

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» _____ 2024 р

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «Програмна модель рухомої платформи з механум-колесами»

Захищено на засіданні
Атестаційної комісії № 44
протокол № __ від __.06.2024 р.
Оцінка ____ / ____
Голова Атестаційної комісії
_____ **СКОБ Ю. О.**
(підпис) (прізвище та ініціали)

Виконав:
студент 4 курсу, групи КУ– 41
Галузь знань: 15 – Автоматизація та
приладобудування
Спеціальність: 151 – «Автоматизація та
комп'ютерно-інтегровані технології»
ГОРЕНКО Данієль Васильович

(підпис)

Керівник: к.ф.-м.н., доцент ЗВО
кафедри ТПС
КОТВИЦЬКИЙ Альберт Тадеушевич

(підпис)

Рецензент: к.т.н, доцент ЗВО кафедри
електроніки і управляючих систем
РЕВА Сергій Миколайович

(підпис)

Харків – 2024

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 72 сторінки, із яких 47 сторінок основної частини з 35 рисунками, 3 таблиці, 19 найменуваннями списку використаних джерел та чотирма додатками.

Метою кваліфікаційної роботи є візуалізація алгоритму керування платформою на механум колесах та вивчення способів керування DC-моторів.

Об'єкт дослідження – процес керування рухомими платформами.

Предмет дослідження – математична та програмна моделі керування рухомої платформи на механум-колесах.

Проблема, яка вирішується в кваліфікаційній роботі полягає в тому, щоб створити візуальне представлення алгоритму керування та способу обертання механум колесами.

Область застосування – робототехніка та мехатроніка. Розроблена програмна модель допоможе розробці нових алгоритмів керування платформами на механум-колесами.

Ключові слова: механум-колесо, рухома платформа, візуалізація алгоритму, Wolfram Mathematica, DC-мотори, H-міст.

ABSTRACT

An explanatory note to a bachelor's thesis consists of an introduction, three sections, conclusions, a list of sources used, and four appendices.

The total volume of the work is 72 pages, of which 47 pages are the main part with 35 figures, 3 tables, 19 names of the list of used sources and four appendices. The purpose of the qualification work is to visualize the platform control algorithm on mecanum wheels and to study the methods of controlling DC motors.

The problem that is solved in the qualification work is to create a visual representation of the control algorithm and the method of rotation of the mecanum with wheels.

Scope – robotics and mechatronics. The developed software model will help in the development of new control algorithms for mecanum-wheeled platforms.

Keywords: mecanum-wheel, moving platform, algorithm visualization, Wolfram Mathematica, DC-motors, H-bridge.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1 АКТУАЛЬНІСТЬ РУХОМИХ ПЛАТФОРМ	9
1.1 Актуальність рухомих платформ	9
1.2 Види рухомих платформ	11
1.2.1 Колісні платформи.....	11
1.2.2 Гусеничні платформи	12
1.2.3 Платформи на механум-колесах	13
Висновки до розділу 1	15
РОЗДІЛ 2 МОДЕЛЮВАННЯ РУХУ	16
2.1 Опис основних напрямів руху	16
2.2 Математична модель рухомої платформи.....	20
2.3 Модель рухомої платформи.....	22
2.3.1 Середовище розробки інтерактивної програмної моделі	22
2.3.2 Програмна модель рухомої платформи.....	23
2.4 Тестування моделі	25
2.5 Обертання	27
2.5.1 Основи обертального руху платформи на механум колесах	27
2.5.2 Геометрія обертального руху	29
Висновки за розділом 2	31
РОЗДІЛ 3 МОДЕЛЮВАННЯ КЕРУВАННЯ DC-МОТОРАМИ	33
3.1 Опис основних механічних актуаторів.....	33
3.1.1 STEPPER Motor	33

3.1.2 BLDC Motor	36
3.1.3 DC Motor	37
3.2 Керування DC-двигунами.	40
3.2.1 H-міст	40
3.2.2 Драйвер L293D	42
3.2.3 Драйвер TB6612FNG	43
3.3 Тестування драйверів в ISIS Proteus	44
3.3.1 Тестування макету L293D	45
3.3.2 Тестування макету TB6612FNG	48
Висновки до розділу 3	50
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
ДОДАТКИ	55
Додаток А	55
Додаток Б	57
Додаток В.....	61
Додаток Г	67

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

BLDC - Motor	—	Brushless Direct Current Motor;
DC - Motor	—	Direct Current Motor.

ВСТУП

Робототехніка є однією з тих галузей сучасної науки та техніки, що активно розвиваються. Вона охоплює широкий спектр застосувань, від промислового виробництва до медичної сфери та космічних досліджень. Широке впровадження робототехніки дозволяє значно підвищити ефективність виробничих процесів, автоматизувати рутинні завдання та покращити якість життя людей. Розвиток робототехніки сприяє створенню нових технологій, які забезпечують безпечні умови праці, підвищують продуктивність і зменшують витрати.

Одним з важливих напрямків у робототехніці є розробка роботизованих рухомих платформ. Такі платформи здатні самостійно переміщатися в просторі, виконуючи різноманітні завдання в автономному або напіваавтономному режимі. Роботизовані рухомі платформи використовуються у різних галузях: від логістики, де вони оптимізують процеси зберігання й транспортування товарів, до медичної сфери, де вони можуть допомагати в переміщенні пацієнтів та медичного обладнання. Військові та рятувальні операції також значно вииграють від використання таких платформ, оскільки вони можуть виконувати завдання в небезпечних або недоступних для людини умовах.

Завдяки можливості самостійного пересування, роботизовані платформи мають потенціал значно підвищити ефективність та безпеку в різних сферах діяльності. Вони оснащені складними системами навігації та керування, які дозволяють їм орієнтуватися в просторі, уникати перешкод та виконувати задані завдання з високою точністю. Сучасні технології, такі як сенсори, системи комп'ютерного зору та алгоритми штучного інтелекту, роблять ці платформи ще більш автономними й універсальними в застосуванні.

Враховуючи все вищезазначене, дослідження та розробка роботизованих рухомих платформ є актуальними й перспективними напрямками наукових досліджень і технічних розробок. Вони відкривають нові можливості для

вдосконалення існуючих технологій та створення інноваційних рішень, які зможуть вирішувати складні задачі в різних галузях промисловості та сервісу.

РОЗДІЛ 1

АКТУАЛЬНІСТЬ РУХОМИХ ПЛАТФОРМ

Основною задачею рухомих платформ є забезпечення високої маневреності та точності руху. Рухомі платформи використовуються в різних галузях, включаючи промисловість, логістику та інші сфери, де необхідна автоматизація та ефективність переміщення. Проте, забезпечення маневреності залишається однією з найбільш важливих та складних технічних задач.

Основними проблемами, з які виникають при розробці рухомих платформ, є обмежена мобільність, складність керування та висока вартість реалізації. Традиційні рухомі платформи часто не здатні рухатися у всіх напрямках, що знижує їх ефективність у складних умовах. Забезпечення точного й надійного керування платформою вимагає складних алгоритмів і сучасного обладнання, що ускладнює процес. Крім того, високі витрати на розробку та впровадження систем керування й сенсорних систем, необхідних для досягнення високої маневреності, є значною перешкодою для широкого використання таких платформ.

У зв'язку з цими проблемами, виникає потреба у дослідженні та розробці нових рішень для покращення маневреності рухомих платформ.

1.1 Актуальність рухомих платформ

Рухомі платформи є надзвичайно важливими в сучасному світі через їхню здатність автоматизувати й оптимізувати процеси в різних галузях. Вони забезпечують підвищену ефективність та продуктивність, дозволяючи виконувати завдання з високою точністю та надійністю.

У промисловості рухомі платформи використовуються для автоматизації виробничих ліній, транспортування матеріалів та виконання складних маніпуляцій, що підвищує загальну ефективність виробничих процесів. У логістиці вони сприяють оптимізації процесів зберігання та доставки,

забезпечуючи швидке та точне переміщення вантажів на складах та транспортних вузлах. Прикладом може слугувати роботизація складів компанії Ocado (Рисунок 1.1 [1]), де використовуються роботи на основі рухомих платформ для розподілення замовлень клієнтів. Ці роботи здатні самостійно переміщуватися по складу, збираючи товари та формуючи замовлення, що значно прискорює процес обробки замовлень та підвищує ефективність складу.



Рисунок 1.1 – Роботизація складів Ocado

Крім того, рухомі платформи є незамінними в обставинах, що становлять небезпеку для людини. Вони можуть використовуватися в екстремальних і шкідливих умовах, таких як ядерні об'єкти, хімічні заводи, зони стихійних лих та інші середовища з високим рівнем ризику. У таких ситуаціях платформи можуть виконувати завдання з дослідження, моніторингу та навіть проведення рятувальних операцій, гарантуючи безпеку для людського життя.

Таким чином, актуальність рухомих платформ постійно зростає завдяки їхній здатності працювати в широкому спектрі умов, підвищувати ефективність і гарантувати безпеку в різних галузях.

1.2 Види рухомих платформ

Існує багато видів рухомих платформ. Кожна має свою конструкцію, особливості, переваги та недоліки.

1.2.1 Колісні платформи

Колісні платформи є одним із найпоширеніших і найвідоміших типів рухомих механізмів у транспорті. Вони засновані на принципі руху за допомогою коліс і використовуються у багатьох транспортних засобах, таких як автомобілі, мотоцикли, вантажні фури та поїзди. (рисунок 1.2 [2]).



Рисунок 1.2 – Колісна база автомобіля

Однією з ключових особливостей колісних платформ є їх конструкція, що складається з колісних пар, які забезпечують контакт з дорожнім покриттям. Зазвичай, одна пара коліс поворотна, а інша – ні. Ці платформи можуть мати різні типи коліс, включаючи одинарні, подвійні, трійні або багатоколісні, залежно від потреб і функціональності транспортного засобу.

Основною перевагою колісних платформ є їх висока швидкість та маневреність на твердому покритті, що є гарним варіантом для швидкісних транспортних засобів. Вони здатні перевозити великі вантажі та пасажирів, а

також мають різноманітні конфігурації та розміри, що дає змогу використовувати їх у різних сферах транспорту та промисловості.

Однак в колісних платформах є і свої обмеження. Наприклад, вони мають обмежену прохідність на м'якому або нерівному ґрунті порівняно з іншими типами рухомих механізмів. І хоч вони мають гарну маневреність, але в умовах замкнутого приміщення можуть виникнути проблеми. Можна зробити так, щоб усі колеса були поворотними, але це може ускладнити конструкцію та керування платформою [3, 4].

1.2.2 Гусеничні платформи

Гусеничні платформи – це тип рухомих механізмів, які використовують гусеничні стрічки замість коліс для руху. Їхній принцип дії полягає в русі по поверхні за допомогою гусеничних ланцюгів, які обертаються навколо колісних барабанів (Рисунок 1.3 [5]). Цей метод дозволяє платформам пересуватися по різних типах поверхні, у тому числі м'які, нерівні покриття або умови зі снігом та піском. Гусеничні платформи використовуються у техніці для будівництва, сільському господарстві, військовій сфері, лісозаготівлі та навіть для моніторингу навколишнього середовища.



Рисунок 1.3 – Гусеничний трактор John Deere 8370RT

Однією з основних переваг гусеничних платформ є їхній високий рівень прохідності. Гусениці дозволяють їм подолати перешкоди, такі як долини, болота або кам'янистий ґрунт, які можуть бути недосяжні для колісних платформ. Це робить їх особливо корисними для роботи в умовах поганого дорожнього покриття.

Як і колісні платформи, гусеничні мають свої обмеження. Одне з них – це нижча швидкість руху на твердому покритті. Гусениці створюють більше тертя, що призводить до менш ефективної передачі енергії, обмеження швидкості а також пошкодження дорожнього покриття. Крім того, гусеничні платформи можуть бути менш маневрені на вулицях та інших вузьких місцях, оскільки їхні габарити зазвичай більші, а обертання менш ефективно порівняно з колісними системами.

1.2.3 Платформи на механум-колесах

Платформи на механум-колесах є інноваційним типом рухомих механізмів, які використовують спеціальні колеса.

Механум-колесо [6] винайшов шведський інженер компанії «Mecanum AB» Бенг Ілон у 1973 році, і вже у 1975 році він отримав патент у США на свій винахід. Ці колеса, які можна також зустріти під назвами «колесо Ілона» або «механум-колесо», були створені для того, щоб транспортні засоби могли легко рухатись у різні боки або обертатись на місці з мінімальною силою тертя. Це вдалося зробити завдяки особливій конструкції колеса, яка дозволяла маневрувати з надзвичайною легкістю та ефективністю.

Механум-колесо складається з основного колеса (див. рисунок 1.4), по зовнішньому колу якого рівномірно розташовані незалежні один від одного ролики під кутом 45° до осі основного колеса. Така конструкція надає колесу три ступені вільності: обертання навколо основної осі, обертання роликів та обертання навколо точки контакту ролика з поверхнею.

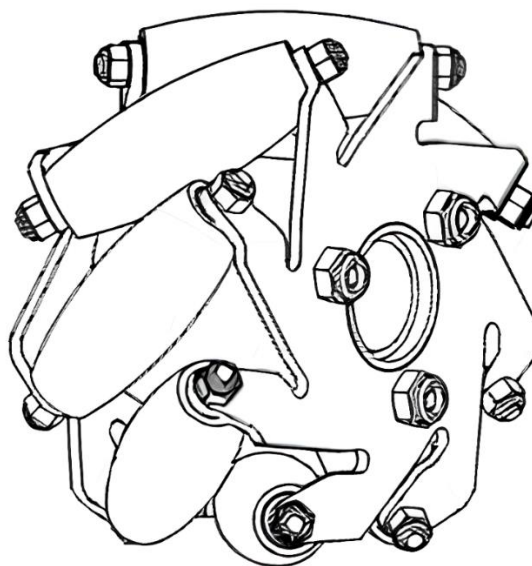


Рисунок 1.4 – Конструкція механум-колеса

Керуючи швидкістю та напрямом обертання кожного з коліс, транспортний засіб, оснащений механум-колесами, може рухатися під будь-яким кутом – не лише вперед-назад, праворуч-ліворуч але й по діагоналі та навколо своєї осі. Таким чином, платформа без повороту може одразу рухатись у заданому довільному напрямку, це є дуже важливою перевагою в умовах обмеженого простору. Платформи на механум-колеса стали широко використовуватися з початком росту та розвитку роботів

Однією з ключових особливостей платформ на механум-колесах є їхній універсальний і гнучкий характер руху, що робить їх дуже корисними в умовах, де потрібна висока маневреність і точність переміщення. Вони можуть використовуватися в різних сферах, включаючи промисловість, магістральну логістику, робототехніку та інші сфери, де потрібна надзвичайна точність та контроль руху.

Однак, платформи на механум-колесах мають і свої обмеження. Наприклад, вони можуть бути менш ефективними на деяких типах поверхонь, таких як нерівний або м'який ґрунт, де колеса можуть просікатися або втрачати тягу.

Висновки до розділу 1

У цьому розділі були розглянуті різні типи рухомих платформ, зокрема колісні платформи та гусеничні механізми. Було проаналізовано їхні переваги, недоліки та особливості використання в різних сферах. Порівнявши можна зробити висновок, що кожен тип платформи має свої унікальні переваги та обмеження, відповідно – певні умови використання.

Платформи на механум-колесах - інноваційний тип рухомого механізму, що відзначається своєю універсальністю та гнучкістю в керуванні. Високий рівень маневреності та точності переміщення робить платформи на механум-колесах ідеальним вибором для завдань, де потрібні всі ці параметри.

Отже, подальша робота буде спрямована на більш детальний аналіз платформ на механум-колесах, включаючи їхню конструкцію, принцип роботи та керування.

РОЗДІЛ 2

МОДЕЛЮВАННЯ РУХУ

2.1 Опис основних напрямки руху

Розглянемо основні можливі напрямки руху платформи [7].

Прямий рух уперед. Для того, щоб платформа почала рухатися прямо уперед, усі колеса повинні обертатися з однаковою швидкістю в одному напрямку. Схематично рух уперед можна побачити на рисунку 2.1

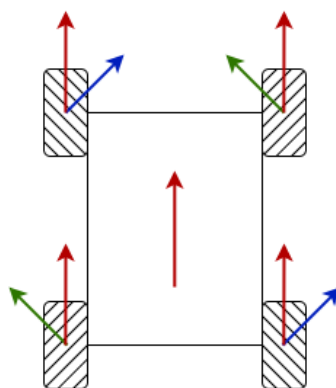


Рисунок 2.1 – Прямий рух вперед

Прямий рух назад. Для прямого руху назад треба, щоб усі колеса обертатися з однаковою швидкістю в протилежному напрямку. Схематично рух назад можна побачити на рисунку 2.2

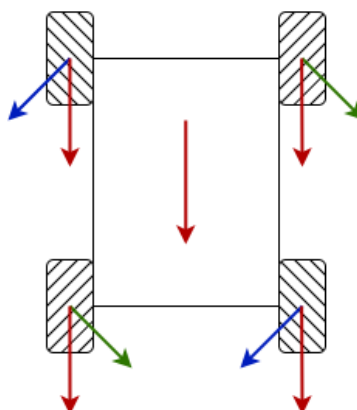


Рисунок 2.2 – Прямий рух назад

Бічний рух вліво. Для цього треба, щоб переднє праве та заднє ліве колеса оберталося в передньому напрямку, а переднє ліве та заднє праве – у протилежному напрямку. Саме так відбувається бічний рух вліво. Схематично зображено на рисунку 2.3

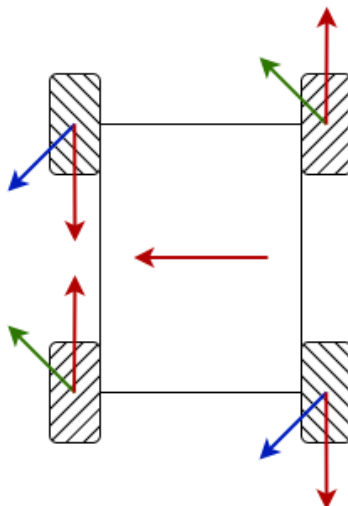


Рисунок 2.3 – Бічний рух вліво

Бічний рух вправо. Щоб платформа почала бічний рух вправо треба щоб переднє ліве та заднє праве колеса оберталося в передньому напрямку, а переднє праве та заднє ліве – у протилежному. Схематично бічний рух вправо зображено на рисунку 2.4

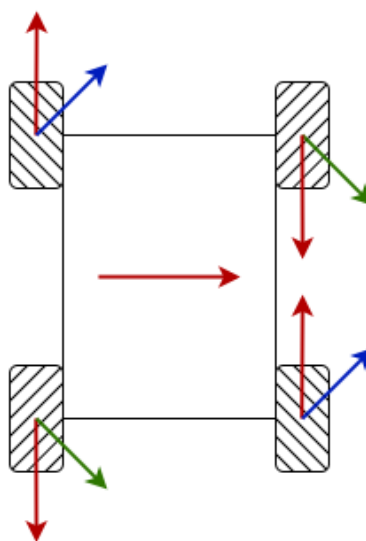


Рисунок 2.4 – Бічний рух вправо

Діагональний рух вгору-вліво. Для початку руху в платформи в цьому напрямку треба, щоб переднє праве та заднє ліве колеса оберталися в передньому напрямку. Схематично діагональний рух вгору-вліво зображено на рисунку 2.5

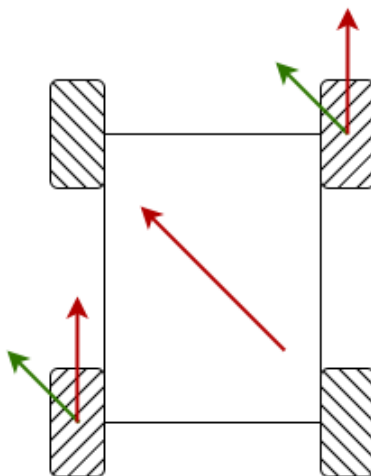


Рисунок 2.5 – Діагональний рух вгору-вліво

Діагональний рух назад-вліво. Для початку руху в платформи в цьому напрямку треба, щоб переднє ліве та заднє праве колеса оберталися в протилежному напрямку. Схематично діагональний рух назад-вліво зображено на рисунку 2.6

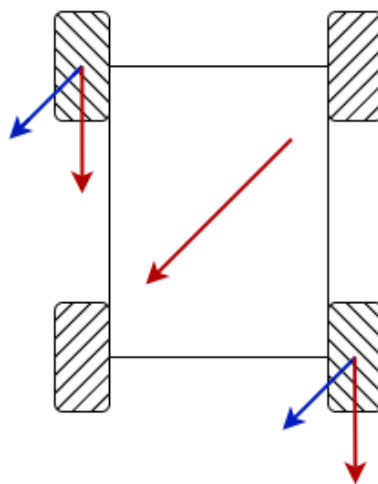


Рисунок 2.6 – Діагональний рух назад-вліво

Діагональний рух вгору-вправо. Для руху платформи в цьому напрямку треба, щоб переднє ліве та заднє праве колеса оберталися в передньому напрямку. Схематично діагональний рух вгору-вправо зображено на рисунку 2.7

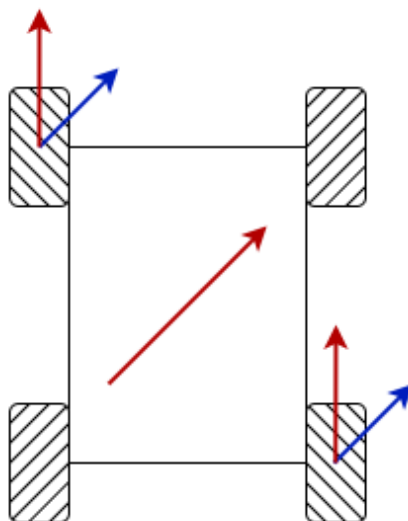


Рисунок 2.7 – Діагональний рух вгору-вправо

Діагональний рух назад-вправо. Для початку руху в платформи в цьому напрямку треба, щоб переднє праве та заднє ліве колеса оберталися в протилежному напрямку. Схематично діагональний рух назад-вправо зображено на рисунку 2.8

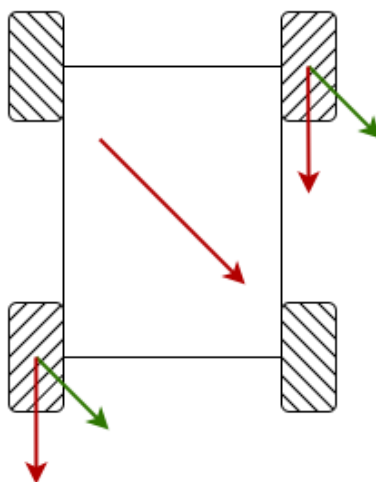


Рисунок 2.8 – Діагональний рух назад-вправо

2.2 Математична модель рухомої платформи

Розглянемо рухому платформу прямокутної форми яка складається з чотирьох механум коліс, схема розташування яких показана на рисунку 2.9

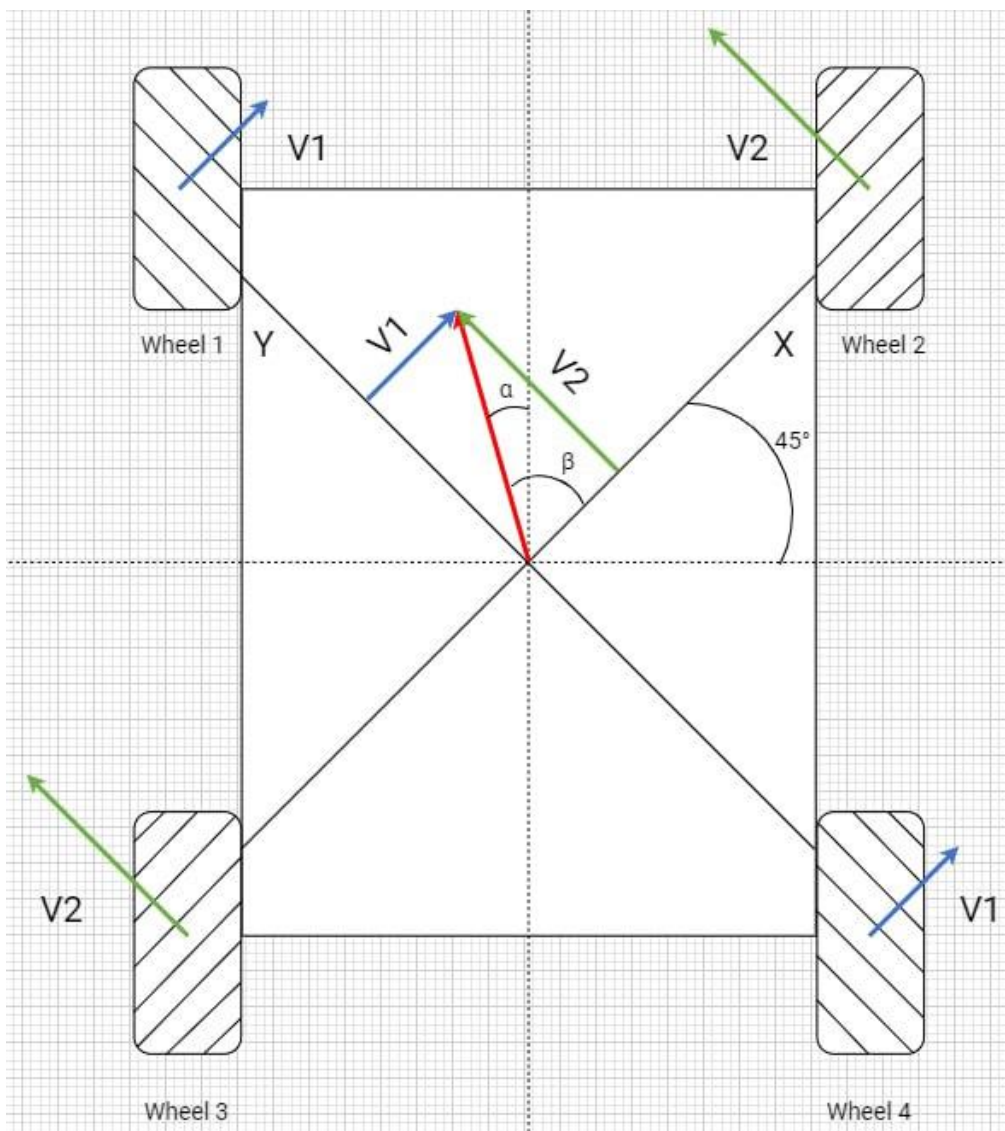


Рисунок 2.9 – Модель платформи на механум колесах

Вважаємо, що перше та друге механум колеса є передні, а третє та четверте задні колеса. Також слід зауважити, що перше та третє колеса розташовані однаково, якщо враховувати розташування роликів. На рисунку вони схематично позначені невеликими відрізками під кутом 45 градусів до осі обертання колеса. Таке розташування коліс призведе до того, що обертаний напрямок руху

платформи буде складатися з двох векторів $\vec{v}_1$ і $\vec{v}_2$, де перша швидкість (сині стрілки) – це швидкість, яку отримує платформа при обертанні першого й четвертого коліс, а друга швидкість (зелені стрілки) - від другого й третього коліс. Таким чином платформа, як ціле, буде рухатися зі швидкістю

$$\vec{v} = \vec{v}_1 + \vec{v}_2, \quad (2.1)$$

яка зображена на рисунку 2.9 червоною стрілкою.

Для зручності виберемо осі координат під кутом 45 градусів (див. рисунок 2.9), тоді швидкості $\vec{v}_1$ і $\vec{v}_2$ будуть також відповідними компонентами для повної швидкості $\vec{v}$. Будемо вказувати напрямок руху платформи за допомогою кута α , який вимірюватимемо від вертикалі вліво для негативних значень цього кута й позитивних для відхилення вправо. Тоді при русі платформи прямо вперед, цей кут дорівнює нулю, якщо платформа рухається одночасно вперед і вліво, то кут негативний і відповідно, якщо вперед і праворуч, то позитивний. Під час руху суворо назад, кут дорівнює як +180 градусів, так і -180 градусів.

За умовами завдання вважається, що кут α і модуль швидкості $|\vec{v}|$ нам задається. Завдання ставиться таким чином, що потрібно розрахувати відповідні швидкості $\vec{v}_1$ та $\vec{v}_2$. З рисунку 2.9 знаходимо

$$\begin{aligned} v_1 &= v \cdot \cos\beta \\ v_2 &= v \cdot \sin\beta \end{aligned} \quad (2.2)$$

де $\beta = \alpha + 45^\circ$.

Будемо вважати, що зв'язок між кутовою швидкістю ω_i обертання колеса та швидкістю v_i , яка передається платформі має вигляд

$$v_i = \zeta \omega_i R, \quad (2.3)$$

де $i = \{1,2\}$, а ζ - безрозмірний коефіцієнт, який знаходиться експериментально і залежить від багатьох чинників.

2.3 Модель рухомої платформи

2.3.1 Середовище розробки інтерактивної програмної моделі

Для розробки інтерактивної програмної моделі було обрано середовище Wolfram Mathematica [8, 9]. Це середовище було обране через його потужні можливості, що дозволяють ефективно вирішувати складні задачі моделювання та аналізу.

Wolfram Mathematica є високорівневою обчислювальною платформою, що пропонує широкий спектр інструментів для наукових досліджень, інженерних розробок та аналізу даних. Однією з головних переваг Mathematica є її здатність виконувати символічні обчислення. Це дозволяє працювати з рівняннями та формулами в їх загальному вигляді, що є особливо важливим для створення точних математичних моделей.

Крім того, Mathematica підтримує високоточні числові обчислення, що є критичним для моделювання реальних систем. Платформа надає потужні інструменти для візуалізації даних, включаючи побудову графіків, тривимірних моделей та інтерактивних візуалізацій. Це значно полегшує процес аналізу та інтерпретації результатів, роблячи їх більш наочними й зрозумілими.

Однією з ключових можливостей Mathematica є підтримка інтерактивних моделей. Це дозволяє користувачам змінювати параметри моделі та миттєво бачити результати, що спрощує процес моделювання та аналізу.

У порівнянні з іншими середовищами, такими як MATLAB, Simulink та Python, Wolfram Mathematica має декілька важливих переваг. MATLAB та Simulink є потужними інструментами для числового моделювання та аналізу даних, проте вони менш ефективні у виконанні символічних обчислень. Mathematica інтегрує символічні та числові обчислення в одному середовищі, що дозволяє значно спростити робочий процес.

Python є гнучкою та універсальною мовою програмування з численними бібліотеками для наукових обчислень і візуалізації, такими як NumPy, SciPy та Matplotlib. Однак, використання Python може вимагати більше часу та зусиль для

інтеграції різних бібліотек та інструментів. Mathematica забезпечує всі необхідні інструменти в одному пакеті, що робить її більш зручною для комплексних задач моделювання.

Таким чином, вибір Wolfram Mathematica як середовища для розробки інтерактивної програмної моделі є обґрунтованим завдяки її потужним можливостям для символічних і числових обчислень, інтерактивної візуалізації та моделювання динамічних систем.

Для студентів ХНУ ім. В.Н. Каразіна доступна безкоштовна навчальна ліцензія.

2.3.2 Програмна модель рухомої платформи

На основі побудованої математичної моделі, будуюмо програмну модель яка зображено на рисунку 2.10

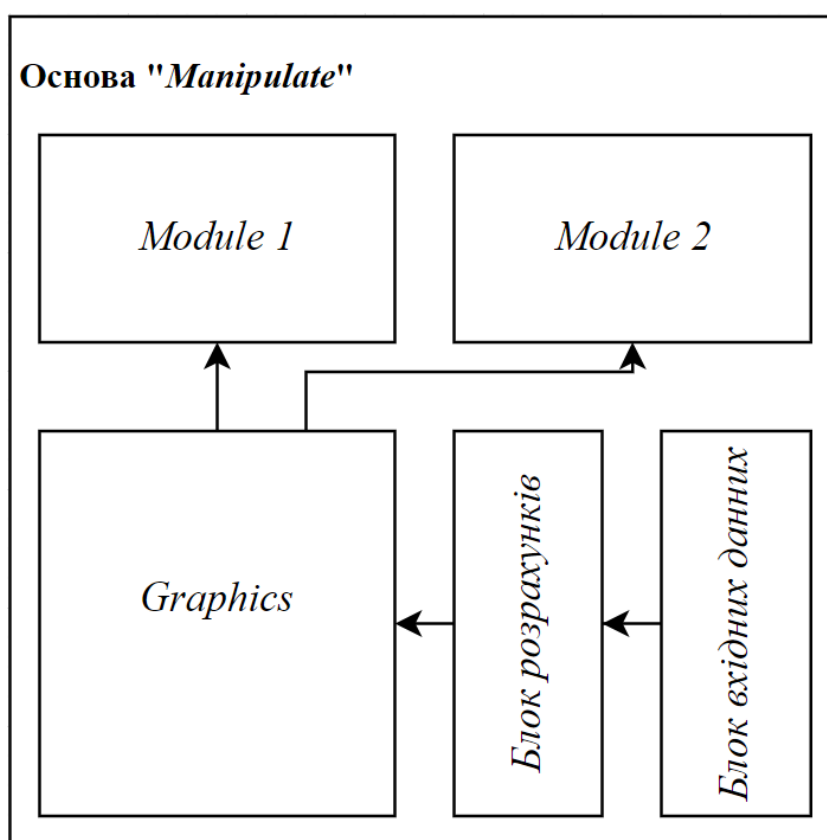


Рисунок 2.10 – Програмна модель рухомої платформи

Уся програма лежить в основній функції Manipulate [11] - потужному інструменті для візуалізації та взаємодії з динамічними об'єктами та параметрами. У цій функції лежать блоки: «Module 1» та «Module 2», блок «Graphics», блок розрахунків та блок для введення даних.

Значення кута α та модуля швидкості $|\vec{v}|$ задаються через блок вхідних даних. Вони потрапляють до блоку розрахунків, який за формулами 2.1 та 2.2 знаходить значення $\vec{v}_1$ та $\vec{v}_2$. Результати розрахунків передаються в блок «Graphics» який виводить інформацію про коефіцієнти кожного колеса, а також за допомогою блоків «Module 1» та «Module 2», візуалізує стрілки на моделі.

Кожний блок «Module» відповідає за два види візуалізації стрілок: «Module 1» відповідає за відображення зеленої та синьої стрілок біля червоної, для візуального представлення розрахунків; «Module 2» відповідає за стрілки безпосередньо на колесах.

Результатом роботи програмної моделі є візуалізація методу керування рухомої платформи на меканум-колесах яка представлена на рисунку 2.11

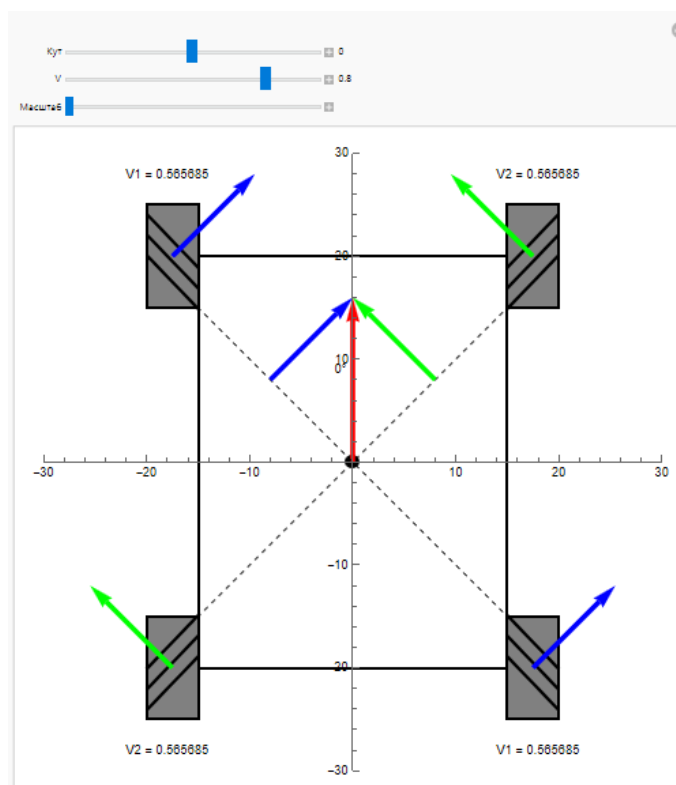


Рисунок 2.11 – Інтерактивна модель платформи

Модель дозволяє візуалізувати напрямок руху платформи, коефіцієнти швидкості та напрямки кожного колеса в залежності від вхідних параметрів. Код використовує інструменти Mathematica для створення інтерактивної візуалізації.

2.4 Тестування моделі

Створивши програмну модель, треба її протестувати. Протестуємо її на трьох ознаках: правильність розрахунків, мінімальні та максимальні вхідні дані та правильність візуального відображення.

Тест на правильність розрахунків. Нехай буде виставлений один із ідеальних діагональних напрямків (Наприклад діагональний рух вгору-вправо $\alpha = 45^\circ$ зображеного на рисунку 2.7) та визначений вхідний модуль швидкості (Наприклад $|\vec{v}| = 0,8$). Тоді за формулами 2.1 та 2.2 знаходимо:

$$\begin{aligned} v_1 &= 0,8 * \cos(45 + 45) = 0,8 \\ v_2 &= 0,8 * \sin(45 + 45) = 0 \end{aligned} \quad (2.3)$$

На рисунку 2.12 зображено результати роботи програмної моделі при заданих вхідних даних.

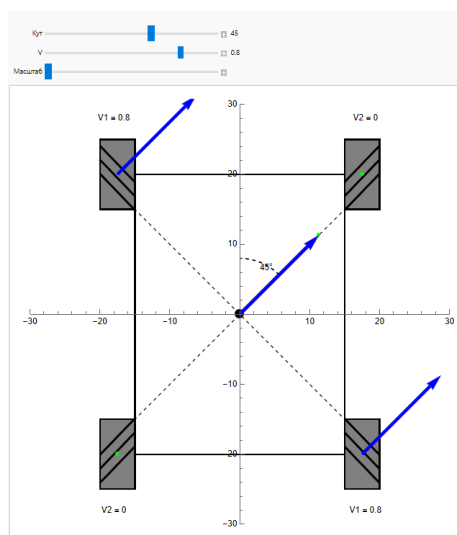


Рисунок 2.12 – Результати візуалізації моделі

У результаті, розраховані значення v_1 та v_2 (формула 2.3) відповідають значенням, зображеними на рисунку 2.12.

Тест на мінімальні та максимальні вхідні дані. Діапазон вхідного кута α дорівнює від -180° до 0 (для руху в ліву сторону) та від 0 до 180° (для руху в праву сторону). Діапазон вхідного модуля швидкості від 0 до 1 .

Оскільки положення стрілки при значеннях -180° та 180° однакове, тоді немає значення який кут обирати. Нехай для перевірки на мінімальні та максимальні значення вхідний кут $\alpha = -180^\circ$, а модуль швидкості $|\vec{v}| = 0$ (для мінімуму) та $|\vec{v}| = 1$ (для максимуму). Отримуємо результати, зображені на рисунку 2.13

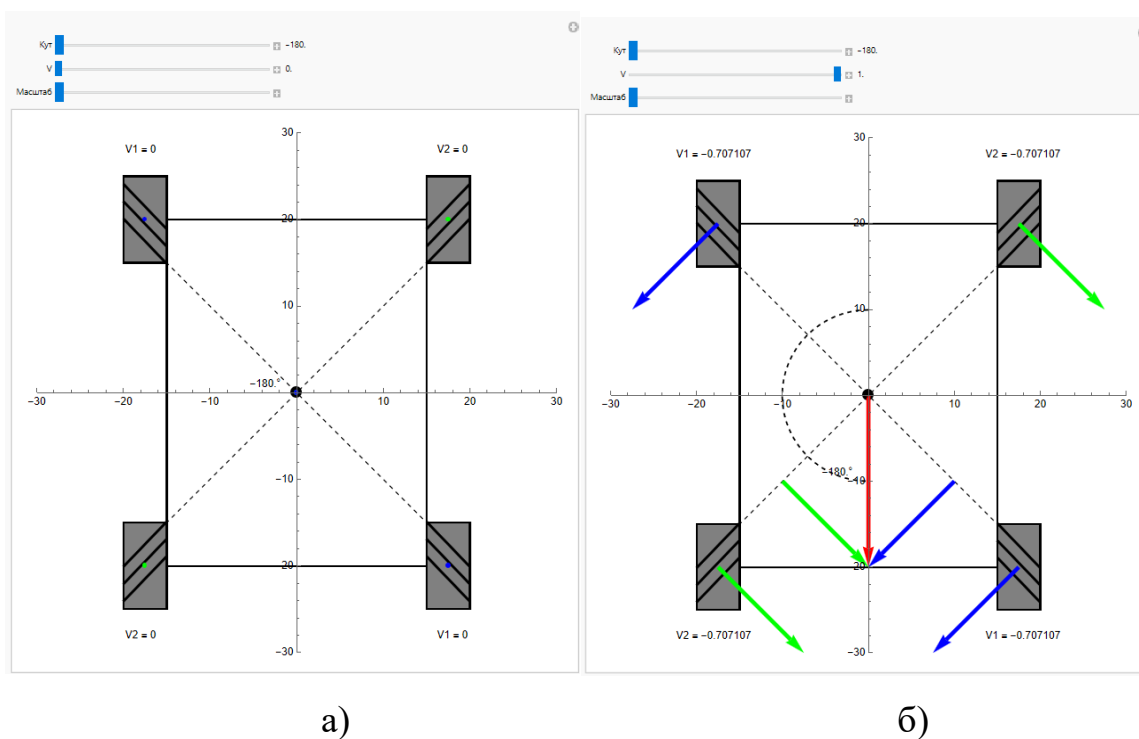


Рисунок 2.13 – Результати візуалізації моделі при $\alpha = -180^\circ$,

а) $|\vec{v}| = 0$, б) $|\vec{v}| = 1$

Модель реагує на зміни мінімальних і максимальних вхідних даних, при цьому коефіцієнти швидкості залишаються в межах допустимих границь.

Тест на правильність візуального відображення. Нехай задано бічний рух вправо (рисунок 2.4). Червона стрілка, яка вказує на сторону напрямку руху платформи лежить під кутом $\alpha = 90^\circ$, до передньої сторони платформи. Модуль

швидкості дорівнюватиме $|\vec{v}| = 0,8$. Отримуємо результат, зображений на рисунку 2.14

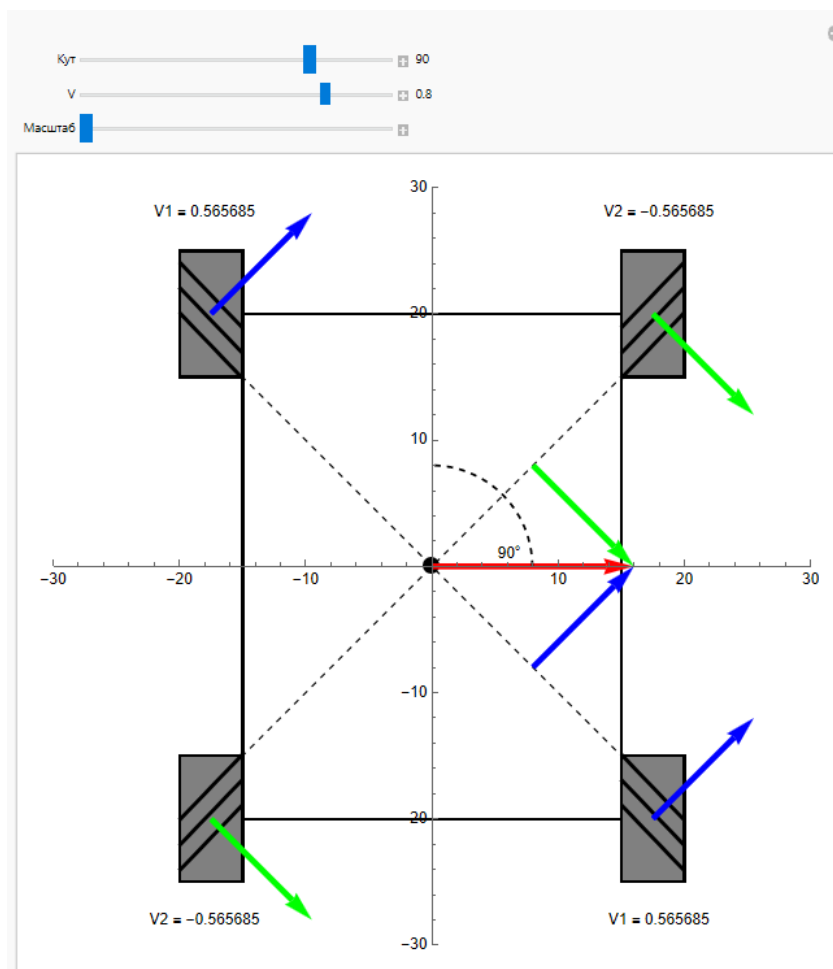


Рисунок 2.14 – Результати візуалізації моделі при $\alpha = 90^\circ$, $|\vec{v}| = 0,8$

Стрілки з рисунку 2.14 вказують в такий самий напрямок, що і на рисунку 2.4.

2.5 Обертання

2.5.1 Основи обертального руху платформи на меканум колесах

Цікавою темою дослідження є обертання платформи. Розглянемо основні напрями обертання.

Обертання за годинниковою стрілкою (праворуч). Для того, щоб платформа почала обертатися, треба щоб ліва пара колес оберталася вперед, а права пара в протилежному напрямку. Схематично зображено на рисунку 2.13

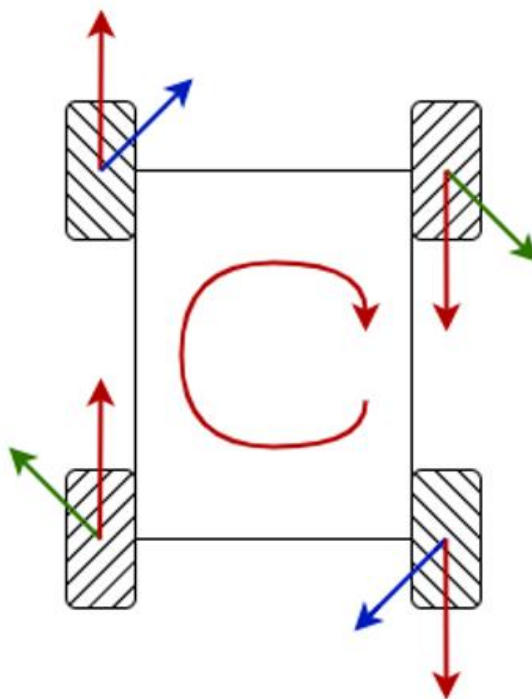


Рисунок 2.13 – Обертання за годинниковою стрілкою (праворуч)

Обертання проти годинникової стрілки (ліворуч). Для того, щоб платформа почала обертатися ліворуч, треба щоб права пара колес оберталася вперед, а ліва пара в протилежному напрямку. Схематично зображено на рисунку 2.14

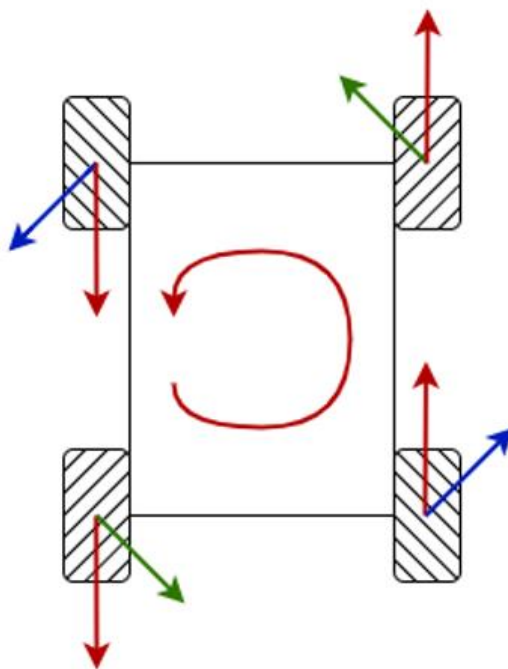


Рисунок 2.14 – Обертання проти годинникової стрілки (ліворуч)

Хоча принцип обертання платформи виглядає досить простим і ефективним, однак у загальному випадку отримати ідеальне обертання досить важко.

2.5.2 Геометрія обертального руху

У даному підрозділі ідеальним обертанням рухомої платформи будемо вважати те обертання, при якому не виникають сили бокового тертя.

Для того, щоб розібратися з обертанням платформи на механічних колесах, треба згадати, як відбувається поворот платформи на звичайних колесах [12].

Коли платформа здійснює поворот, внутрішні колеса описують менший радіус повороту, ніж зовнішні колеса. Це відбувається через те, що внутрішні колеса знаходяться ближче до центру повороту. Для забезпечення плавного повороту треба, щоб перпендикуляри опущені від векторів швидкості всіх чотирьох колес збігалися в точці, навколо якої відбувається поворот (Рисунок 2.15).

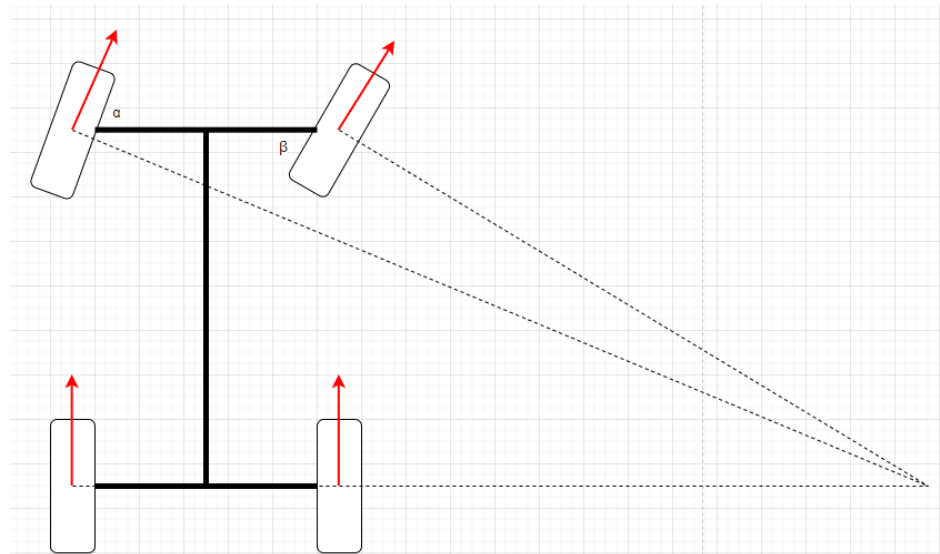


Рисунок 2.15 – Плавний поворот навколо точки

Обертання – це певний вид повороту, тому геометрія обертання платформи на меканум-колесах схожа на геометрію повороту платформи на звичайних колесах. Взявши за основу принцип обертання за годинниковою стрілкою (праворуч), провели перпендикуляри (Рисунок 2.16).

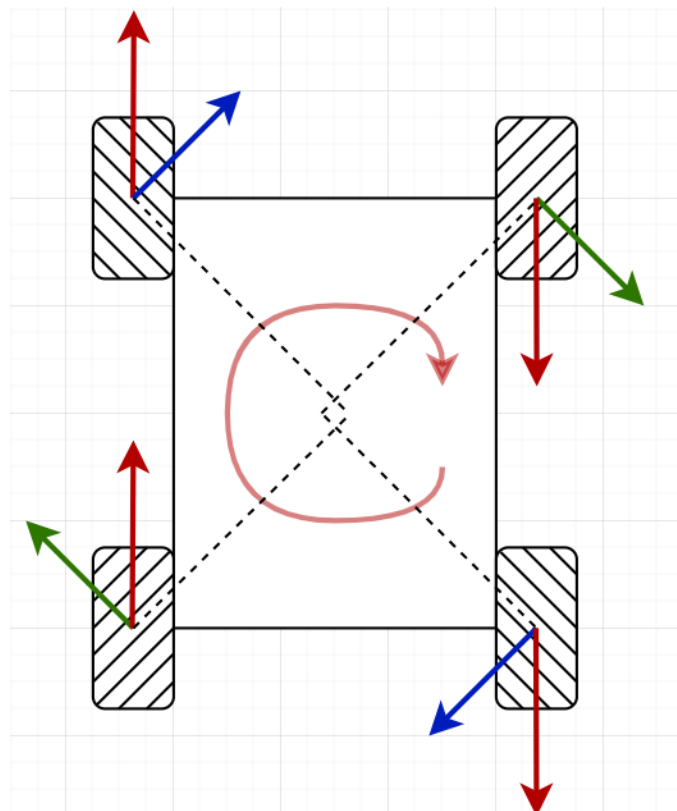


Рисунок 2.16 – Геометрія обертання платформи на меканум-колесах

Як видно з рисунку 2.16, опущені перпендикуляри (пунктирна лінія) не сходяться в одній точці, а утворюють квадрат. Саме тому платформа в загальному випадку розміщення коліс не зможе досягти ідеального обертання.

Для досягнення ідеального обертання на платформі з механум-колесами треба змінити її конструкцію, а саме форму основи розроблювати так, щоб виконувалася умова з перпендикулярами (Рисунок 2.17).

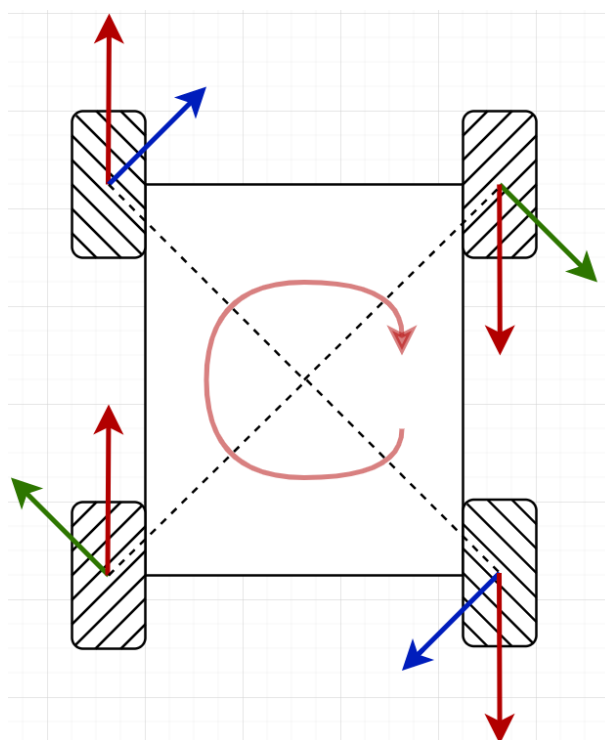


Рисунок 2.17 – Конструкція, яка виконує умови ідеального обертання

Оскільки така конструкція є окремим випадком, обертання не було додано до розроблювальної моделі.

Висновки за розділом 2

У даному розділі було розглянуто основні напрями руху платформи на механум-колесах. Зокрема, було побудовано математичну модель цієї платформи, яка дозволила створити інтерактивну програмну модель на мові Wolfram Language для візуалізації методу керування платформою на механум-колесах.

Дослідження за допомогою моделі показало, що обертання платформи на механум-колесах можливе. Однак, у загальному випадку виникають сили бокового тертя, які можуть призвести до значного зношення роликів на колесах та руйнування самих коліс. Це важливий аспект, який потрібно враховувати при проектуванні та експлуатації механізмів на механум-колесах.

РОЗДІЛ 3

МОДЕЛЮВАННЯ КЕРУВАННЯ DC-МОТОРАМИ

3.1 Опис основних механічних актуаторів

Механічні актуатори є ключовими компонентами багатьох систем автоматизації та робототехніки. Вони перетворюють електричну енергію в механічний рух, забезпечуючи виконання різноманітних завдань, від простих лінійних переміщень до складних багатокоординатних маневрів. Розуміння їхньої конструкції та принципів роботи є основою для ефективного проектування та керування роботизованими системами.

Основними типами механічних актуаторів є лінійні та обертальні. Лінійні актуатори забезпечують прямолінійний рух, який може бути корисним для задач, які вимагають точного переміщення об'єктів уздовж однієї осі. Вони використовуються в різних галузях, таких як машинобудування, будівництво та медицина. Обертальні актуатори, навпаки, забезпечують обертальний рух і застосовуються там, де необхідний поворот або обертання компонентів. Вони часто використовуються в робототехніці, транспортних засобах і промислових машинах.

До уваги слід взяти обертальні актуатори, які необхідні для обертання коліс розроблюваної платформи. У наступних підрозділах будуть розглянуті основні види обертальних актуаторів, їхні характеристики та особливості використання.

3.1.1 STEPPER Motor

STEPPER Motor [13] (з англ. STEPPER Motor – кроковий двигун) – це електричний двигун, у якому імпульсне живлення струмом спричиняє рух ротора не в безперервному обертанні, а на визначений кут під час кожного імпульсу. Завдяки цьому кут обертання ротора залежить від кількості імпульсів

струму, а його кутова швидкість пропорційна частоті імпульсів, помноженій на кут повороту ротора за один цикл роботи двигуна (Рисунок 3.1).

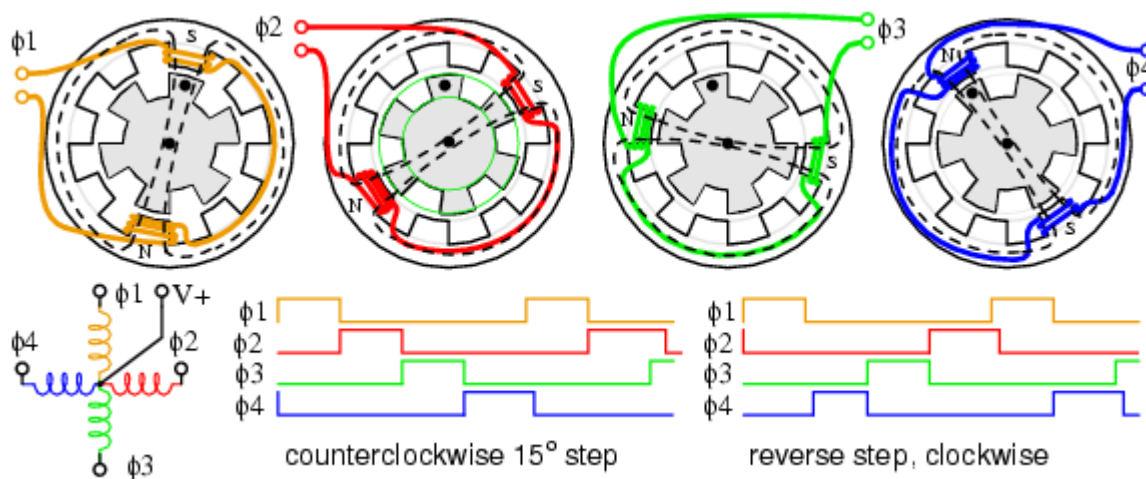


Рисунок 3.1 – Принцип роботи крокового двигуна

Кут обертання ротора під впливом одного імпульсу може різнитися залежно від конструкції двигуна, зазвичай знаходиться в межах від кількох градусів до десятків градусів. Крокові двигуни призначені для роботи в широкому діапазоні швидкостей: від частки оберту за хвилину до кількох тисяч обертів за хвилину.

Крокові двигуни можуть бути класифіковані за кількістю фаз, що визначає їхню продуктивність і складність керування. Основними типами є двофазні, трифазні та п'ятифазні двигуни. Двофазні крокові двигуни є найбільш поширеними завдяки своїй простоті та надійності, що робить їх ідеальними для багатьох загальних застосувань, таких як 3D-принтери та невеликі роботи. Трифазні крокові двигуни забезпечують вищу плавність руху та ефективність, що робить їх придатними для прецизійних систем автоматизації. П'ятифазні крокові двигуни відрізняються ще більшою плавністю та точністю обертання, але є складнішими в керуванні та дорожчими, що обмежує їх використання у високоточних системах, таких як прецизійні станки та високоточні робототехнічні системи.

Основними перевагами крокових двигунів є:

- Точність позиціонування. Крокові двигуни дозволяють точно контролювати положення ротора, що забезпечує високоточне позиціонування без зворотного зв'язку.
- Миттєвий старт і зупинка. Крокові двигуни здатні починати рух та зупиняти його миттєво, що корисно для застосувань, де потрібна швидка реакція.
- Високий крутний момент при низьких швидкостях. Крокові двигуни забезпечують високий крутний момент при низьких швидкостях, що корисно для керування важкими навантаженнями.
- Можливість роботи без зворотного зв'язку. Оскільки крокові двигуни контролюються кількістю імпульсів, зворотний зв'язок для визначення позиції часто не потрібний.

Основними недоліками є:

- Вібрації та шум. Крокові двигуни можуть створювати вібрації та шум, особливо при роботі на високих швидкостях, що може впливати на точність і довговічність механічних систем.
- Низька ефективність на високих швидкостях. При високих швидкостях крокові двигуни можуть втрачати ефективність і не забезпечувати достатнього крутного моменту.
- Ризик пропуску кроків. Якщо навантаження на двигун перевищує його можливості, двигун може пропускати кроки, втрачаючи точність позиціонування.
- Високе енергоспоживання. Крокові двигуни можуть споживати більше енергії, оскільки котушки статора залишаються під напругою навіть у стані спокою.
- Межа робочої частоти. При певних високих частотах імпульсів керування кроковий двигун може втратити синхронність і контроль.

Вибір крокового двигуна залежить від вимог до точності позиціонування, швидкості та робочого середовища.

Переваги крокових двигунів роблять їх популярними в автоматизації, робототехніці та інших високоточних застосуваннях.

3.1.2 BLDC Motor

BLDC Motor [14] (з англ. Brushless DC Motor - безщітковий двигун постійного струму) – це тип електричного двигуна, який використовує постійні магніти на роторі та обмотки на статорі для створення руху. Він називається безщітковим через відсутність механічних щіток і колектора, які використовуються в традиційних двигунах постійного струму для передачі електричної енергії до ротора.

Двигун працює за принципом взаємодії між обертовим магнітним полем статора та постійними магнітами на роторі. Контролер подає електричний струм до обмоток статора, створюючи обертове магнітне поле. Це поле взаємодіє з магнітним полем постійних магнітів на роторі, викликаючи його обертання. Контролер використовує датчики положення, такі як датчики Hall-ефекту, для визначення позиції ротора й регулювання струму до обмоток, щоб підтримувати обертання ротора в потрібному напрямку та зі швидкістю, залежно від вимог системи (Рисунок 3.2).

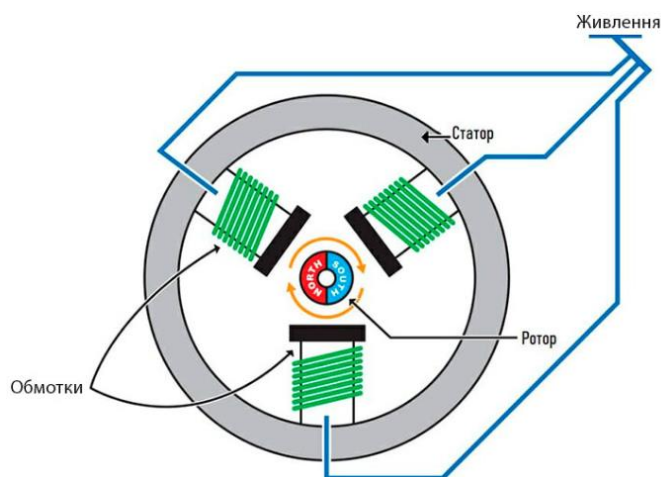


Рисунок 3.2 – Модель безщіткового двигуна

BLDC двигуни мають як переваги, так і недоліки, які впливають на їх застосування в різних галузях:

Основними перевагами безщіткових двигунів постійного струму є:

- Вони більш ефективні, оскільки відсутність щіток зменшує енергоспоживання.
- Вони тихіші через відсутність механічного контакту щіток із колектором.
- Довший термін служби завдяки відсутності зношувальних деталей, таких як щітки.
- Високий крутний момент та широкий діапазон швидкостей роблять їх універсальними.
- Точність контролю позиції та швидкості дозволяє використовувати їх у багатьох сучасних системах.

Основними недоліками безщіткових двигунів постійного струму є:

- BLDC-двигуни можуть бути дорогими через використання складніших електронних керувань.
- Вони вимагають більш складної системи керування, що може ускладнювати їхнє застосування.
- Залежність від контролера та датчиків може обмежувати надійність системи, особливо в складних умовах експлуатації.

Завдяки високій надійності та точному керуванню, вентильні двигуни використовуються в різноманітних сферах в тому числі в робототехніці та RC-моделях.

3.1.3 DC Motor

DC Motor [15] (з англ. DC Motor - Двигун постійного струму) – це електрична машина, яка перетворює електричну енергію постійного струму в механічну енергію.

У двигуні постійного струму взаємодія струму в обмотці ротора (якоря) з основним магнітним полем призводить до появи електромагнітного обертового моменту завдяки явищу електромагнітної індукції.

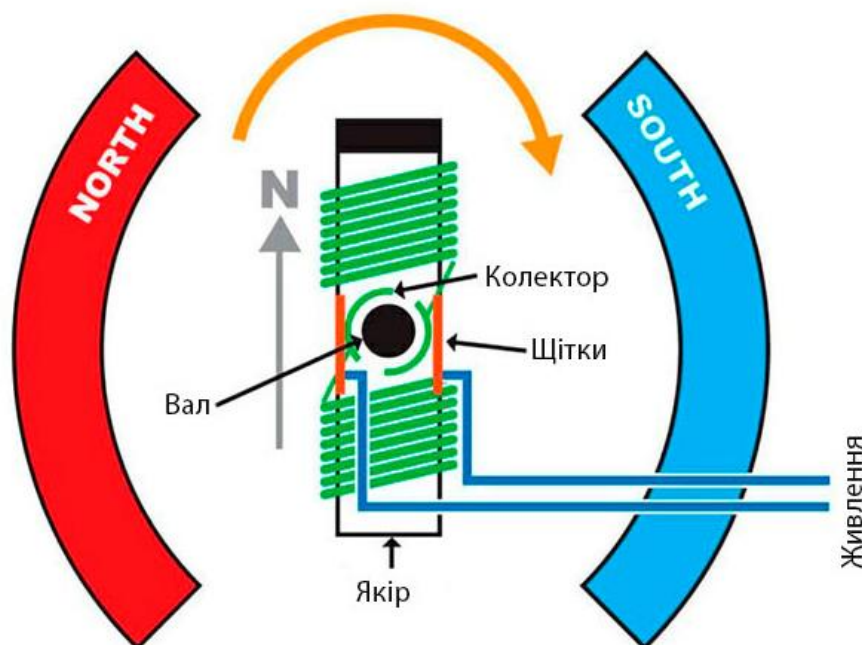


Рисунок 3.3 – Модель двигуна постійного струму

Розрізняють двигуни постійного струму з незалежним та залежним збудженням. Потужність ДПС може варіюватися від часток вата до кількох тисяч кіловат, а ККД – від 0,02-0,03 до 0,93.

Двигуни постійного струму дозволяють плавно й в широких межах керувати швидкістю обертання, змінюючи струм у колі ротора через додатковий електричний опір, регулюючи живильну напругу або магнітний потік реостатом.

ДС-двигуни мають як переваги, так і недоліки, які впливають на їх застосування в різних галузях:

Основними перевагами двигунів постійного струму є:

- Просте керування швидкістю. ДС-двигунами легко керувати за допомогою зміни напруги або струму в колі, що дозволяє досягати різних швидкостей обертання.

- Високий пусковий крутний момент. DC-двигуни здатні забезпечити високий крутний момент при пуску, що робить їх корисними для застосувань, де потрібен великий початковий імпульс.
- Легкість обертання в обох напрямках. DC-двигуни легко змінюють напрямок обертання, що спрощує їх використання в різних системах.
- Простота конструкції: DC-двигуни мають відносно просту конструкцію, що спрощує їх обслуговування та ремонт.
- Висока ефективність. У багатьох випадках DC-двигуни працюють з високою ефективністю, особливо у діапазоні низьких і середніх швидкостей.

Основними недоліками двигунів постійного струму є:

- Високий знос щіток і колектора. Через контакт між щітками та колектором, ці частини можуть швидко зношуватися, що вимагає регулярного обслуговування.
- Складність у обслуговуванні. Оскільки двигуни мають рухомі частини, такі як колектор і щітки, їх обслуговування може бути складнішим.
- Нестабільність у високих швидкостях. У деяких випадках, особливо без систем контролю, DC-двигуни можуть бути нестабільними при високих швидкостях.
- Втрата продуктивності при зростанні навантаження. Двигуни можуть втрачати продуктивність при великих навантаженнях або при тривалій роботі.
- Електричний шум. Працюючі щітки та колектор можуть створювати електричний шум, який може впливати на роботу інших електронних пристроїв.

Попри ці недоліки, DC-двигуни залишаються популярним вибором у багатьох галузях завдяки своїм перевагам і простоті використання. Завдяки простоті та дешевизні будемо розглядати DC-двигуни.

3.2 Керування DC-двигунами.

3.2.1 H-міст

Двигуни постійного струму (DC-двигуни) – це конструктивно найпростіші пристрої. Для запуску подібних двигунів їх необхідно підключити до джерела постійної напруги й вал такого двигуна повинен відразу почати обертання. Причому зміни напрямки обертання «плюс» і «мінус» джерела живлення необхідно поміняти місцями

Для керування такими двигунами зазвичай використовують пристрій під назвою H-міст.

H-міст [16] – це електронна схема, яка дає можливість змінювати полярність прикладеного до навантаження напруги. У нашому випадку будемо використовувати його для керування моторами.

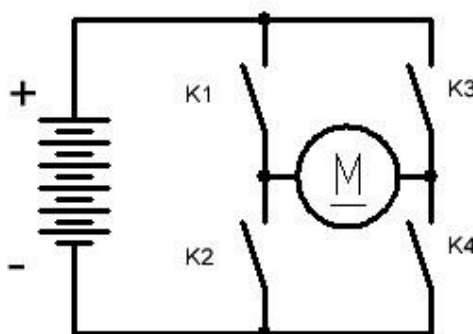


Рисунок 3.4 - H-міст

H-міст може бути зібраний зі звичайних механічних вимикачів, електромеханічних вимикачів чи електронних вимикачів.

Розглянемо принцип дії схеми.

Якщо K2 і K3 замкнуті, а K1 і K4 розімкнуті, таке включення еквівалентно лівій верхньої схеми. Якщо K1 і K4 замкнуті, K2 і K3 розімкнуті, то правою.

Якщо включити K1 і K3 при K2 і K4 розімкнених, то при такому включенні обмотка двигуна буде замкнута, це сприятиме якнайшвидшому гальмуванню

ротора двигуна. Аналогічна ситуація буде при включенні К2 та К4, при розімкнених К1 та К3.

Зауважимо, якщо К1 і К2 (чи К3, К4), то вийде коротке замикання, що є вкрай неприпустимим явищем. Усі режими роботи Н-мосту наведені у таблиці 3.1

Таблиця 3.1

Режими роботи Н-мосту

К1	К2	К3	К4	Результат
1	0	0	1	Вал крутиться проти годинникової стрілки
0	1	1	0	Вал крутиться за годинниковою стрілкою
0	0	0	0	Вал вільно обертається за інерцією
0	1	0	1	Вал гальмується
1	0	1	0	Вал гальмується
1	1	0	0	Коротке замикання джерела живлення
0	0	1	1	Коротке замикання джерела живлення

Зі схеми (Рисунок 3.4) видно, що ключі К1 та К3 підключають вихід двигуна до «плюсу» живлення, а ключі К2 та К4 – до «мінуса» живлення. Це означає, що для створення мосту нам необхідні два верхні (К1, К3) і два нижні (К2, К4) ключі.

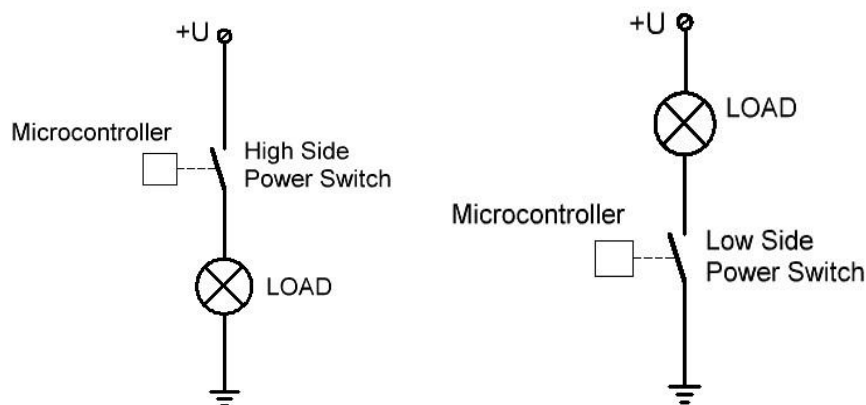


Рис. 3.5 Верхній ключ (ліворуч), нижній ключ (праворуч).

Терміни верхній та нижній ключ пояснює рисунок 3.2.2 на якому видно, що верхній ключ підключає навантаження до плюсу живлення, а нижній – до мінуса. Верхніми та нижніми ключами можуть виступати електромеханічні реле, біполярні та польові (мосфети) транзистори, транзисторні зборки Дарлінгтона та інші пристрої. Для управління навантаженням, що працює на змінному струмі, зазвичай використовують тиристори, і симістори (симетричні тиристори).

Конструктивно електронний Н-міст можна зібрати з різних дискретних елементів, а можна використовувати готове складання у вигляді мікросхеми. У будь-якому випадку така схема називається – драйвер.

3.2.2 Драйвер L293D

Одним із найвідоміших драйверів є L293D [17] у корпусі DIP16 зображений на рисунку 3.6



Рисунок 3.6 - Драйвер L293D

DIP від англійської dual in-line package - назва типу корпусу, що використовується для мікросхем. Корпуси цього типу мають прямокутну форму та два ряди виводів з довгих сторін. Число 16 означає кількість виводів, яка може бути від 4 до 64.

Таблиця 3.2

Основні характеристики драйвера L293D.

Wide Supply-Voltage Range: 4.5 V to 36 V	Напруги живлення: від 4,5 до 36В
Separate Input-Logic Supply	Окреме живлення логічної частини
Internal ESD Protection	Електростатичний захист
High-Noise-Immunity Inputs	Перешкодостійкі входи
Output Current 1 A Per Channel (600 mA for L293D)	Вихідний струм 600 мА з одного каналу
Peak Output Current 2 A Per Channel (1.2 A for L293D)	Максимальний струм 1.2 А з одного каналу
Output Clamp Diodes for Inductive Transient Suppression (L293D)	Діоди для підключення індуктивного навантаження

З цієї таблиці стає зрозуміло, що логічні сигнали керування розділені від живлення двигуна. При цьому підтримуються двигуни з напругою живлення від 4.5 до 36 вольт та постійним струмом до 600 мА=0.6А (і до 1.2 А в піку).

3.2.3 Драйвер TB6612FNG

TB6612FNG [18] – це двоканальний Н-міст драйвер для двох колекторних двигунів або одного крокового зображений на рисунку 3.7



Рисунок 3.7 – Драйвер TB6612FNG

Модуль зібраний на мікросхемі TB6612FNG від Toshiba, це двоканальний мостовий драйвер, мікросхема функціонально схожа з популярним драйвером L293D, але на відміну від нього, у драйвері використані MOSFET транзистори, драйвер також має захист від перепаду температур та короткого замикання. Може керуватись логічним рівнем 3.3 вольта.

Таблиця 3.3

Основні характеристики драйвера L293D.

• Supply Voltage; $V_M = 15\text{ V}$ (Max.) • Output Current; $I_{out} = 1.2\text{ A (avg) } / 3.2\text{ A (Peak of continuous pulse)}$ • Output Low-On Resistor; $0.3\ \Omega$ (Upper+Lower Typ. @ $V_{CC} = V_M = 5\text{ V}$) • Standby (Power save) system • CW / CCW / Shortbrake / Stop function modes • Direct-PWM input terminal included \ • Built-in thermal shut down circuit (TSD), low voltage detection circuit (UVD), And over current detection circuit (ISD)	Напруга живлення ; $V_M = 15\text{ В}$ (макс.) • Вихідний струм; $I_{out} = 1,2\text{ А}$ (середнє) / $3,2\text{ А}$ (Пік безперервного імпульсу) • Вихідний резистор низького рівня; $0,3\text{ Ом}$ (Верхній + Нижній тип. @ $V_{CC} = V_M = 5\text{ В}$) • Система очікування (Енергозбереження). • Режими функції CW / CCW / Shortbrake / Stop • Вхідний термінал прямого ШІМ включений • Вбудована схема термічного відключення (TSD), схема виявлення низької напруги (UVD), І схема виявлення перевищення струму (ISD)
--	--

З таблиці 3.3 стає зрозуміло, що драйвер має низький вихідний опір та широкий діапазоном напруги живлення до 15 В. Драйвер може забезпечувати середній вихідний струм до 1,2 А на канал із піковим струмом до 3,2 А. Його малий розмір (SSOP16) дозволяє використовувати його в компактних проектах.

3.3 Тестування драйверів в ISIS Proteus

Для тестування усіх драйверів буде використовуватися програмне забезпечення ISIS Proteus [19]. ISIS Proteus – це програмний інструмент для проектування електронних схем, симуляції та макетування плат. Для керування

швидкістю обертання електромоторів був використаний мікроконтролер ATmega2560. Під кожний макет була написана тестова програма.

3.3.1 Тестування макету L293D

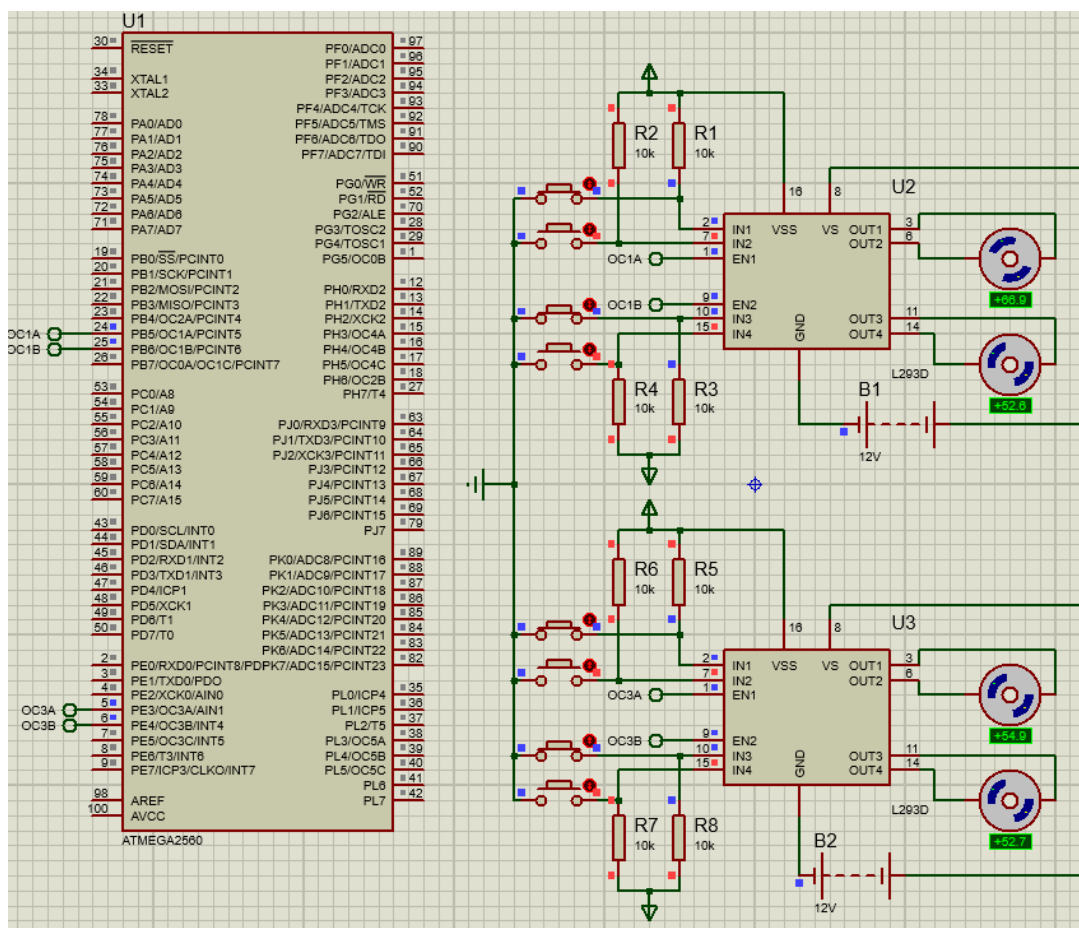


Рисунок 3.8 – Макет драйверу L293D в середовищі ISIS Proteus

В схемі, яка зображена на рисунку 3.8 було використано дві мікросхеми L293D для керування двома парами моторів. Кожна зі схем має дві пари входів для керуючих сигналів та дві пари виходів для підключення електромоторів. Крім того, у L293D є два входи для включення кожного з драйверів, які використовуються для керування швидкістю обертання електромоторів за допомогою ШІМ (PWM).

Для підключення електромотора MOTOR1 використовуються виходи OUTPUT1 та OUTPUT2 (для мікросхеми в DIP-корпусі – контакти 3 та 6). Відповідно, MOTOR2 підключається до виходів OUTPUT3 та OUTPUT4 (контакти 11 та 14). Сигнали, що подаються на ENABLE1 (2), керують відповідним драйвером (контакти 1 та 9 відповідно). Подаючи на вхід ENABLE1 сигнал HIGH (або просто з'єднавши з плюсом джерела живлення +5V), включаємо драйвер першого мотора.

Якщо при цьому на входи INPUT1 та INPUT2 сигнали не подавати, то двигун обертатися не буде. Подаючи HIGH на INPUT1 та LOW на INPUT2, двигун почне обертатися. Якщо ж поміняти сигнали місцями, подавши на INPUT1 сигнал LOW, а на INPUT2 – HIGH, двигун обертатиметься в іншу сторону. Аналогічно діє для другого драйвера.

Вихід Vss (контакт 16) відповідає за живлення самої мікросхеми, а вихід Vs (контакт 8) – за живлення моторів. Це забезпечує розділення електроживлення для мікросхеми та для керованих нею двигунів, що дозволяє підключити електродвигуни з напругою живлення, відмінною від напруги живлення мікросхеми. Розділення електроживлення також необхідне для зменшення перешкод, викликаних стрибками напруги, пов'язаними з роботою двигунів. Чотири контакти GND (контакти 4, 5, 12, 13) потрібно з'єднати з землею, вони також забезпечують тепловідвід від мікросхеми1.

```

1  #define F_CPU 16000000UL // частота - 16MHz
2  #include <avr/io.h>      // бібліотека вводу/виводу
3
4  int main(void){
5      //Перша пара двигунів на першому 16-бітному таймері
6      DDRB |= (1<<DDB5);    //PB5=OC1A як ВИХІД
7      DDRB |= (1<<DDB6);    //PB6=OC1B як ВИХІД
8
9      //Друга пара двигунів на третьому 16-бітному таймері
10     DDRE |= (1<<DDE3);    //PB5=OC3A як ВИХІД
11     DDRE |= (1<<DDE4);    //PB6=OC3B як ВИХІД
12
13     //Налаштування режиму для першого та третього таймерів
14     TCCR1A |= (1<<WGM10) | (1<<WGM12); //режим FAST PWM №5 (восьмибітний)
15     TCCR3A |= (1<<WGM30) | (1<<WGM32); //режим FAST PWM №5 (восьмибітний)
16
17     //Встановлення дільника(N = 8, F = 7812.5 Гц., )
18     TCCR1B |= (1<<CS11); //дільник N = 8
19     TCCR3B |= (1<<CS31); //дільник N = 8
20
21     //Налаштування типу сигналу
22     TCCR1A |= (1<<COM1A1) | (1<<COM1B1); //неінвертований ШІМ OC1A та OC1B
23     TCCR3A |= (1<<COM3A1) | (1<<COM3B1); //неінвертований ШІМ OC3A та OC3B
24
25     //Швидкість першої пари коліс (від 0 до 255)
26     OCR1A = 250; // максимум 255
27     OCR1B = 205; // максимум 255
28
29     //Швидкість другої пари коліс (від 0 до 255)
30     OCR3A = 205; // максимум 255
31     OCR3B = 205; // максимум 255
32
33     while (1){}; //основна програма
34 }

```

Рисунок 3.9 – Тестова програма макету драйвера L293D в середовищі ISIS Proteus

На рисунку 3.9 зображена тестова програма, яка призначена для керування двома парами двигунів з використанням широтно-імпульсної модуляції (ШІМ). Цей код використовує таймери мікроконтролера для генерації ШІМ-сигналів, які потім можуть бути використані для керування швидкістю двигунів. Значення OCR1A, OCR1B, OCR3A та OCR3B визначають ширину імпульсів ШІМ, що впливає на швидкість обертання двигунів. Максимальне значення для цих регістрів – 255, що відповідає 100% заповнення імпульсу.

3.3.2 Тестування макету TB6612FNG

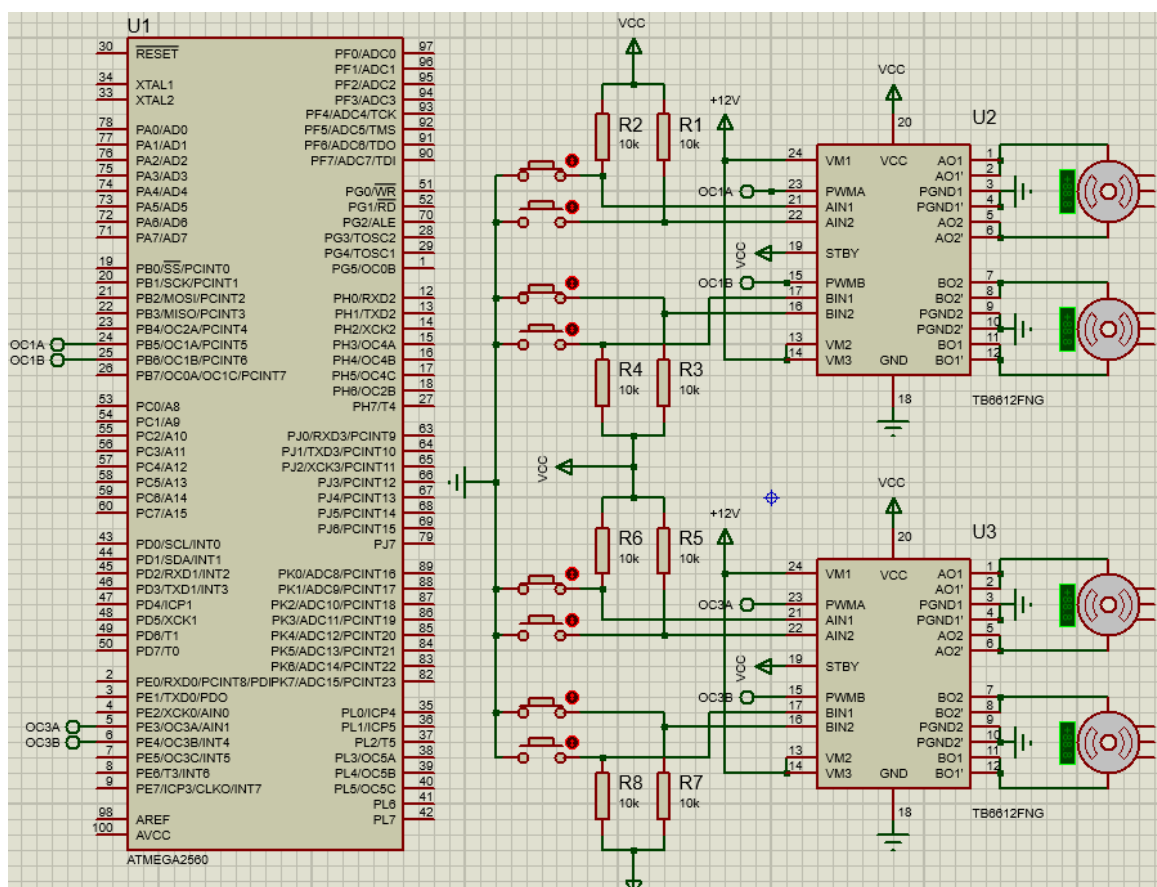


Рисунок 3.10 – Макет драйверу TB6612FNG в середовищі ISIS Proteus

В схемі на рисунку 3.10 було використано дві мікросхеми TB6612FNG для керування двома парами моторів. Для керування двигуном використовуються вхідні контакти AIN1/AIN2 для першого каналу та BIN1/BIN2 для другого каналу.

Вхідні сигнали на ці контакти визначають напрямок обертання двигуна: подача сигналу HIGH на AIN1 та LOW на AIN2 змусить двигун обертатися в одному напрямку, тоді як зміна сигналів на протилежні змусить двигун обертатися в іншому напрямку. Аналогічно працює й для другого каналу з BIN1 та BIN2.

Контакти PWMA та PWMB використовуються для керування швидкістю обертання двигунів за допомогою ШІМ-сигналів. Широко-імпульсна модуляція дозволяє змінювати швидкість двигунів, змінюючи заповнення імпульсу.

Контакт STBY використовується для включення або виключення роботи двигунів. Коли на цей контакт подається сигнал LOW, двигуни переходять у режим очікування і не обертаються.

Контакти VM та VCC забезпечують живлення для двигунів та логічної частини мікросхеми відповідно. Важливо, що VM завжди повинен бути вищим за VCC, щоб мікросхема працювала правильно.

Контакти GND слід з'єднати з загальним заземленням системи, щоб забезпечити правильну роботу мікросхеми.

TB6612FNG також має вбудовані захисні функції, такі як захист від перегріву та низької напруги, що допомагає запобігти пошкодженню мікросхеми та двигунів при неправильному використанні.

```

1  #define F_CPU 16000000UL // частота - 16MHz
2  #include <avr/io.h>      // Бібліотека вводу/виводу
3
4  int main(void){
5      //Перша пара двигунів на першому 16-бітному таймері
6      DDRB |= (1<<DDB5);    //PB5=OC1A як ВИХІД
7      DDRB |= (1<<DDB6);    //PB6=OC1B як ВИХІД
8
9      //Друга пара двигунів на третьому 16-бітному таймері
10     DDRE |= (1<<DDE3);    //PB5=OC3A як ВИХІД
11     DDRE |= (1<<DDE4);    //PB6=OC3B як ВИХІД
12
13     //Налаштування режиму для першого та третього таймерів
14     TCCR1A |= (1<<WGM10); //режим FAST PWM №5 (восьмибітний)
15     TCCR1B |= (1<<WGM12);
16
17     TCCR3A |= (1<<WGM30); //режим FAST PWM №5 (восьмибітний)
18     TCCR1B |= (1<<WGM32);
19
20
21     //Встановлення дільника(N = 1, F = 62500 Гц., F_CPU / N * (1 + TOP))
22     TCCR1B |= (1<<CS10); //дільник N = 1
23     TCCR3B |= (1<<CS30); //дільник N = 1
24
25     //Налаштування типу сигналу
26     TCCR1A |= (1<<COM1A1) | (1<<COM1B1); //неінвертований ШІМ OC1A та OC1B
27     TCCR3A |= (1<<COM3A1) | (1<<COM3B1); //неінвертований ШІМ OC3A та OC3B
28
29     //Швидкість першої пари коліс (від 0 до 255)
30     OCR1A = 254; // максимум 255
31     OCR1B = 254; // максимум 255
32
33     //Швидкість другої пари коліс (від 0 до 255)
34     OCR3A = 250; // максимум 255
35     OCR3B = 250; // максимум 255
36     while (1){}; //основна програма
37 }

```

Рисунок 3.11 – Тестова програма макету драйвера TB6612FNG в середовищі
ISIS Proteus

Тестова програма, яка зображена на рисунку 3.11 призначена для керування двома парами двигунів за допомогою широтно-імпульсної модуляції (ШІМ). Використовуючи таймери 1 і 3, код генерує ШІМ-сигнали, що регулюють швидкість обертання двигунів через порти PB5, PB6, PE3 і PE4. Налаштування таймерів на режим швидкої ШІМ та встановлення неінвертованих вихідних сигналів дозволяють точно контролювати швидкість кожного двигуна. Програма встановлює значення ширини імпульсів для кожної пари двигунів.

Висновки до розділу 3

У цьому розділі було проведено детальний аналіз та порівняння різних типів двигунів для визначення найбільш прийняттого варіанту необхідного розроблюваної моделі.

В результаті було обрано DC-двигун, як найдешевший та найпростіший варіант з-поміж інших.

Далі, було розроблено методику керування DC-двигуном через H-міст, що дозволяє точно контролювати швидкість та напрямок обертання. Також було досліджено драйвери двигунів L293D та TB6612FNG, визначено їх технічні характеристики та можливості, що забезпечують надійне та стабільне управління двигунами.

На основі отриманих даних було створено схеми, які демонструють підключення та роботу драйверів, підтверджуючи їх придатність для використання в проекті.

ВИСНОВКИ

У даній роботі був проведений аналіз актуальності рухомих платформ та існуючих варіантів платформ. За основу було взято платформу на механум-колесах.

Було побудовано математичну модель рухомої платформи. За допомогою програмного забезпечення Wolfram Mathematica та на основі математичної моделі була створена інтерактивна програмна модель платформи на механум-колесах. Данна розробка дозволяє наочно демонструвати результати розрахунків. Така візуалізація є вагомим інструментом для навчання та подальшого вдосконалення методів керування рухомими платформами на механум-колесах.

У роботі проведено детальний аналіз різних типів механічних актуаторів, які можуть бути використані для обертання механум-колес, зокрема STEPPER Motor (кроковий двигун), BLDC Motor (безщітковий двигун постійного струму) та DC Motor (двигун постійного струму). Кожен із цих двигунів має свої переваги та недоліки, які були детально проаналізовані для вибору найбільш відповідного варіанту. Вибір було зроблено на користь DC-двигунів завдяки їхній простоті управління та дешевизні.

Було досліджено та реалізовано методи керування DC-моторами з використанням драйверів на основі H-моста: L293D та TB6612FNG, вивчено їх технічні характеристики та можливості, а також проведено тестування в середовищі ISIS Proteus.

Розроблені моделі та алгоритми можуть бути використані для подальших досліджень та розробок в області робототехніки та автоматизації. Досягнуті результати відкривають нові можливості для вдосконалення роботизованих систем, забезпечуючи їх високу маневреність та точність управління в різних умовах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Designing the future [Електронний ресурс]. – режим доступу: URL: <https://www.linkedin.com/pulse/designing-future-anat-treanor> (дата звернення – 09.04.2024).
2. Different Types of Suspensions: Explained in Detail [Електронний ресурс]. – режим доступу: URL: <https://www.spinny.com/blog/index.php/types-of-suspensions/> (дата звернення – 12.05.2024).
3. Penglei Dai, Javad Taghia, Stanley Lam, Jay Katupitiya Integration of sliding mode based steering control and PSO based drive force control for a 4WS4WD vehicle // Auton Robot. – 2018. – V.42. – P.553–568. DOI: 10.1007/s10514-017-9649-6
4. Kotvytskiy A.T., Jasinski Ja. A. Програмна модель рухомої платформи з чотирма провідними поворотними колесами // Scientific Trends And Trends In The Context Of Globalization, Umeå, Sweden (2021) №93, p.537-552 DOI 10.51582/interconf.21-22.12.2021.057
5. Гусеничний трактор John Deere 8370RT [Електронний ресурс]. – режим доступу: URL: <https://www.deere.ua/ru/%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D1%8B/%D1%81%D0%B5%D1%80%D0%B8%D1%8F-8r/8370rt/> (дата звернення – 15.04.2024).
6. Mecanum wheel [Електронний ресурс]. – режим доступу: URL: https://en.wikipedia.org/wiki/Mecanum_wheel (дата звернення – 12.01.2024).
7. Experimental Analysis of Mecanum wheel and Omni wheel [Електронний ресурс]. – режим доступу: URL: https://www.ijiset.com/v1s3/IJISSET_V1_I3_50.pdf (дата звернення – 12.01.2024).
8. Wolfram: computation meets knowledge. Wolfram: Computation Meets Knowledge [Електронний ресурс]. – режим доступу: URL: <https://www.wolfram.com> (дата звернення – 03.03.2024).

9. Wolfram S. Elementary introduction to the wolfram language. Wolfram Media, Incorporated, 2016. 328 p.
10. ВІЗУАЛІЗАЦІЯ АЛГОРИТМУ КЕРУВАННЯ ЧОТИРМА МЕКАНУМ КОЛЕСАМИ [Електронний ресурс]. – режим доступу: URL: <https://www.sworldjournal.com/index.php/swj/issue/view/swj24-01/swj24-01> (дата звернення – 20.05.2024).
11. Introduction to Manipulate [Електронний ресурс]. – режим доступу: URL: <https://reference.wolfram.com/language/tutorial/IntroductionToManipulate.html> (дата звернення – 04.03.2024).
12. Чому колеса при повороті керма повертаються на різний кут? [Електронний ресурс]. – режим доступу: URL: <https://zauto.com.ua/chomu-kolesa-pry-povoroti-kerma-povertaiutsia-na-riznyi-kut/> (дата звернення – 20.04.2024).
13. Clarence W. de Silva. Mechatronics: An Integrated Approach (2005). CRC Press. p. 675.
14. Duane C Hanselman, Brushless Motors: Magnetic Design, Performance, and Control. 2012. 412 p.
15. Herman, Stephen. Industrial Motor Control. 6th ed. Delmar, Cengage Learning, 2010. 544 p.
16. H-Bridges: Theory and Practice [Електронний ресурс]. – режим доступу: URL: <http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/> (дата звернення – 20.01.2024).
17. L293D Datasheet [Електронний ресурс]. – режим доступу: URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/22432/STMICROELECTRONICS/L293D.html> (дата звернення – 22.01.2024).
18. TB6612FNG Datasheet [Електронний ресурс]. – режим доступу: URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/807693/TOSHIBA/TB6612FNG.html> (дата звернення – 23.05.2024).

19. PCB Design Software [Электронный ресурс]. – режим доступа: URL: <https://www.labcenter.com/pcb/#tuning> (дата звернения – 28.01.2024).

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр
галузь знань: 15 – Автоматизація та приладобудування
спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Завідувач кафедри теоретичної
та прикладної системотехніки



д.т.н., проф. Шматков С. І.
«21» грудня 2024 року

З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ

Горенка Данієля Васильовича

(прізвище, ім'я, по батькові студента)

1. Тема роботи «Програмна модель рухомої платформи з механум-колесами»

керівник роботи Котвицький Альберт Тадеушевич, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» травня 2024 року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити

- 1) Аналіз існуючих рухомих платформ.
- 2) Вивчення принципів керування DC-моторами.
- 3) Побудова математичної моделі рухомої платформи.
- 4) Аналіз програмних середовищ для створення моделі.
- 5) Розробка програмної моделі рухомої платформи.
- 6) Вивчення методів тестування.

4. План роботи

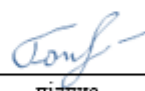
№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз наукової літератури	21.12.2023 - 25.01.2024
2	Вивчення можливостей ISIS Proteus для моделювання керування різними типами двигунів	19.12.2023 - 2.01.2024
3	Аналіз існуючих драйверів для керування DC-моторами	3.01.2024 - 2.02.2024
4	Дослідження методів керування DC-моторами	3.01.2024 - 2.02.2024
5	Розробка математичної моделі руху платформи з механізм-колесами	5.02.2024 - 30.03.2024
6	Аналіз та побудова програмної моделі	3.03.2024 - 29.04.2024
7	Розробка пояснювальної записки.	31.03.2024 - 30.05.2024
8	Оформлення звіту за результатами переддипломної практики.	15.05.2024 – 31.05.2024
9	Представлення кваліфікаційної роботи керівнику та рецензенту.	31.05.2024
10	Оформлення пояснювальної записки та підготовка презентації.	15.04.2024 – 31.05.2024

5. Дата видачі завдання 21.12.2023

Студент

Д. В. Горенко

ініціали, прізвище



підпис

Керівник роботи

А. Т. Котвицький

ініціали, прізвище



підпис

Додаток Б

Затверджую

«_____» _____ 2024 р.

Технічне завдання
на розробку програмного виробу «Програмна модель рухомої
платформи з механум-колесами»

1.	Введення	1.1. Назва: «Програмна модель рухомої платформи з механум-колесами» 1.2. Галузь застосування: робототехніка.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології 2.2. Завдання на кваліфікаційну роботу бакалавра № № 4101-5/909 від «03» травня 2024 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: побудова математичної моделі платформи, що дозволяє визначити швидкості обертання кожного колеса залежно від заданого напрямку руху платформи та модуля швидкості її руху. 3.2. Призначення розробки надає можливість ефективно керувати рухомою платформою і досягати необхідної точності в її переміщенні. 3.3 Вхідні данні: кут, коефіцієнт максимальної швидкості.

		3.4. Вихідні дані розробки: коефіцієнти та візуальне представлення
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: точність до 1 градуса та візуальне відображення напрямлення руху кожного колеса. 4.2. Вимоги до надійності: проміжок від -180 градусів до 180, та коефіцієнт максимально швидкості від 0 до 1. 4.3. Вимоги до умов експлуатації: наявність необхідного встановленого програмного забезпечення. 4.4. Вимоги до складу і параметрів технічних засобів: персональний комп'ютер на базі Windows 11 64 bit. 4.5. Вимоги до інформаційної та програмної сумісності: відповідно до системних вимог розробника програмного забезпечення (при використанні додаткових можливостей необхідно встановити додаткове програмне забезпечення) 4.6. Вимоги до маркування та упаковки: немає 4.7. Вимоги до транспортування і зберігання: на звичайних носіях інформації 4.8. Спеціальні вимоги: ліцензія на програмне забезпечення
5.	Вимоги до програмної документації	Програмною документацією до виробу «Програмна модель рухомої платформи з механум-колесами» вважати: 1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

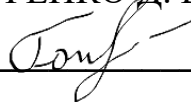
		2) Методику розрахунку коефіцієнтів: За умовами завдання вважається, що кут α і модуль швидкості $\vec{v}$ нам задається. Завдання ставиться таким чином, що потрібно розрахувати відповідні швидкості $\vec{v}_1$ та $\vec{v}_2$. Знаходимо $v_1 = v \cdot \cos\beta$ $v_2 = v \cdot \sin\beta$ де $\beta = \alpha + 45^\circ$. Будемо вважати, що зв'язок між кутовою швидкістю ω_i обертання колеса та швидкістю v_i, яка передається платформі має вигляд $v_i = \zeta \omega_i R,$ де $i = \{1,2\}$, а ζ - безрозмірний коефіцієнт, який знаходиться експериментально і залежить від багатьох чинників. 3) Опис програмної моделі: Інтерактивна програмна модель в середовищі Wolfram Mathematica, в якій за допомогою спеціальних засобів можна змінювати в заданих межах параметри α і швидкість $\vec{v}$ та бачити, як при цьому повинні змінюватися швидкості $\vec{v}_1$ та $\vec{v}_2$
6.	Вимоги до техніко-економічних показників	Програмною документацією до виробу «Програмна модель рухомої платформи з механум-колесами» вважати: 1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

		2) Методику розрахунку коефіцієнтів: як зазначено в пункті 5 3) Опис програмної моделі: як зазначено в пункті 5																																	
7.	Стадії і етапи розробки	<table> <tr> <th>№ з/п</th><th>Назви етапів роботи</th><th>Термін виконання етапів роботи</th></tr> <tr> <td>1</td><td>Аналіз наукової літератури</td><td>21.12.2023 - 25.01.2024</td></tr> <tr> <td>2</td><td>Вивчення можливостей ISIS Proteus для моделювання керування різними типами двигунів</td><td>19.12.2023 - 2.01.2024</td></tr> <tr> <td>3</td><td>Аналіз існуючих драйверів для керування DC-моторами</td><td>3.01.2024 - 2.02.2024</td></tr> <tr> <td>4</td><td>Дослідження методів керування DC-моторами</td><td>3.01.2024 - 2.02.2024</td></tr> <tr> <td>5</td><td>Розробка математичної моделі руху платформи з механум-колесами</td><td>5.02.2024 - 30.03.2024</td></tr> <tr> <td>6</td><td>Аналіз та побудова програмної моделі</td><td>3.03.2024 - 29.04.2024</td></tr> <tr> <td>7</td><td>Розробка пояснювальної записки.</td><td>31.03.2024 - 30.05.2024</td></tr> <tr> <td>8</td><td>Оформлення звіту за результатами переддипломної практики.</td><td>15.05.2024 - 31.05.2024</td></tr> <tr> <td>9</td><td>Представлення кваліфікаційної роботи керівнику та рецензенту.</td><td>31.05.2024</td></tr> <tr> <td>10</td><td>Оформлення пояснювальної записки та підготовка презентації.</td><td>15.04.2024 - 31.05.2024</td></tr> </table>	№ з/п	Назви етапів роботи	Термін виконання етапів роботи	1	Аналіз наукової літератури	21.12.2023 - 25.01.2024	2	Вивчення можливостей ISIS Proteus для моделювання керування різними типами двигунів	19.12.2023 - 2.01.2024	3	Аналіз існуючих драйверів для керування DC-моторами	3.01.2024 - 2.02.2024	4	Дослідження методів керування DC-моторами	3.01.2024 - 2.02.2024	5	Розробка математичної моделі руху платформи з механум-колесами	5.02.2024 - 30.03.2024	6	Аналіз та побудова програмної моделі	3.03.2024 - 29.04.2024	7	Розробка пояснювальної записки.	31.03.2024 - 30.05.2024	8	Оформлення звіту за результатами переддипломної практики.	15.05.2024 - 31.05.2024	9	Представлення кваліфікаційної роботи керівнику та рецензенту.	31.05.2024	10	Оформлення пояснювальної записки та підготовка презентації.	15.04.2024 - 31.05.2024
№ з/п	Назви етапів роботи	Термін виконання етапів роботи																																	
1	Аналіз наукової літератури	21.12.2023 - 25.01.2024																																	
2	Вивчення можливостей ISIS Proteus для моделювання керування різними типами двигунів	19.12.2023 - 2.01.2024																																	
3	Аналіз існуючих драйверів для керування DC-моторами	3.01.2024 - 2.02.2024																																	
4	Дослідження методів керування DC-моторами	3.01.2024 - 2.02.2024																																	
5	Розробка математичної моделі руху платформи з механум-колесами	5.02.2024 - 30.03.2024																																	
6	Аналіз та побудова програмної моделі	3.03.2024 - 29.04.2024																																	
7	Розробка пояснювальної записки.	31.03.2024 - 30.05.2024																																	
8	Оформлення звіту за результатами переддипломної практики.	15.05.2024 - 31.05.2024																																	
9	Представлення кваліфікаційної роботи керівнику та рецензенту.	31.05.2024																																	
10	Оформлення пояснювальної записки та підготовка презентації.	15.04.2024 - 31.05.2024																																	
8.	Порядок контролю і приймання програмного продукту (моделі)	1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати в електронному вигляді в 1 примірнику.																																	

Виконавець

студент групи КУ- 41

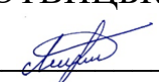
ГОРЕНКО Д. В.



Замовник

к.ф.-м.н., доцент

КОТВИЦЬКИЙ А. Т.



Програма і методика випробувань програмного виробу

«ПРОГРАМНА МОДЕЛЬ РУХОМОЇ ПЛАТФОРМИ З МЕКАНУМ-КОЛЕСАМИ»

1 Об'єкт випробувань

1. Назва програмного виробу : «Програмна модель рухомої платформи з механум-колесами»
2. Галузь застосування : Автоматизація та приладобудування
3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної моделі заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

1. Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

1. працювати на основних операційних системах: Windows, Linux, MacOS;
2. вимоги до надійності;
3. передбачити захист від некоректних дій користувача;
4. бути легко розширюваною;
5. елементи моделі повинні бути ізольовані одне від одного для зменшення їх впливу на роботи програми під час редагування програмного коду;
6. вимоги до складу і параметрів технічних засобів;
7. вимоги до маркування та упаковки (не висуваються);
8. вимоги до транспортування і зберігання (не висуваються).

Спеціальні вимоги (не пред'являються).

5. Вимоги до програмної документації

Програмною документацією до виробу «Програмна модель рухомої платформи з механум-колесами» вважати:

1. Програмною документацією щодо розроблюваного програмного продукту вважати:
2. справжнє технічне завдання на розробку програми (представити як Додаток Б до пояснювальної записки до кваліфікаційної роботи);
3. Програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи);

4. рекомендацій щодо застосування створеної програмної стандартизації у проєктах (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).
5. Текст програми(представити як Додаток Г до пояснювальної записки до кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Для проведення випробувань необхідний проєкт програмної моделі рухомої платформи з механум-колесами

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.
2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
3. Перевірку комплектності складу технічних і програмних засобів.
4. Методику проведення перевірок на 1 етапі випробувань.
5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1. перевірку відповідності технічних характеристик програми вимогам технічного завдання;
2. перевірку ступеня виконання функціональних вимог до програми;

3. методику проведення перевірок, що входять до переліку по 2 етапу випробувань.

1. Програма працює відповідно до умов експлуатації операційних систем MS Windows, Linux та MacOS.
2. Для роботи необхідне програмне забезпечення Wolfram Mathematica
3. Порядок проведення випробувань:
 - 3.1. Запуск моделі здійснюється за допомогою комбінації «Shift + Enter»;
 - 3.2. Після запуску, модель буде готова отримувати вхідні данні для розрахунків та візуалізації;
 - 3.3. Після введення даних, на екрані з'явиться результати розрахунків та візуальне представлення.

Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

Тест 1

1. Перевірка виконання програми;
2. введення даних, для перевірки правильності розрахунків;
3. отримання результатів розрахунків.

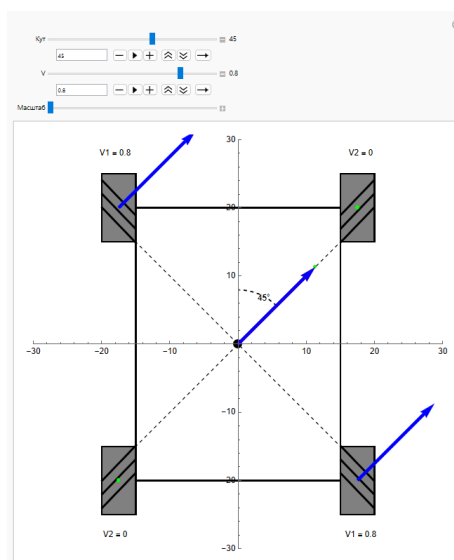


Рис. В.1 Тест 1

Тест вважається пройденим, якщо при куті 45 градусів отримуємо коефіцієнти $V1 = V$ вхідному та $V2 = 0$.

Тест 2

1. Перевірка виконання програми
2. введення даних для перевірки діапазонів вхідних даних;
3. отримання результатів.

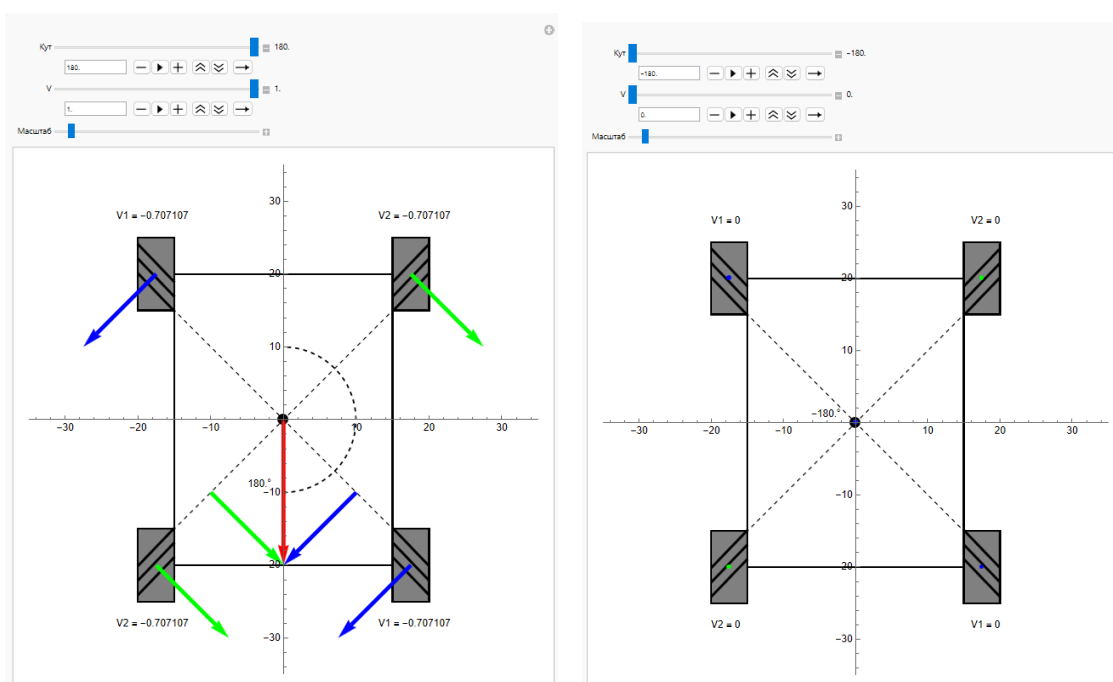


Рис. В.2 Тест 2

Тест вважається пройденим, якщо модель реагує на зміни мінімальних і максимальних вхідних даних, при цьому коефіцієнти швидкості залишаються в межах допустимих границь.

Тест 3

1. Перевірка виконання програми
2. введення даних для перевірки візуального відображення;
3. отримання вірного відображення .

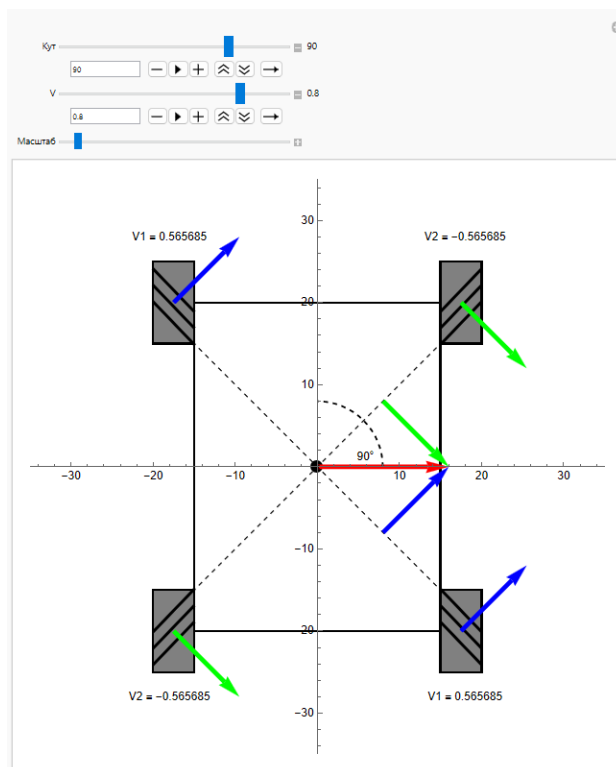


Рис. В.3 Тест 3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КУ-41, Горенко Д. В.

Вихідний код програми на Wolfram Mathematica

Manipulate
[маніпулювати]

```
Module[{GreenArrowStart, GreenArrowEnd, GreenArrowArrow, GreenDashedLineStart, GreenDashedLineEnd,
[програмний модуль]

  GreenPerpendicularPoint, BlueArrowStart, BlueArrowEnd, BlueArrowArrow, BlueDashedLineStart,
  BlueDashedLineEnd, BluePerpendicularPoint},

  GreenArrowEnd = V * 20 * {Cos[Pi / 2 - angle Degree], Sin[Pi / 2 - angle Degree]};
                                [ко... [число пі] [градус] [си... [число пі] [градус]
  (*Кінцева точка зеленої стрілки*)
  BlueArrowEnd = V * 20 * {Cos[Pi / 2 - angle Degree], Sin[Pi / 2 - angle Degree]};
                                [ко... [число пі] [градус] [си... [число пі] [градус]
  (*Кінцева точка синьої стрілки*)

  (*Розраховуємо координати початку та кінця пунктирної лінії*)
  GreenDashedLineStart = {-20, -20};
  GreenDashedLineEnd = {20, 20};

  BlueDashedLineStart = {-20, 20};
  BlueDashedLineEnd = {20, -20};

  (*Розраховуємо точку перетину прямої, що проходить через кінець червоної стрілки та прямої,
  що проходить через початок пунктирної лінії*)
  GreenPerpendicularPoint = RegionIntersection[InfiniteLine[{GreenArrowEnd, GreenArrowEnd + {1, -1}}],
                                [перетин геометричних ... [пряма
    InfiniteLine[{GreenDashedLineStart, GreenDashedLineEnd}]];
    [пряма
  BluePerpendicularPoint = RegionIntersection[InfiniteLine[{BlueArrowEnd, BlueArrowEnd + {1, 1}}],
                                [перетин геометричних ... [пряма
    InfiniteLine[{BlueDashedLineStart, BlueDashedLineEnd}]];
    [пряма
  (*Визначаємо початкову точку синьої стрілки як перетин вищезгаданих прямих*)
  GreenArrowStart = GreenPerpendicularPoint[[1]];
  BlueArrowStart = BluePerpendicularPoint[[1]];

  GreenArrowArrow = {Green, Thickness[0.008], Arrow[{GreenArrowStart, GreenArrowEnd}]}];
                                [зеле... [товщина] [стрілка]
  (*Створення зеленої стрілки*)
  BlueArrowArrow = {Blue, Thickness[0.008], Arrow[{BlueArrowStart, BlueArrowEnd}]}];
                                [синій [товщина] [стрілка]
  (*Створення синьої стрілки*)
```

```

Module| {FrontLeftWheelCenter, BlueArrowEnd1, BlueArrowArrow1, BlueArrowArrow2, RearRightWheelCenter,
|програмний модуль

    GreenArrowArrow1, GreenArrowArrow2, FrontRightWheelCenter, RearLeftWheelCenter, GreenArrowEnd1},
    (*Кінцеві точки синіх стрілок*)
    FrontLeftWheelCenter = {-17.5, 20};
    BlueArrowEnd1 = V * 20 * {Cos[Pi / 2 - angle Degree], Sin[Pi / 2 - angle Degree]};
                                |ко... |число пі |градус |си... |число пі |градус
    RearRightWheelCenter = {17.5, -20};
    BlueArrowEnd1 = V * 20 * {Cos[Pi / 2 - angle Degree], Sin[Pi / 2 - angle Degree]};
                                |ко... |число пі |градус |си... |число пі |градус
    (*Кінцеві точки зелених стрілок*)
    FrontRightWheelCenter = {17.5, 20};
    GreenArrowEnd1 = V * 20 * {Cos[Pi / 2 - angle Degree], Sin[Pi / 2 - angle Degree]};
                                |ко... |число пі |градус |си... |число пі |градус
    RearLeftWheelCenter = {-17.5, -20};
    GreenArrowEnd1 = V * 20 * {Cos[Pi / 2 - angle Degree], Sin[Pi / 2 - angle Degree]};
                                |ко... |число пі |градус |си... |число пі |градус

    (*Створення синіх стрілок*)
    BlueArrowArrow1 = {Blue, Thickness[0.008],
                        |синій |товщина
                        Arrow[ {FrontLeftWheelCenter, FrontLeftWheelCenter + BlueArrowEnd1 - BlueArrowStart} ]};
                        |стрілка
    BlueArrowArrow2 = {Blue, Thickness[0.008],
                        |синій |товщина
                        Arrow[ {RearRightWheelCenter, RearRightWheelCenter + BlueArrowEnd1 - BlueArrowStart} ]};
                        |стрілка

    (*Створення зелених стрілок*)
    GreenArrowArrow1 = {Green, Thickness[0.008],
                        |зеле... |товщина
                        Arrow[ {FrontRightWheelCenter, FrontRightWheelCenter + GreenArrowEnd1 - GreenArrowStart} ]};
                        |стрілка
    GreenArrowArrow2 = {Green, Thickness[0.008],
                        |зеле... |товщина
                        Arrow[ {RearLeftWheelCenter, RearLeftWheelCenter + GreenArrowEnd1 - GreenArrowStart} ]};
                        |стрілка

```

```

Graphics[ {
  (*Малюємо головний прямокутник*)
  EdgeForm[ {Thick, Black}],
  |стиль ребер |жирни... |чорний
  FaceForm[White],
  |стиль грані |білий
  Rectangle[{-15, -20}, {15, 20}],
  |прямокутник

  (*Малюємо колеса*)
  EdgeForm[ {Thick, Black}], FaceForm[Gray],
  |стиль ребер |жирни... |чорний |стиль грані |сірий
  Rectangle[{-20, 15}, {-15, 25}], (*Левое переднее колесо*)
  |прямокутник
  Rectangle[{15, 15}, {20, 25}], (*Правое переднее колесо*)
  |прямокутник
  Rectangle[{-15, -15}, {-20, -25}], (*Левое заднее колесо*)
  |прямокутник
  Rectangle[{15, -25}, {20, -15}], (*Правое заднее колесо*)
  |прямокутник

  Black, PointSize[0.02], Point[{0, 0}],
  |чорний |розмір точки |точка
  {Red, Thickness[0.008], Rotate[Arrow[{0, 0}, {0, V*20}], -angle Degree, {0, 0}],
  |че... |товщина |повер... |стрілка |градус
  Black, Text[Row[{angle, "°"}], V*10 {Cos[Pi/2 - angle Degree], Sin[Pi/2 - angle Degree]}, {2, -1.5}],
  |чорний |текст |ряд |ко... |число пі |градус |си... |число пі |градус
  {Dashed, Thickness[0.003], Circle[{0, 0}, V*10, {Pi/2, Pi/2 - angle Degree}]},
  |штрихо... |товщина |коло |число... |число пі |градус

  (*Косі лінії для верхніх колес*)
  {Black, Thickness[0.005], Table[Line[{{-15.1, 20 - 2 i + 1}, {-19.9, 25 - 2 i + 1}}], {i, 1, 3}]},
  |чорний |товщина |табл... | (ламана) лінія
  {Black, Thickness[0.005], Table[Line[{{15.1, 20 - 2 i + 1}, {19.9, 25 - 2 i + 1}}], {i, 1, 3}]},
  |чорний |товщина |табл... | (ламана) лінія

  (*Косі лінії для нижніх колес*)
  {Black, Thickness[0.005], Table[Line[{{-15.1, -20 + 2 i - 1}, {-19.9, -25 + 2 i - 1}}], {i, 1, 3}]},
  |чорний |товщина |табл... | (ламана) лінія
  {Black, Thickness[0.005], Table[Line[{{15.1, -20 + 2 i - 1}, {19.9, -25 + 2 i - 1}}], {i, 1, 3}]},
  |чорний |товщина |табл... | (ламана) лінія

  (*Додані пунктирні лінії під кутом 45 градусів до осей координат*)
  {Dashed, Line[{{-20, -20}, {20, 20}}]}, (*Пунктирна лінія під кутом 45 градусів до осі X*)
  |штрих... | (ламана) лінія
  {Dashed, Line[{{-20, 20}, {20, -20}}]}, (*Пунктирна лінія під кутом 45 градусів до осі Y*)
  |штрих... | (ламана) лінія

```

```

(*Розрахунки V1 и V2*)
Betta = -  $\frac{\text{angle} * \text{Pi}}{180}$  + Pi / 4;
           |число пі

Black, Text[
|чорний |текст
  If[Abs[V * Cos[Betta]] < 10^-6,
  |... |абсол... |косинус
    Row[{"V1 = ", 0}],
    |ряд
    Row[{"V1 = ", V * Cos[Betta]}], {-18, 28}],
    |ряд |косинус
Black, Text[
|чорний |текст
  If[Abs[V * Cos[Betta]] < 10^-6,
  |... |абсол... |косинус
    Row[{"V1 = ", 0}],
    |ряд
    Row[{"V1 = ", V * Cos[Betta]}], {18, -28}],
    |ряд |косинус
Black, Text[
|чорний |текст
  If[Abs[V * Sin[Betta]] < 10^-6,
  |... |абсол... |синус
    Row[{"V2 = ", 0}],
    |ряд
    Row[{"V2 = ", V * Sin[Betta]}], {18, 28}],
    |ряд |синус
Black, Text[
|чорний |текст
  If[Abs[V * Sin[Betta]] < 10^-6,
  |... |абсол... |синус
    Row[{"V2 = ", 0}],
    |ряд
    Row[{"V2 = ", V * Sin[Betta]}], {-18, -28}],
    |ряд |синус

(*Створюємо головні стрілки*)
GreenArrowArrow, (*додана синя стрілка*)
BlueArrowArrow, (*додана синя стрілка*)

(*Створюємо стрілки на колесах*)
BlueArrowArrow1,
BlueArrowArrow2,

GreenArrowArrow1,
GreenArrowArrow2,
},

```

```

Axes → True,
  |осі      |істина
PlotRange → plotRange,
  |діапазон значень на графіку
ImageSize → 500]
  |розмір зображення
]],
{{angle, 0, "Кут"}, -180, 180, Appearance → "Labeled"}, {{V, 0.5, "V"}, 0, 1, Appearance → "Labeled"},
  |зовнішній вигляд      |зовнішній вигляд
{{plotRange, 30, "Масштаб"}, 30, 100}
]

```