Металевий
Маса	13.4 г
Номінальний споживаний струм	12 – 100 мА
Пусковий струм	До 1А
Швидкість обертання (при 6В)	60°/0.08s
Крутний момент (при 4.8В)	1.8 кг·см
Крутний момент (при 6В)	2.2 кг·см



Сервопривід MG90S

					362.4.00			
Изм.	Лист		Подп.	Дата	Характеристика двигуна та його комплектуючих запчас- тин	Лит.	Лист	Листов
						у		
Разраб.		Гаркуша В.М						
Пров.		Доц..К.Т.Н Гавриленко О.В						



Розпіновка плати

Arduino Mega 2560 – флагманська платформа для розробки на базі мікроконтролера ATmega2560.

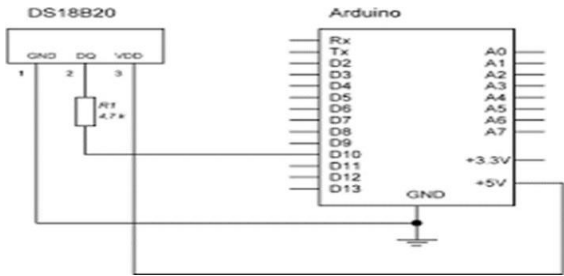
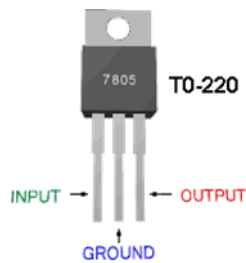


Схема підключення до Arduino

Характеристика Arduino Mega 2560

Характеристика	Значення
Мікроконтролер	ATmega2560
Тактова частота	16 МГц
Напруга логічних рівнів	5 В
Вхідна напруга живлення	7–12 В
Максимальний струм із піна введення виводу	40 мА
Максимальний вихідний струм піна 3.3V	50 мА
Максимальний вихідний струм піна 5V	800 мА
Оперативна пам'ять	8 КБ
Розрядність АЦЦ	10 біт
Flash-пам'ять	256 КБ
EEPROM-пам'ять	4 КБ



Лінійний стабілізатор напруги L7805



Диск із прорізами із чорного оргскла для оптичного датчика

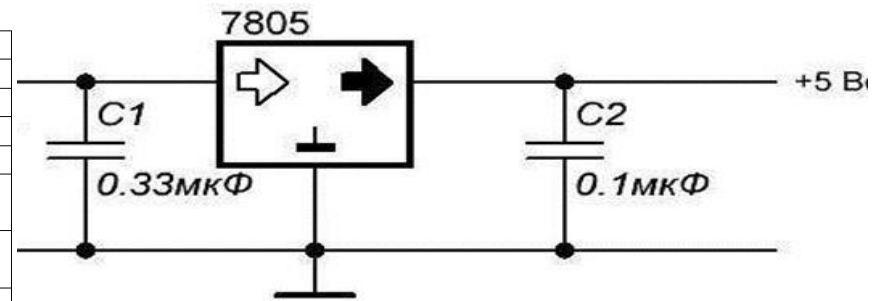


Схема підключення стабілізатора



Схема роботи датчика швидкості обертання

Оптичний датчик швидкості

					362.7.00			
Изм.	Лист		Подп.	Дата	Моделювання динаміки КОМ-плектуючих плат та датчиків	Лит.	Лист	Листов
Разраб.	Гаркуша В.М					у		
Пров.	доц..к.т.н Гавриленко О.В							
Н.								

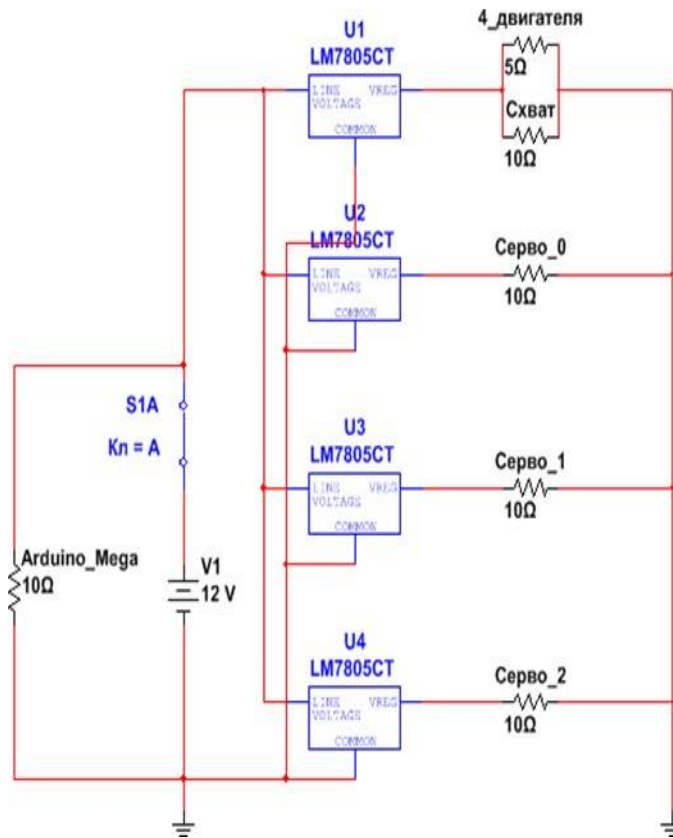


Схема живлення елементів мобільного робота

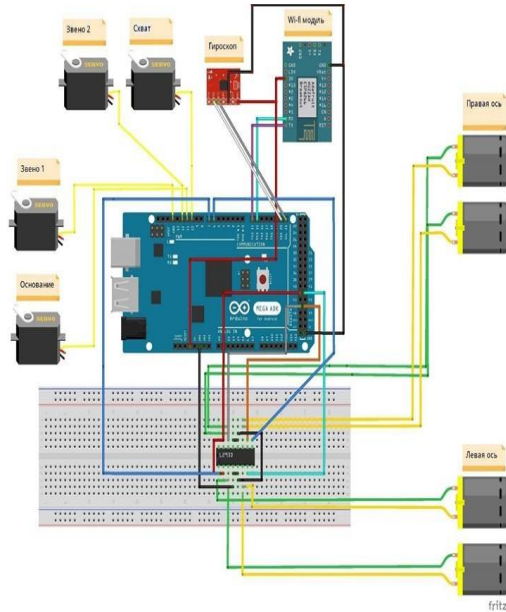
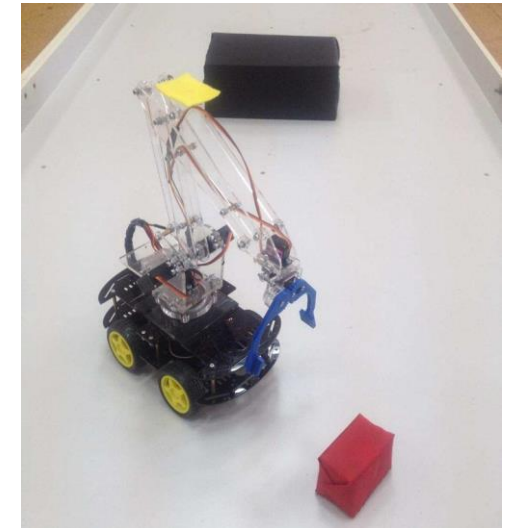


Схема підключення всіх компонентів робота

Сервоприводи підключені до входів 10-13, які дозволяють виводити аналогові 8-бітові значення у вигляді ШІМ-сигналу.

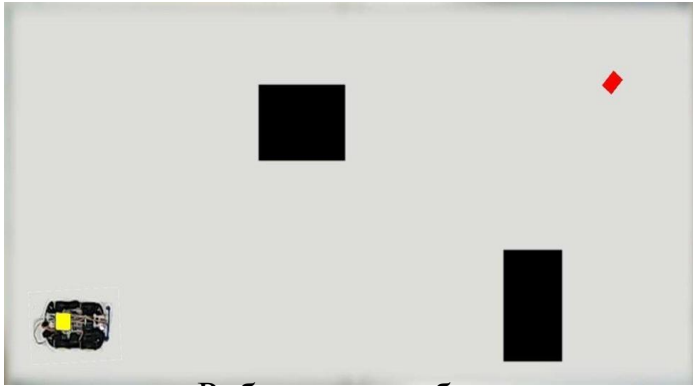
Мотор-редуктори розділені на праву вісь та ліву. Два двигуни підключаються до однієї сторони каналу драйвера L293D, і два з іншого боку. Виходи мікросхеми це: IN1, IN2, IN3, IN4, EN1, EN2.



Мобільний маніпуляційний робот

Для живлення основних елементів мобільного робота використовується блок живлення на 12 Вольт. Це суперечить визначенню мобільного робота, він не повинен бути прив'язаний його до проводів, він має бути повністю автономним. Але у зв'язку з бюджетом проєкту, на етапі розробки було вирішено використовувати живлення

					362.6.00			
Изм.	Лист		Подп.	Дата	Схема живлення елементів мобільного робота	Лит.	Лист	Листов
Разраб.		Гаркуша В.М				у		
Пров.		доц..к.т.н Гавриленко О.В						
Н.								



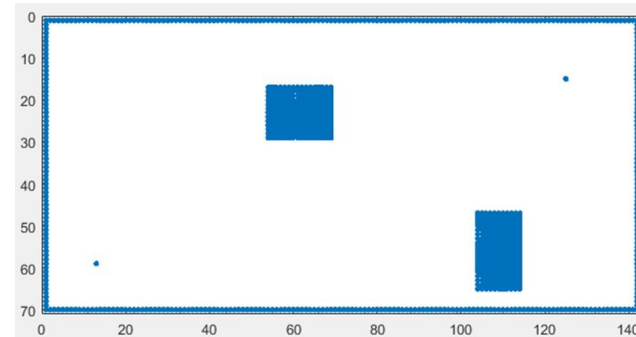
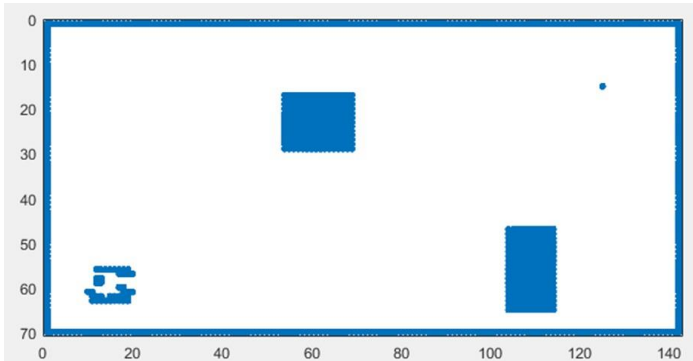
Робоча зона робота

На відстані 3.5м від підлоги встановлена веб-камера Sven IC-975 HD. Камера підключена до ПК через USB 2.0 і передає дані до пакета матлаб для використання веб-

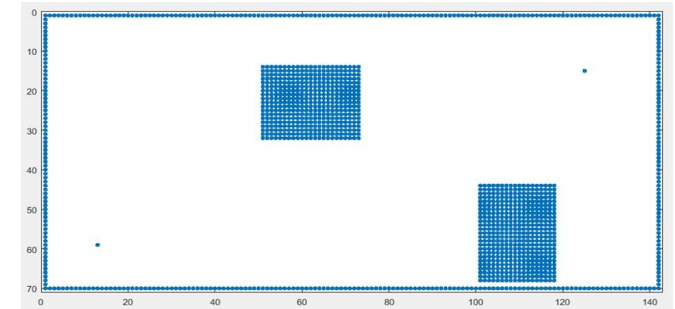
[X,Y]: [1246 144]
Index: 1
[R,G,B]: [1 0 0]

Індекс кінцевої точки

Після застосування цієї функції складаємо матрицю з елементів: 0,1,2 - 1.0 - це вільний прохідний осередок; 2 - це початкове положення робота; 1 – положення об'єкта для маніпулювання.

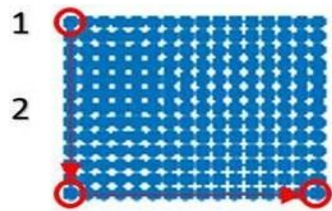


Готова матриця з початковою і кінцевою координатою в середовищі з перешкодами



Перешкоди, збільшені на 3 комірки з кожної сторони

Початкове положення робота визначається ще й іншими значеннями, оскільки мобільний робот має забарвлення чорного кольору.



3

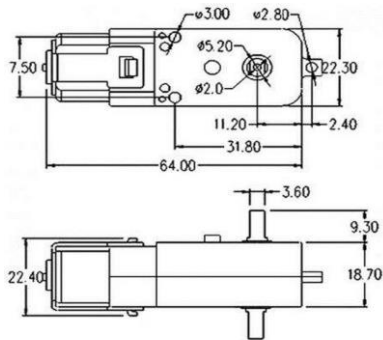
Алгоритм збільшення перешкоди

Ма-

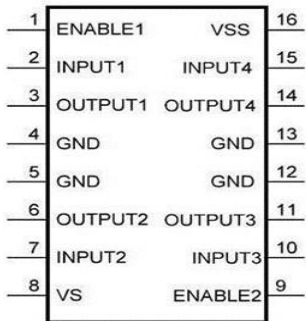
					362.7.00			
Изм.	Лист		Подп.	Дата	Моделювання динаміки сис-теми управління при заданних початкових	Лит.	Лист	Листов
Разраб.	Гаркуша В.М					у		
Пров.	доц..к.т.н Гавриленко О.В							
Н.								



Мотор-редуктор DC-6v



Розміри мотор-редуктора DC-6v

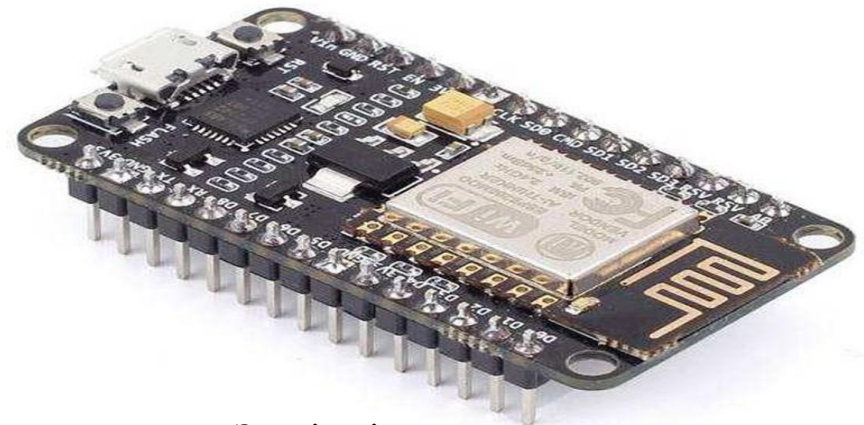


L293D

Мікросхема L293D

Характеристика двигуна постійного струму для платформи

Характеристика	Значення
Робоча напруга	3-12В
Рекомендована напруга	6-8В
Передатне число редуктора (без навантаження 3В)	1:48
Номінальний споживаний струм	70мА
Максимальний струм (при 6В)	250мА
Швидкість обертання (при 6В)	90 об/хв
Обертаючий момент	2 кг·см
Щум	65 dB



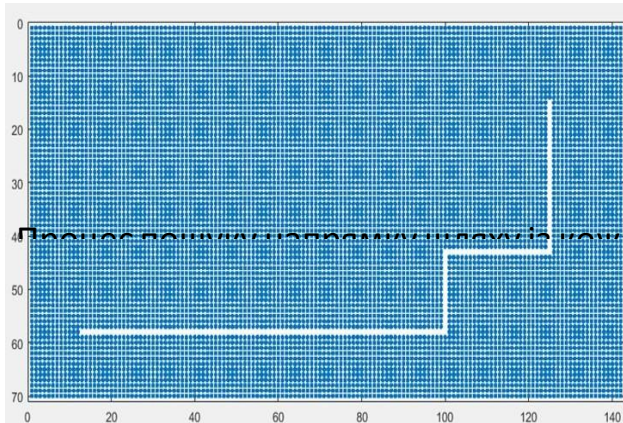
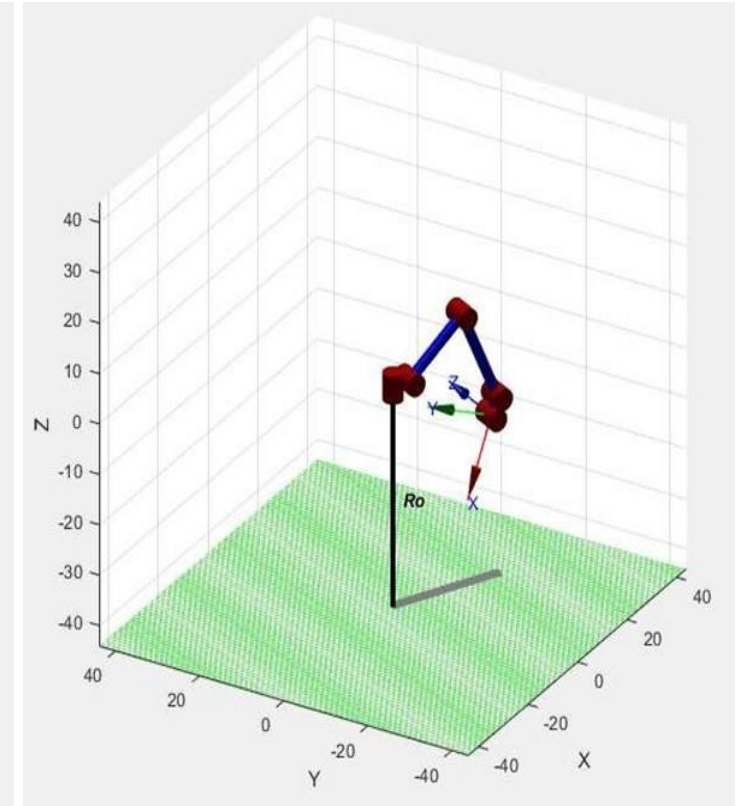
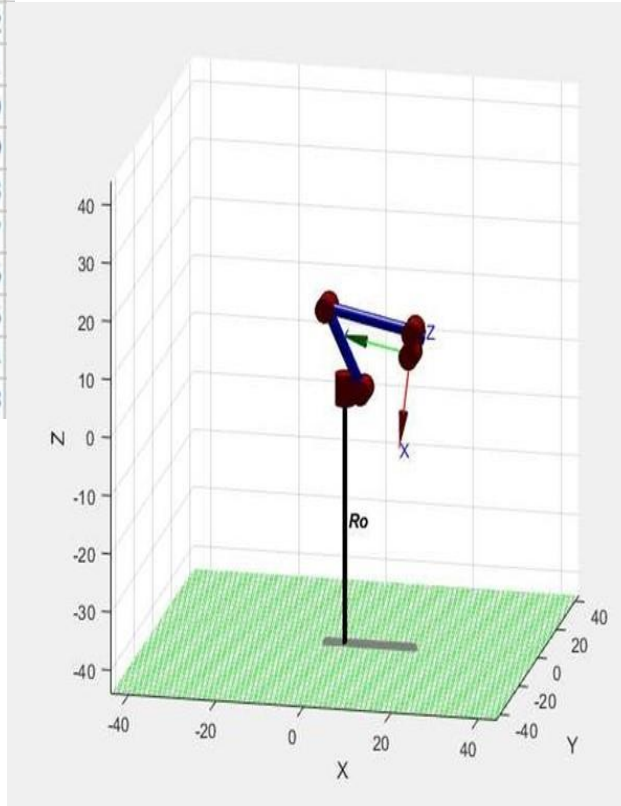
Зовнішній вигляд модуля



Порти модуля

					362.8.00			
Изм.	Лист		Подп.	Дата	Конструкторська частина	Лит.	Лист	Листов
Разраб.		Гаркуша В.М				у		
Пров.		доц..к.т.н Гавриленко О.В						
Н.								

158	159	160	161	162
157	158	159	160	161
156	157	158	159	160
155	156	157	158	159
154	155	156	157	158
153	154	155	156	157
152	153	154	155	156
151	152	153	154	155
150	151	152	153	154
149	150	151	152	153

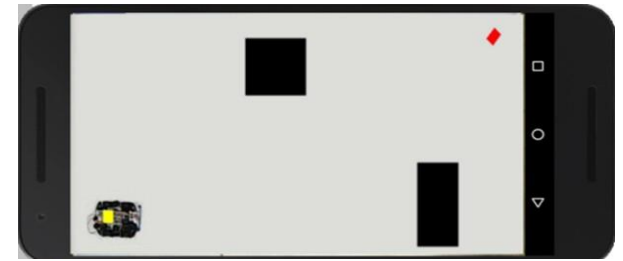
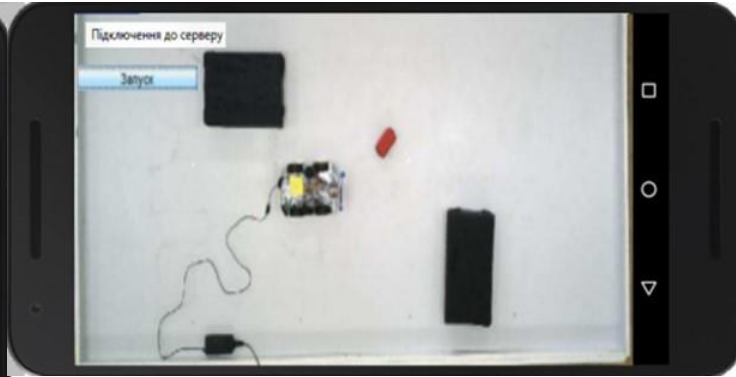


Шлях в середовищі із перешкодами

Початкове положення ланок

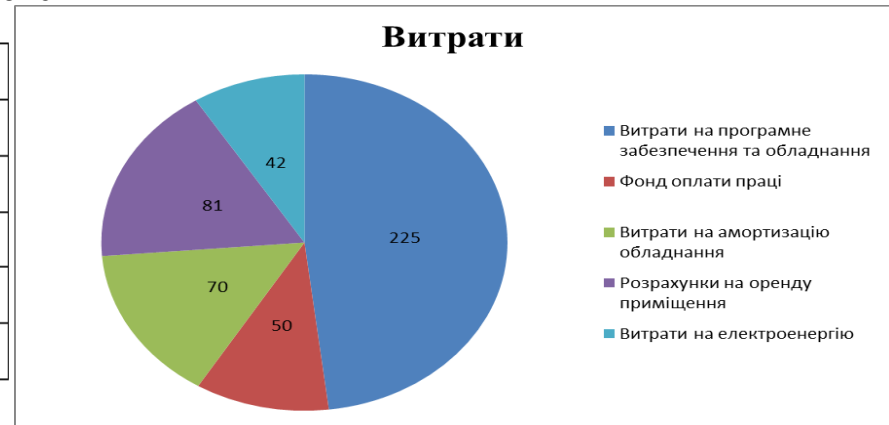
Кінцеве положення ланок маніпулятора

					362.9.00					
Изм.	Лист		Подп.	Дата	Дослідницька частина			Лит.	Лист	Листов
Разраб.	Гаркуша В.М							у		
Пров.	доц..к.т.н Гавриленко О.В									
Н.										



Вікно після натискання кнлпки «підключення до сервера»

Назва	Вартість, грн
Microsoft Windows 7 Home Premium	1746
Microsoft Office 2016	5240
MATLAB and Simulink Student Suite	900
SolidWorks 2018 Standard	90000
Всього:	97886



Крок 1	Інсталяція та конфігурація пристроїв захоплення зображень
Крок 2	Отримання інформації, яка однозначно ідентифікує пристрій захоплення зображення в OpenCV
Крок 3	Створення вихідних об'єктів відеопослідовності
Крок 4	Попередній перегляд відеопослідовності (необов'язково)
Крок 5	Формування властивостей об'єктів захопленого зображення (необов'язково)
Крок 6	Захват даних зображення
Крок 7	Очищення даних

Изм.	Лист	Подп.	Дата	362.10.00				
Разраб.	Гаркуша В.М			Експериментальна частина.		Лит.	Лист	Листов
Пров.	доц..к.т.н Гавриленко О.В					у		
Н. конт.				Економічна частина				