

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та охоронних систем

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА СТАБІЛІЗАЦІЇ ТА НАВІГАЦІЇ
БАЛАНСУЮЧОГО РОБОТА НА БАЗІ МІКРОКОНТРОЛЕРА**

**INTELLIGENT STABILIZATION AND NAVIGATION SYSTEM
FOR A BALANCING ROBOT BASED ON A
MICROCONTROLLER**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42

Бурковський Ярослав Анатолійович

(підпис)

Керівник:

к.т.н., доцент

Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу
допущено до захисту

« » червня 2026 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т. Терлецький

« 23 » 12 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Бурковському Ярославу Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Інтелектуальна система стабілізації та навігації балансуєчого робота на базі мікроконтролера

Керівник роботи к.т.н., доцент Гринюк Сергій Васильович

затверджені наказом закладу вищої освіти від «20» грудня 2025 року № 536/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 28.05.2026 р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Аналіз теоретичних підходів до побудови балансуєчих робіт

Проектування та апаратно-програмна реалізація системи керування на базі платформи Arduino

Практична реалізація та тестування прототипу балансуєчого робота

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Загальна структура алгоритму стабілізації положення балансуєчого робота

Етап ініціалізації мікроконтролера та інерціального сенсора MPU6050

Зчитування даних акселерометра та гіроскопа з модуля MPU6050

Фільтрація даних інерціального сенсора для визначення кута нахилу

Блок-схема алгоритму стабілізації положення балансуєчого робота

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз теоретичних підходів до побудови балансуючих робіт</i>	<i>Гринюк С.В., доцент</i>		
<i>Проектування та апаратно-програмна реалізація системи керування на базі платформи Arduino</i>	<i>Гринюк С.В., доцент</i>		
<i>Практична реалізація та тестування прототипу балансуючого робота</i>	<i>Гринюк С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст. викладач</i>		

7. Дата видачі завдання 23.12.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень</i>	до 10.02.2026 р.	
2.	<i>Аналіз теоретичних підходів до побудови балансуючих робіт</i>	до 02.03.2026 р.	
3.	<i>Проектування та апаратно-програмна реалізація системи керування на базі платформи Arduino</i>	до 02.04.2026 р.	
4.	<i>Практична реалізація та тестування прототипу балансуючого робота</i>	до 10.04.2026 р.	
5.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	до 10.05.2026 р.	
6.	<i>Нормоконтроль</i>	до 23.05.2026 р.	
7.	<i>Інструментальна перевірка на академічний плагіат</i>	до 25.05.2026 р.	
8.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедру</i>	до 28.05.2026 р.	

Здобувач вищої освіти

(підпис)

Ярослав БУРКОВСЬКИЙ

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Сергій ГРИНЮК

(прізвище, ініціали)

АНОТАЦІЯ

Бурковський Я. А. Інтелектуальна система стабілізації та навігації балансуєчого робота на базі мікроконтролера. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, переліку використаних джерел, додатків.

У першому розділі проаналізовано теоретичні підходи до побудови двоколісних балансуєчих робототехнічних систем, розглянуто їхні конструктивні особливості, перспективи розвитку, математичну модель робота як зворотного маятника.

У другому розділі виконано проектування апаратно-програмної системи керування на базі платформи Arduino. Обґрунтовано вибір мікроконтролера, сенсорної підсистеми, виконавчих механізмів і навігаційного модуля. Особливу увагу приділено використанню інерціального сенсора для визначення кута нахилу робота, мотор-редукторів для реалізації руху та ультразвукового датчика для виявлення перешкод.

У третьому розділі описано практичну реалізацію та тестування прототипу балансуєчого робота. Виконано складання апаратної частини, розроблено алгоритм стабілізації на основі даних інерціального сенсора, реалізовано програмне керування мікроконтролером і навігаційний модуль для руху робота в середовищі. Проведене тестування підтвердило працездатність розробленої системи, її здатність реагувати на зміну кута нахилу, формувати керуючі сигнали для двигунів і виявляти перешкоди під час руху.

Ключові слова: балансуєчий робот, мікроконтролер, Arduino, інерціальний сенсор, MPU6050, ПІД-регулятор, стабілізація, автономна навігація, ультразвуковий датчик, HC-SR04, робототехнічна система.

ANNOTATION

Burkovsky Ya. Intelligent stabilization and navigation system for a balancing robot based on a microcontroller. Manuscript.

Bachelor's qualification work of the EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2026.

The qualification work consists of an introduction, three sections, conclusions, a list of sources used, and appendices.

The first section analyzes theoretical approaches to the construction of two-wheeled balancing robotic systems, considers their design features, development prospects, and a mathematical model of the robot as an inverted pendulum.

The second section designs a hardware-software control system based on the Arduino platform. The choice of a microcontroller, sensor subsystem, actuators, and navigation module is justified. Particular attention is paid to the use of an inertial sensor to determine the angle of inclination of the robot, motor-gearboxes for the implementation of movement and an ultrasonic sensor for detecting obstacles.

The third section describes the practical implementation and testing of the prototype of the balancing robot. The hardware part was assembled, a stabilization algorithm was developed based on inertial sensor data, software control of the microcontroller and a navigation module for the movement of the robot in the environment were implemented. The testing confirmed the operability of the developed system, its ability to respond to changes in the angle of inclination, generate control signals for the motors and detect obstacles during movement.

Keywords: balancing robot, microcontroller, Arduino, inertial sensor, MPU6050, PID controller, stabilization, autonomous navigation, ultrasonic sensor, HC-SR04, motor-gearbox, robotic system.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ТЕОРЕТИЧНИХ ПІДХОДІВ ДО ПОБУДОВИ БАЛАНСУЮЧИХ РОБОТІВ.....	10
1.1 Огляд та перспектив розвитку двоколісних робототехнічних систем	10
1.2 Математичний опис динаміки робота як зворотного маятника	13
1.3 Порівняльний аналіз методів стабілізації нестійких об’єктів	16
1.4 Огляд методів автономної навігації та детектування перешкод у локальному просторі.....	19
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА АПАРАТНО-ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ ПЛАТФОРМИ ARDUINO	22
2.1 Розробка структурної схеми апаратної архітектури та обґрунтування вибору мікроконтролера.....	22
2.2 Обґрунтування вибору сенсорної підсистеми та виконавчих механізмів..	25
2.3 Алгоритмічна реалізація каскадної системи ПІД-керування та розробка програмного забезпечення.....	27
2.4 Розробка підсистеми автономної навігації та планування шляху	30
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОТОТИПУ БАЛАНСУЮЧОГО РОБОТА.....	33
3.1 Складання апаратної частини прототипу.....	33
3.2 Розробка алгоритму стабілізації положення робота на основі даних інерціального сенсора	39
3.3 Програмна реалізація системи керування мікроконтролером	46
3.4 Реалізація навігаційного модуля та алгоритму руху робота в середовищі	51
ВИСНОВКИ	57
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	59

ВСТУП

У сучасних умовах розвитку комп'ютерної інженерії, робототехніки та вбудованих систем особливої актуальності набуває створення автономних мобільних роботів, здатних самостійно орієнтуватися в просторі, підтримувати стійке положення та виконувати задані дії без постійного втручання користувача. Одним із перспективних напрямів у цій сфері є розробка балансуєчих роботів, які функціонують за принципом динамічної рівноваги. Такі системи поєднують у собі елементи мехатроніки, автоматичного керування, сенсорної обробки даних, програмування мікроконтролерів та алгоритмів навігації.

Балансуєчий робот є складною технічною системою, оскільки його робота ґрунтується на постійному контролі кута нахилу, швидкому аналізі показників сенсорів і формуванні керувальних сигналів для електродвигунів. На відміну від традиційних мобільних платформ, такий робот не має статичної стійкості, тому повинен безперервно коригувати своє положення у просторі. Це зумовлює необхідність використання інерціальних вимірювальних модулів, мікроконтролерів, драйверів двигунів та програмних алгоритмів стабілізації, які забезпечують утримання рівноваги в реальному часі.

Актуальність роботи полягає в тому, що балансуєчі роботи можуть використовуватися як навчальні, дослідницькі та експериментальні платформи для вивчення принципів автоматичного керування, обробки сенсорних даних, реалізації PID-регуляторів, автономної навігації та побудови інтелектуальних робототехнічних систем. Крім того, такі розробки мають практичне значення для створення компактних мобільних платформ, сервісних роботів, роботизованих транспортних засобів та систем, здатних функціонувати в умовах обмеженого простору.

Метою роботи є розробка інтелектуальної системи стабілізації та навігації балансуєчого робота на базі мікроконтролера, що забезпечує підтримання рівноваги, обробку даних від сенсорів і керування рухом мобільної платформи.

Об'єктом дослідження є процес стабілізації та навігації балансуєчого мобільного робота на базі мікроконтролерної системи керування.

Предметом дослідження є апаратні засоби, програмні алгоритми та методи керування, що забезпечують стабілізацію положення і навігацію балансуєчого робота.

Для досягнення поставленої мети в роботі необхідно вирішити такі завдання:

- проаналізувати теоретичні основи побудови балансуєчих роботів та принципи їхньої динамічної рівноваги;
- дослідити методи стабілізації положення мобільних роботизованих систем;
- охарактеризувати сенсорні засоби, які використовуються для визначення кута нахилу та просторової орієнтації робота;
- обґрунтувати вибір мікроконтролера, інерціального модуля, двигунів, драйверів керування та джерела живлення;
- розробити структурну схему інтелектуальної системи стабілізації та навігації балансуєчого робота;
- описати алгоритм обробки сенсорних даних і формування керувальних сигналів для двигунів;
- написати програмне забезпечення для стабілізації та навігації робота на базі мікроконтролера;
- провести тестування працездатності створеної системи та оцінити її ефективність.

Методи дослідження включають аналіз науково-технічних джерел, порівняльний аналіз апаратних компонентів, методи структурного та функціонального моделювання, алгоритмічне проєктування, експериментальне тестування, а також методи обробки даних, отриманих від інерціальних сенсорів. Для програмної реалізації системи можуть бути використані середовище Arduino IDE або інше сумісне середовище розробки, мова програмування C/C++ та бібліотеки для роботи з інерціальними модулями і драйверами двигунів.

Практичне значення роботи полягає у створенні прототипу балансуючого робота, який може бути використаний як навчально-дослідна платформа для вивчення мікроконтролерних систем, алгоритмів стабілізації, сенсорної обробки даних і навігації мобільних роботів.

РОЗДІЛ 1

АНАЛІЗ ТЕОРЕТИЧНИХ ПІДХОДІВ ДО ПОБУДОВИ БАЛАНСУЮЧИХ РОБОТІВ

1.1 Огляд стану та перспектив розвитку двоколісних робототехнічних систем

Сучасний етап розвитку мобільної робототехніки характеризується інтенсивним переходом від статично стійких багатоколісних платформ до систем із динамічною стабілізацією, серед яких ключове місце посідають двоколісні балансуючі роботи (Two-Wheeled Self-Balancing Robots – TWSBR) [1]. Концептуально такі механізми базуються на принципі зворотного маятника, де підтримка вертикального положення корпусу забезпечується безперервним коригуванням крутного моменту двигунів коліс на основі даних просторової орієнтації [2]. Поштовхом до комерційного та наукового інтересу в цій галузі стала поява проектів індивідуального транспорту на початку 2000-х років, проте за період 2022-2026 років технологія зазнала суттєвих трансформацій, перетворившись із простих транспортних засобів на складні кіберфізичні системи з елементами штучного інтелекту.

На рисунку 1.1 представлена класична кінематична схема, що описує механічну структуру двоколісного балансуючого робота, представленого як фізичний зворотний маятник, встановлений на рухомій осі коліс. Схема виконана у двовимірній площині xz , де горизонтальна вісь x позначає опорну поверхню. У точці контакту O , яка є віссю коліс, знаходиться колесо масою m_w та радіусом r . До цієї ж точки O шарнірно прикріплений корпус робота, змодельований у вигляді тонкого стрижня довжиною l . Маса корпусу m_p (маса маятника) зосереджена в точці, розташованій на відстані l від осі O .

Поточний стан системи визначається двома узагальненими координатами: лінійним переміщенням x центру колеса та кутом θ відхилення корпусу робота від вертикальної осі z . Окремо показано дію прискорення вільного падіння (вектор g), спрямованого вниз, та крутний момент T , що прикладається

двигунами до коліс для стабілізації системи та забезпечення руху. Ця схема є фундаментальною основою для виведення рівнянь динаміки системи за допомогою формалізму Лагранжа.

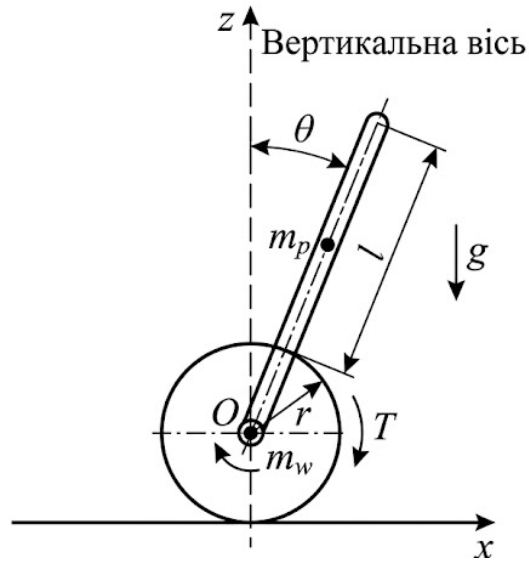


Рисунок 1.1 – Кінематична схема двоколісного балансуєчого робота як моделі зворотного маятника [2]

Аналіз існуючих комерційних рішень демонструє широку диверсифікацію застосування: від засобів мікромобільності до спеціалізованих роботів телеприсутності. Останні використовують двоколісну схему для забезпечення мінімальної площі опори (footprint), що дозволяє пристрою ефективно маневрувати у вузьких коридорах та офісних приміщеннях, імітуючи присутність людини на рівні очей. У наукових дослідженнях останніх років спостерігається чітка тенденція до поєднання колісного та ногого способів пересування, що призвело до появи гібридних «колісно-ногих» (wheel-legged) роботів, яскравими представниками яких є дослідницькі платформи серії Ascento та їхні аналоги [3]. Такі системи поєднують високу швидкість та енергоефективність колісного ходу на рівних поверхнях із здатністю долати складні перешкоди, такі як сходи або бордюри, шляхом динамічної зміни геометрії маніпуляторів, що виконують роль ніг.

Керування подібними нестійкими об'єктами в період 2024-2026 років все частіше базується на сучасних методах теорії автоматичного керування, зокрема на навчанні з підкріпленням (Reinforcement Learning) та прогнозованому керуванні моделями (Model Predictive Control). Традиційні лінійні регулятори, такі як пропорційно-інтегрально-диференціальний (ПІД) та лінійно-квадратичний (LQR), тепер ефективно доповнюються алгоритмами на базі штучних нейронних мереж, які дозволяють системі адаптуватися до нерівностей покриття та зовнішніх збурень без необхідності точного математичного опису навколишнього середовища [4].

На рисунку 1.2 подано структурну схему контуру стабілізації балансуєчого робота. Вона відображає послідовність проходження керувального сигналу від заданого кута нахилу до виконавчих механізмів, а також формування зворотного зв'язку на основі даних сенсорного блока. Така схема дає змогу простежити взаємодію ПІД-регулятора, драйверів двигунів, механічної системи робота та сенсорів, які забезпечують оцінювання поточного кута нахилу й положення платформи.

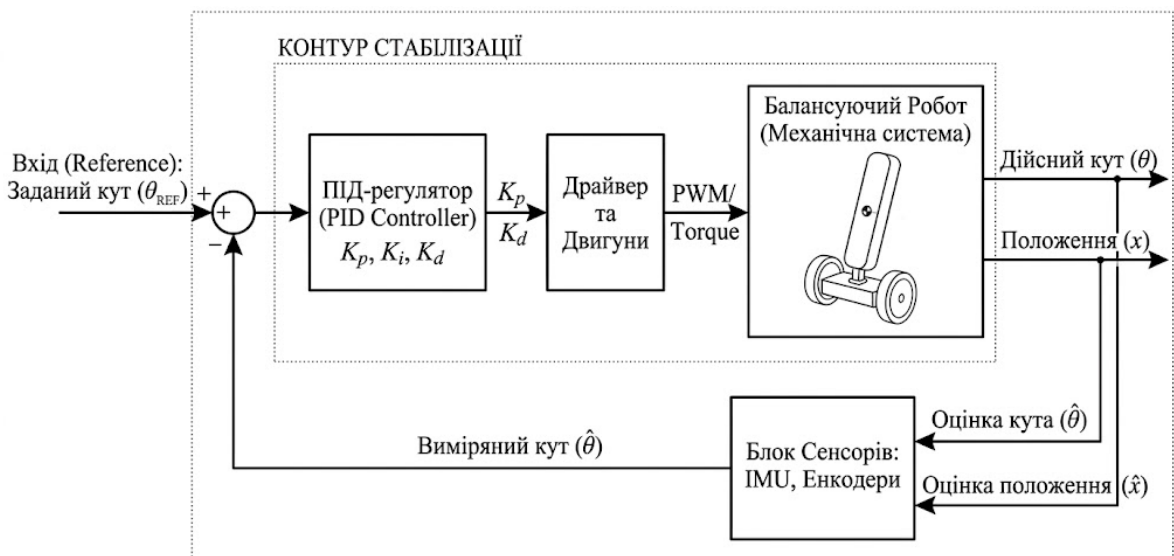


Рисунок 1.2 – Блок-схема типового контуру керування балансуєчим роботом [4]

Використання сучасних обчислювальних платформ з підтримкою алгоритмів векторного керування (Field-Oriented Control) та високоточних

мікроелектромеханічних (MEMS) сенсорів дозволяє реалізовувати ці складні математичні стратегії безпосередньо на борту робота. Глибока оптимізація алгоритмів комплексування даних (sensor fusion), зокрема адаптивних фільтрів Калмана, забезпечує високу плавність ходу та енергоефективність системи навіть при використанні низькобюджетних мікроконтролерних платформ [5].

Перспективи розвитку двоколісних платформ безпосередньо пов'язані з їхньою інтеграцією в інфраструктуру «розумних міст» як автономних засобів доставки «останньої милі». Завдяки впровадженню технологій одночасної локалізації та побудови карти (SLAM), а також сучасних систем комп'ютерного зору, новітні прототипи стають здатними до повноцінної автономної роботи в динамічному просторі, безпечно взаємодіючи з пішоходами та іншими транспортними засобами [1]. Подальше вдосконалення твердотільних акумуляторних технологій та використання композитних матеріалів дозволить значно розширити радіус дії та вантажопідйомність таких систем, перетворюючи їх на ключові елементи логістики та сервісної робототехніки майбутнього.

1.2 Математичний опис динаміки робота як зворотного маятника

Фундаментальною основою для синтезу системи автоматичного керування двоколісним балансуєчим роботом є розробка його адекватної математичної моделі, яка описує динаміку поведінки об'єкта в просторі. Відповідно до законів класичної механіки, такий робот розглядається як нелінійна багатозв'язна електромеханічна система, що є варіацією класичної задачі зворотного маятника на рухомому візку [6]. Для отримання диференціальних рівнянь руху найчастіше застосовується енергетичний метод, заснований на формалізмі Лагранжа другого роду, який дозволяє уникнути складного аналізу внутрішніх сил реакції в шарнірах системи та безпосередньо зв'язати енергетичні стани з керуючими впливами [7, 8].

Для побудови моделі вводиться нерухома декартова система координат і визначається вектор узагальнених координат за формулою (1.2):

$$q = [x, \theta]^T, \quad (1.2)$$

де x – лінійне переміщення осі коліс уздовж поверхні;

θ – кут відхилення поздовжньої осі корпусу робота від абсолютної вертикалі [2].

Кінематика системи описується координатами центру мас корпусу робота (маятника) відносно осі обертання коліс. Координати центру мас маятника x_p та y_p визначаються просторовими геометричними співвідношеннями (1.2) і (1.3):

$$x_p = x + l \sin \theta \quad (1.2)$$

$$y_p = l \cos \theta, \quad (1.3)$$

де l – відстань від осі коліс до центру мас корпусу [8].

Продиференціювавши ці рівняння за часом, отримуємо проекції вектора лінійної швидкості центру мас корпусу (1.4) та (1.5):

$$\dot{x}_p = \dot{x} + l\dot{\theta} \cos \theta \quad (1.4)$$

$$\dot{y}_p = -l\dot{\theta} \sin \theta \quad (1.5)$$

Повна кінетична енергія системи E_k складається з кінетичної енергії поступального та обертального рухів коліс, а також плоскопаралельного руху масивного корпусу [9]. Математично вона виражається формулою (1.6):

$$E_k = \frac{1}{2} M_\omega \dot{x}^2 + \frac{1}{2} I_\omega \left(\frac{\dot{x}}{r} \right)^2 + \frac{1}{2} m_p (\dot{x}_p^2 + \dot{y}_p^2) + \frac{1}{2} I_p \dot{\theta}^2, \quad (1.6)$$

де M_ω – сумарна маса коліс та роторів двигунів;

I_ω – еквівалентний момент інерції коліс;

r — радіус колеса;

m_p – маса корпусу робота;

I_p – момент інерції корпусу відносно його центру мас.

Після підстановки квадратів швидкостей та застосування основних тригонометричних тотожностей кінетична енергія набуває остаточного вигляду. Потенціальна енергія системи E_p визначається виключно зміною висоти центру мас корпусу робота в гравітаційному полі Землі, оскільки вертикальні переміщення коліс у моделі відсутні, і визначаються за формулою (1.7):

$$E_p = m_p g l, \quad (1.7)$$

де g – прискорення вільного падіння [10].

Функція Лагранжа L , яка визначається як різниця між повною кінетичною та повною потенціальною енергіями, підставляється у класичне рівняння Ейлера-Лагранжа (1.8):

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i, \quad (1.8)$$

де Q_i – узагальнені сили [9].

Виконавши операції диференціювання за узагальненими координатами x та θ , отримуємо систему нелінійних диференціальних рівнянь другого порядку (1.9):

$$\begin{cases} \left(M_\omega + \frac{I_\omega}{r^2} + m_p \right) \ddot{x} + m_p l \ddot{\theta} \cos \theta - m_p l \dot{\theta}^2 \sin \theta = F \\ (I_p + m_p l^2) \ddot{\theta} + m_p l \ddot{x} \cos \theta - m_p g l \sin \theta = -T_m \end{cases}, \quad (1.9)$$

де F – сумарна рушійна сила, прикладена до коліс у горизонтальному напрямку;

T_m – керуючий крутний момент електродвигунів.

Отримана система є суттєво нелінійною через наявність тригонометричних функцій та відцентрової складової. Оскільки основною метою є утримання

робота поблизу вертикального положення рівноваги, для синтезу лінійних систем керування модель піддається процедурі лінеаризації за допомогою розкладання в ряд Тейлора в околі робочої точки $\theta = 0, \dot{\theta} = 0$. При цьому приймаються припущення для малих кутів: $\sin \theta \approx \theta, \cos \theta \approx 1$, а нелінійним добутком швидкостей нехтують через його мізерний вплив у заданому околі [7].

Лінійна математична модель зручно подається у формі простору станів, що має класичний матричний вигляд (1.10):

$$\dot{X} = AX + BU, \quad (1.10)$$

де $X = [x, \dot{x}, \theta, \dot{\theta}]^T$ – вектором стану [8].

Матриця системи A відображає внутрішню динаміку об'єкта та підтверджує наявність додатного кореня характеристичного рівняння, що математично доводить структурну нестійкість розімкненої системи. Таке матричне подання є фундаментальним базисом для подальшого аналізу критеріїв керованості та спостережуваності, розрахунку матриць коефіцієнтів LQR - регулятора, а також для налаштування контурів класичної ПІД-стабілізації [12].

1.3 Порівняльний аналіз методів стабілізації нестійких об'єктів

Управління принципово нестійкими нелінійними об'єктами, до класу яких належить двоколісний балансуєчий робот, вимагає застосування високоефективних алгоритмів автоматичного керування. На сучасному етапі розвитку мобільної робототехніки для вирішення задачі стабілізації моделі зворотного маятника найчастіше застосовуються класичні пропорційно-інтегрально-диференціальні (ПІД) регулятори, оптимальне лінійно-квадратичне керування (LQR) та інтелектуальні методи на основі нечіткої логіки (Fuzzy Logic). Кожен із цих математичних апаратів має свої специфічні переваги та жорсткі обмеження щодо обчислювальної складності мікроконтролера, стійкості

до зовнішніх збурень та необхідності наявності точного математичного опису механічної системи [13].

Класичний ПІД-регулятор історично залишається найпоширенішим і найбільш дослідженим рішенням у вбудованих системах керування завдяки простоті програмної імплементації та відсутності категоричної потреби у складній математичній моделі простору станів. Керуючий вплив у такому регуляторі формується як математична сума трьох незалежних складових: пропорційної, яка забезпечує миттєву реакцію на поточну похибку кута нахилу; інтегральної, що відповідає за компенсацію накопиченої статичної похибки та усунення дрейфу; та диференціальної, яка прогнозує швидкість зміни похибки, забезпечуючи необхідне демпфування коливань системи. Незважаючи на доведену ефективність в околі робочої точки рівноваги, класичний ПІД-алгоритм виявляє суттєві недоліки при роботі з сильно нелінійними об'єктами, оскільки його статичні коефіцієнти K_p , K_i та K_d не здатні адаптуватися, що призводить до зниження якості керування при значних відхиленнях кута θ або раптовій зміні маси корисного навантаження робота [14].

Більш кращим аналітичним підходом є застосування лінійно-квадратичного регулятора (LQR), який є фундаментальним методом теорії оптимального керування. На відміну від ПІД-регулятора, коефіцієнти якого зазвичай підбираються евристичним шляхом (наприклад, методом Зіглера - Нікольса), алгоритм LQR гарантує математично точне обчислення оптимальної матриці коефіцієнтів зворотного зв'язку шляхом мінімізації спеціального квадратичного критерію якості (функціоналу вартості). Цей функціонал дозволяє інженеру знайти ідеальний компроміс між швидкістю збіжності похибки до нуля та витратами енергії на керування двигунами. Проте практична ефективність LQR критично залежить від точності отриманої раніше лінійної математичної моделі. Цей факт суттєво ускладнює його автономне застосування в умовах параметричної невизначеності навколишнього середовища або при використанні дешевих інерціальних сенсорів (IMU) із високим рівнем високочастотних шумів [15]. Результати моделювання, що

демонструють часову залежність кута нахилу від часу для кожного з контролерів, наведено нижче (рис. 1.3).

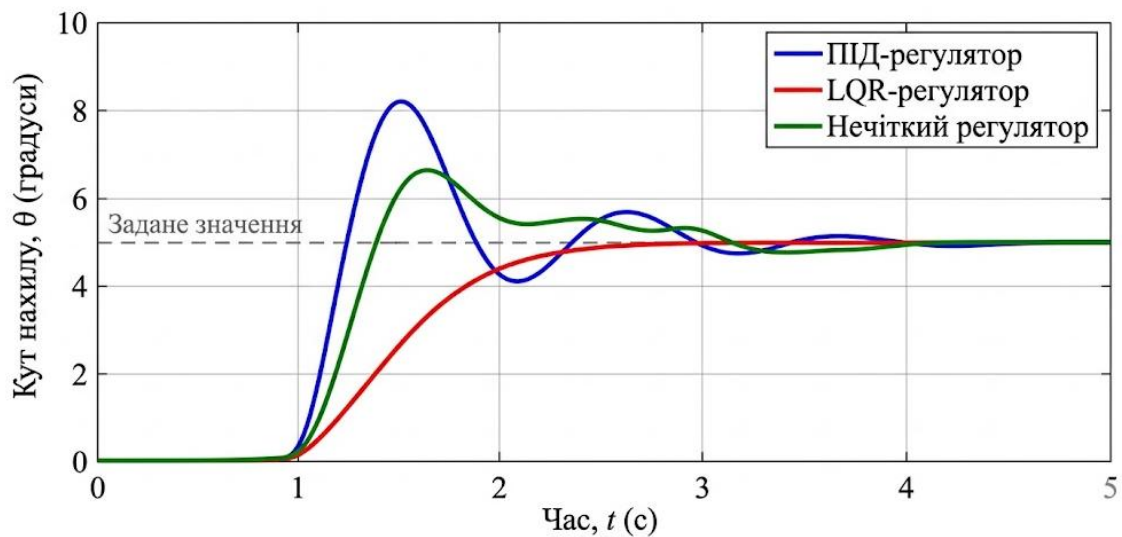


Рисунок 1.3 – Порівняльні графіки переходних процесів [15]

Потужною альтернативою класичним аналітичним методам виступають системи інтелектуального керування на базі нечіткої логіки (Fuzzy Logic Control). Цей математичний апарат імітує людське мислення, оперуючи не строгими числовими значеннями, а лінгвістичними змінними (наприклад, «надто великий кут нахилу», «мала кутова швидкість») та базою логічних продукційних правил формату «якщо-то». Головною перевагою нечіткого керування є здатність ефективно стабілізувати нелінійний об'єкт навіть за повної відсутності його математичної моделі. Водночас програмна імплементація блоків фазифікації, логічного виведення та дефазифікації вимагає значних апаратних ресурсів мікроконтролера. Високе обчислювальне навантаження суттєво ускладнює забезпечення детермінованості циклів керування в системах жорсткого реального часу при використанні базових 8-бітних процесорних ядер [16].

Проведений порівняльний аналіз демонструє, що для практичної реалізації системи стабілізації на базі бюджетних мікроконтролерів найдоцільнішим є використання дискретного ПІД-регулятора з каскадною архітектурою. Такий

вибір обґрунтовується необхідністю забезпечення оптимального балансу між обчислювальною складністю алгоритму, стабільною швидкістю контуру зворотного зв'язку (на рівні 100 Гц) та задоволенням вимог до якості перехідного процесу під час утримання вертикального положення корпусу балансуєчого робота [13, 14].

1.4 Огляд методів автономної навігації та детектування перешкод у локальному просторі

Після забезпечення базової динамічної стабілізації зворотного маятника наступним критичним етапом розробки автономного балансуєчого робота є синтез підсистеми просторової навігації та детектування перешкод. В умовах неструктурованого та динамічного середовища робот повинен не лише утримувати рівновагу, але й безпечно переміщуватися до заданої цільової точки. Для апаратного забезпечення мікроконтролерного рівня (наприклад, платформ сімейства ESP32 або Arduino) використання ресурсоємних алгоритмів глобальної навігації та одночасної побудови карти (SLAM) на базі комп'ютерного зору є ускладненим. Тому акцент зміщується на використання реактивних методів локальної навігації із застосуванням малогабаритних ультразвукових далекомірів, інфрачервоних сенсорів часу польоту (ToF) або бюджетних двовимірних лазерних сканерів (2D LiDAR) [17].

Для розробки ефективного алгоритму обходу перешкод мобільним роботом часто використовується метод штучних потенційних полів. Цей метод заснований на створенні віртуального силового поля в робочому просторі. Цільова точка генерує притягальну силу, яка діє на робота, а кожна перешкода – відштовхувальну силу. У будь-який момент часу на робота діє результуюча сила, яка визначає напрямок і швидкість його руху, що дозволяє йому автономно уникати зіткнень і прямувати до цілі.

Для наочного аналізу взаємодії сил у цьому методі доцільно використовувати векторні діаграми. Розглянемо випадок, коли робот, позначений

точкою, рухається до цілі, позначеної знаком «+», і на його шляху знаходиться перешкода, позначена знаком «-». У цьому випадку на робота діють вектор притягання до цілі та вектори відштовхування від перешкоди. Співвідношення та напрямки цих векторів ілюструє рисунок 1.4 [18].

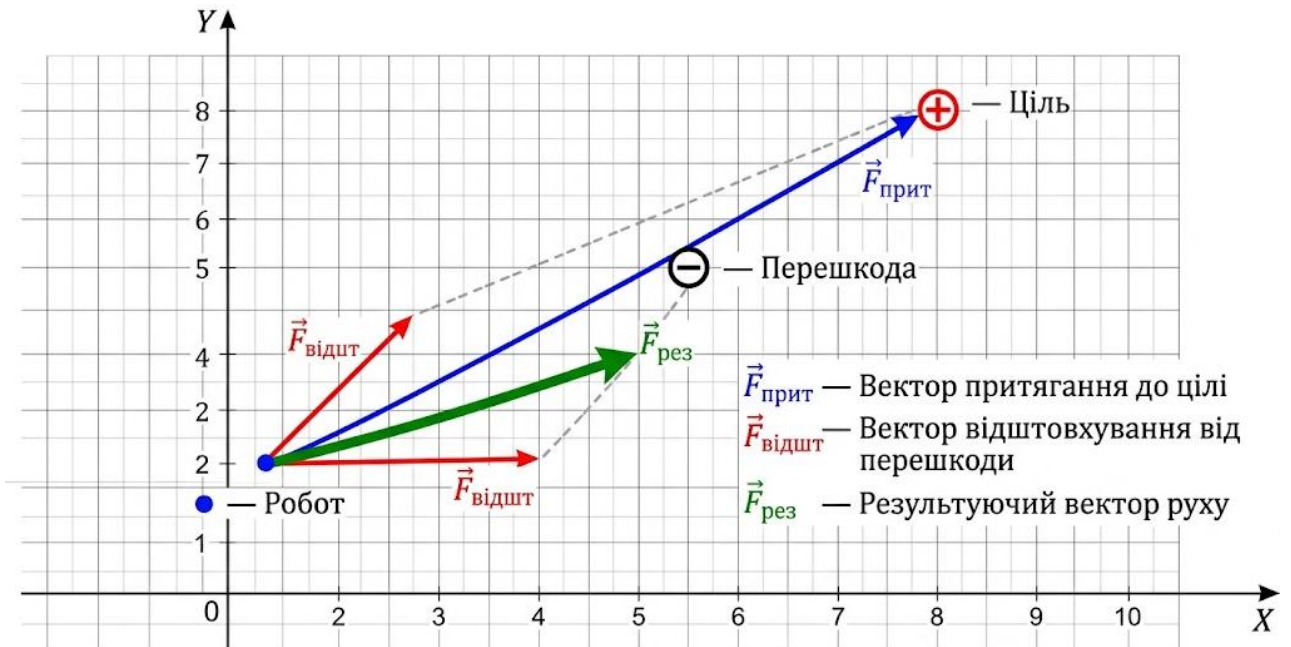


Рисунок 1.4 – Векторна діаграма сил у методі штучних потенційних полів при обході перешкоди [18]

Попри математичну елегантність та низьке обчислювальне навантаження, класичний метод APF має суттєвий недолік – ймовірність потрапляння системи у локальний мінімум, коли сили притягання та відштовхування зрівноважуються до досягнення цілі. У наукових дослідженнях 2024-2025 років для вирішення цієї проблеми пропонується використання вдосконалених функцій потенціалу, які враховують не лише відстань до об'єктів, але й відносну швидкість перешкод, а також впровадження алгоритмів віртуального обходу країв (Virtual Edge Following), які змушують робота рухатися вздовж еквипотенціальних ліній при виявленні застою [19].

Альтернативним підходом для систем із кінематичними обмеженнями є метод динамічних вікон (Dynamic Window Approach – DWA). Цей алгоритм генерує набір допустимих швидкостей та кутових прискорень робота на короткий

проміжок часу, проектує можливі траєкторії та оцінює їх за цільовою функцією, яка максимізує швидкість наближення до цілі та мінімізує ймовірність зіткнення. Поєднання реактивних алгоритмів обходу перешкод із класичним ПД регулятором балансування вимагає використання каскадної архітектури програмного забезпечення: зовнішній навігаційний контур з низькою частотою оновлення генерує еталонний кут нахилу, який потім відпрацьовується внутрішнім високочастотним контуром стабілізації.

На основі проведеного аналізу стану та перспектив розвитку двоколісних робототехнічних систем встановлено, що розробка балансуєчого робота є актуальною науково-технічною задачею. Побудована математична модель динаміки на основі формалізму Лагранжа підтвердила, що об'єкт є структурно нестійкою нелінійною системою. Порівняльний аналіз методів автоматичного керування довів, що для реалізації на базі мікроконтролера найбільш раціональним є використання дискретного ПД-регулятора завдяки його високій швидкодії та помірним вимогам до обчислювальних ресурсів. Для забезпечення автономності функціонування обґрунтовано доцільність застосування локальних реактивних алгоритмів навігації, зокрема методу штучних потенційних полів, інтегрованого з даними ультразвукових далекомірів. Сформована теоретична база дозволяє перейти до етапу апаратного проектування та алгоритмічної реалізації інтелектуальної системи керування.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА АПАРАТНО-ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ ПЛАТФОРМИ ARDUINO

2.1 Розробка структурної схеми апаратної архітектури та обґрунтування вибору мікроконтролера

Перехід від математичної моделі до практичної реалізації двоколісного балансуєчого робота вимагає розробки надійної апаратної архітектури, здатної забезпечити стабільне виконання алгоритмів керування в режимі жорсткого реального часу. Апаратний комплекс кіберфізичної системи робота концептуально поділяється на чотири взаємопов'язані підсистеми (рис. 2.1): центральний обчислювальний модуль (мікроконтролер), сенсорну підсистему збору інформації про просторове положення, виконавчу підсистему електроприводу та підсистему автономного енергозабезпечення. Взаємодія між цими модулями здійснюється через стандартні цифрові інтерфейси, зокрема I2C для підключення інерціальних датчиків та апаратний генератор ШІМ-сигналів (PWM) для керування драйверами двигунів [19].

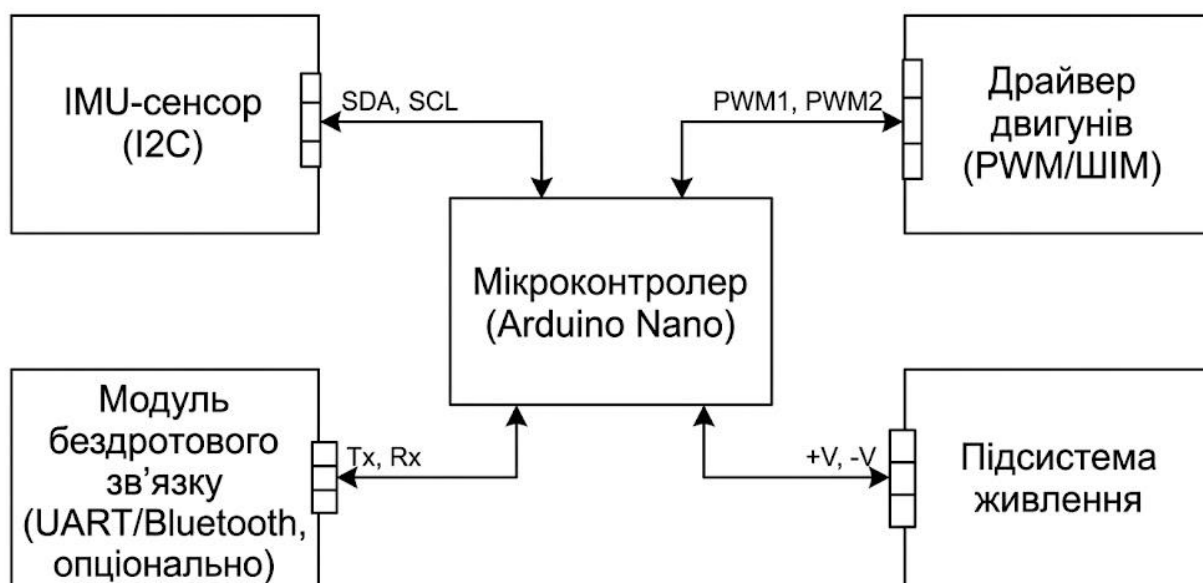


Рисунок 2.1 – Структурна схема апаратного комплексу двоколісного балансуєчого робота [19]

Центральною ланкою структурної схеми, яка визначає архітектуру програмного забезпечення, є мікроконтролер. Специфіка керування зворотним маятником накладає чіткі вимоги до частоти оновлення контуру стабілізації, яка для подібних малогабаритних платформ повинна становити близько 100 Гц. Для вирішення цього завдання в якості головного обчислювального ядра було обрано платформу Arduino Nano (рис. 2.2), побудовану на базі 8-бітного мікроконтролера сімейства AVR – ATmega328P. Вибір цієї архітектури обґрунтовується низкою вагомих інженерних та експлуатаційних факторів, актуальних для прототипування мобільних робототехнічних комплексів [9].

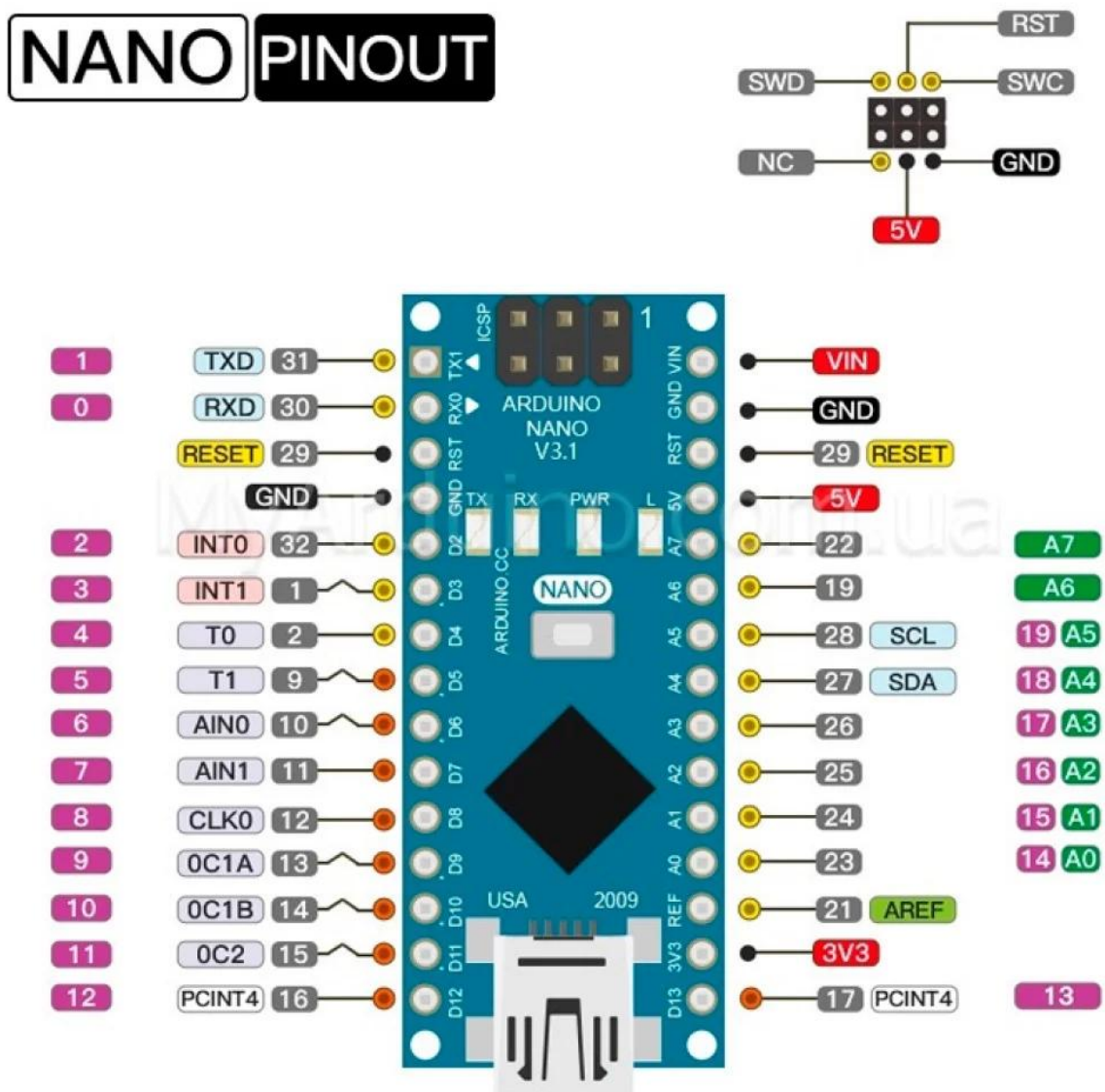


Рисунок 2.2 – Плата Arduino Nano V3.0 AtMega328P TypeC [20]

По-перше, мікроконтролер ATmega328P функціонує на тактовій частоті 16 МГц і володіє 32 КБ Flash-пам'яті та 2 КБ SRAM. Незважаючи на відсутність апаратного блоку операцій з рухомою комою (FPU), цих ресурсів цілком достатньо для реалізації дискретного ПД-регулятора за умови використання оптимізованих алгоритмів. Для зниження обчислювального навантаження на процесор замість ресурсоємного матричного фільтра Калмана застосовується комплементарний фільтр або використовується апаратний цифровий процесор руху (Digital Motion Processor – DMP), безпосередньо інтегрований у сучасні інерціальні модулі. Це дозволяє перенести складні тригонометричні обчислення на периферійний пристрій, залишаючи ядру мікроконтролера виключно задачі виконання логіки регулятора та генерації керуючих впливів [21].

По-друге, плата Arduino Nano має мінімальні масогабаритні показники (вага близько 7 грамів) та компактний форм-фактор. У контексті проектування балансує робота це є критичною перевагою, оскільки дозволяє оптимально розмістити мікроконтролер на верхніх ярусах шасі для підвищення центру мас системи, що, згідно з законами фізики, збільшує момент інерції та робить зворотний маятник менш чутливим до високочастотних збурень [22].

По-третє, екосистема AVR характеризується найвищим рівнем детермінованості під час роботи з апаратними перериваннями (Interrupts). Використання внутрішніх апаратних таймерів (Timer1 або Timer2) мікроконтролера ATmega328P гарантує суворе дотримання часових інтервалів (наприклад, рівно 10 мілісекунд) між ітераціями обчислення ПД-регулятора. Така стабільність кроку дискретизації є фундаментальною вимогою для коректної роботи диференціальної складової регулятора, яка надзвичайно чутлива до джиттеру (тимчасових затримок) у циклі керування [23, 24].

Таким чином, використання платформи Arduino Nano у поєднанні з алгоритмічною оптимізацією програмного коду забезпечує створення надійної, компактною та енергоефективної системи стабілізації.

2.2 Обґрунтування вибору сенсорної підсистеми та виконавчих механізмів

Ефективність функціонування мікропроцесорної системи керування двоколісним роботом критично залежить від точності вимірювання поточного стану об'єкта (кута нахилу та лінійного переміщення) і швидкості відпрацювання керуючих впливів. Відповідно до математичної моделі, описаної в підрозділі 1.2, для формування вектора стану системи необхідно безперервно отримувати значення кута відхилення маятника θ , його кутової швидкості $\dot{\theta}$, а також лінійної швидкості \dot{x} і положення x осі коліс. Для вирішення першої частини цього завдання в якості головного вимірювального елемента обрано мікроелектромеханічний (MEMS) інерціальний модуль MPU6050 (рис. 2.3).



Рисунок 2.3 – Модуль MPU6050 [25]

Вибір сенсора MPU6050 обґрунтовується оптимальним співвідношенням його технічних характеристик та обчислювальних можливостей обраного мікроконтролера Arduino Nano. Цей модуль об'єднує на одному кристалі триосьовий акселерометр та триосьовий гіроскоп, дані з яких оцифровуються за допомогою вбудованих 16-бітних аналого-цифрових перетворювачів (АЦП) і передаються на мікроконтролер по високошвидкісній шині I2C. Ключовою

апаратною перевагою MPU6050, яка робить його ідеальним для використання з 8-бітними процесорами сімейства AVR, є наявність вбудованого цифрового процесора обробки руху (Digital Motion Processor – DMP). Цей апаратний блок здатний самостійно виконувати складні алгоритми злиття сенсорних даних і розраховувати просторову орієнтацію у вигляді кватерніонів або кутів Ейлера. Використання DMP дозволяє повністю розвантажити ядро ATmega328P від важких тригонометричних обчислень, вивільняючи процесорний час для розрахунку ПІД-регулятора в режимі жорсткого реального часу [26].

На рисунку 2.3 зображено схему електричних з'єднань, яка ілюструє фізичне підключення IMU-сенсора MPU6050 до інтерфейсу I2C, драйвера двигунів TB6612FNG до ШІМ-виводів, а квадратурних енкодерів двигунів до портів переривань мікроконтролера Arduino Nano.

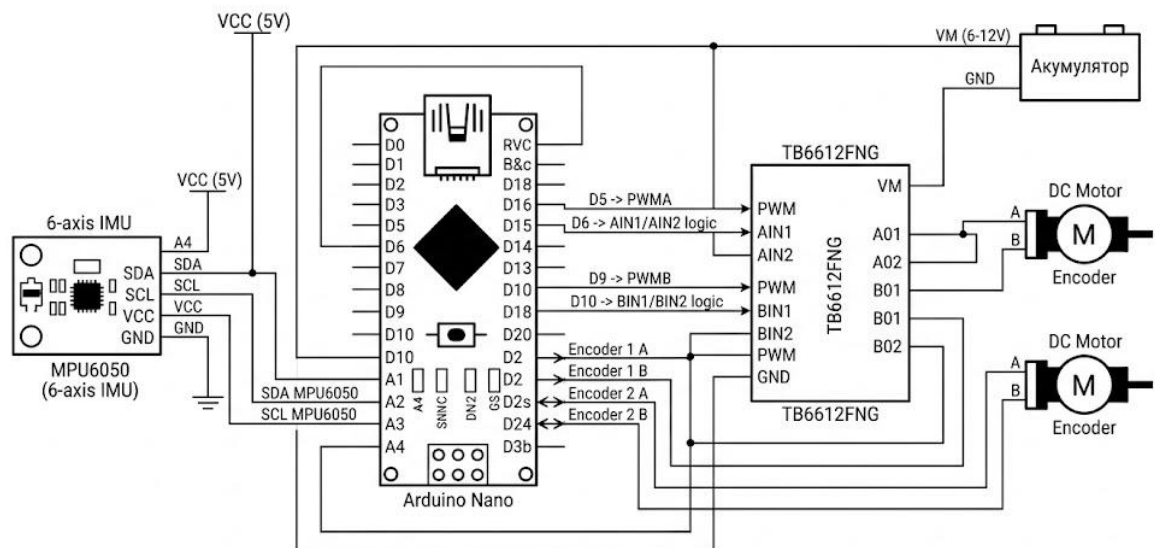


Рисунок 2.2 – Схема електричних з'єднань сенсорної підсистеми та драйвера двигунів з мікроконтролером

Виконавча підсистема робота відповідає за перетворення розрахованих мікроконтролером керуючих впливів (коригуючих моментів T) на реальний механічний рух. Для побудови колісної бази обрано колекторні двигуни постійного струму (DC-мотори) з інтегрованими циліндричними редукторами.

Наявність редуктора є обов'язковою умовою для забезпечення необхідного стартового крутного моменту при мінімальних швидкостях обертання під час балансування поблизу точки рівноваги. Обов'язковим елементом обраних двигунів є вбудовані квадратурні енкодери на ефекті Холла. Датчики Холла генерують імпульси, підрахунок яких за допомогою апаратних переривань Arduino Nano (піни D2 та D3) дозволяє з високою точністю визначати лінійне переміщення та швидкість робота, що необхідно для реалізації повноцінного каскадного керування [27].

Оскільки мікроконтролер не здатний самостійно комутувати великі струми живлення двигунів, структурною схемою передбачено використання драйвера на базі H-моста. Аналіз сучасної елементної бази показує, що класичні біполярні драйвери типу L298N морально застаріли через високе падіння напруги на транзисторах (до 2-3 В) та значне тепловиділення, що є критичним для систем з автономним акумуляторним живленням. Тому для даного проекту обрано драйвер TB6612FNG, побудований на польових транзисторах (MOSFET). Він характеризується мінімальним опором відкритого каналу, високим ККД та компактними розмірами, що ідеально відповідає вимогам до масогабаритних показників балансуємого робота. Керування драйвером здійснюється за допомогою широтно-імпульсної модуляції (ШІМ) з частотою не менше 4 кГц, що забезпечує лінійність тягової характеристики електроприводу та усуває акустичний шум в обмотках двигунів [28].

2.3 Алгоритмічна реалізація каскадної системи ПД-керування та розробка програмного забезпечення

Програмне забезпечення мікроконтролера Arduino Nano є ключовим компонентом, що об'єднує апаратні модулі в єдину кіберфізичну систему та реалізує математичну модель стабілізації. Зважаючи на обмежені обчислювальні ресурси 8-бітної архітектури ATmega328P, архітектура програмного забезпечення побудована за принципом нескінченного циклу (super-loop) із

використанням апаратних переривань (Interrupt Service Routines – ISR). Для забезпечення суворої детермінованості системи в часі, що є критичною вимогою для коректного обчислення диференціальної та інтегральної складових регулятора, основний цикл керування синхронізується за допомогою вбудованого таймера. Частота оновлення (крок дискретизації) встановлена на рівні 100 Гц ($\Delta t = 10$ мс), що забезпечує достатній запас стійкості за фазою для механічної системи з подібною масою та моментом інерції [29].

Практична експлуатація двоколісних балансуючих платформ доводить, що використання єдиного контуру керування кутом нахилу є недостатнім для забезпечення повноцінної автономності. Навіть ідеально налаштований ПІД-регулятор кута з часом призводить до поступового дрейфу робота (неконтрольованого лінійного переміщення) через мікроскопічні похибки в калібруванні гіроскопа, несиметричність розподілу мас або нерівності поверхні. Тому в даній роботі програмно реалізовано архітектуру каскадного керування, яка складається з двох взаємопов'язаних контурів: внутрішнього контуру стабілізації кута та зовнішнього контуру керування лінійною швидкістю і положенням [30].

Внутрішній (швидкий) контур відповідає безпосередньо за утримання вертикального положення корпусу. На кожному кроці дискретизації мікроконтролер зчитує поточний кут нахилу θ з інерціального модуля MPU6050 і порівнює його з еталонним (заданим) кутом θ_{ref} . На основі отриманої похибки $e[k] = \theta_{ref} - \theta[k]$ обчислюється керуючий вплив за формулою дискретного ПІД-алгоритму (2.1):

$$u[k] = K_p e[k] + K_i \sum_{i=0}^k e[i] \Delta t + K_d \frac{e[k] - e[k-1]}{\Delta t}, \quad (2.1)$$

де K_p, K_i, K_d – відповідно пропорційний, інтегральний та диференціальний коефіцієнти підсилення.

Для запобігання ефекту насичення інтегратора (Integral Windup), який виникає при тривалих відхиленнях системи (наприклад, під час зіткнення з перешкодою), у програмному коді реалізовано алгоритм динамічного обмеження акумульованої суми похибок [31].

Зовнішній (повільний) контур керування базується на даних одометрії, що надходять від квадратурних енкодерів двигунів. Мікроконтролер підраховує кількість імпульсів за допомогою зовнішніх апаратних переривань, обчислюючи поточну лінійну швидкість осі коліс. Завдання зовнішнього ПІД-регулятора швидкості полягає в тому, щоб генерувати такий еталонний кут нахилу θ_{ref} для внутрішнього контуру, який змусить робота змістити центр мас у потрібному напрямку для компенсації дрейфу або початку цілеспрямованого руху. Наприклад, якщо робот мимовільно котиться вперед, зовнішній контур задає невеликий негативний кут θ_{ref} , змушуючи внутрішній контур відхилити корпус назад і зупинити рух [32].

На рисунку 2.3 представлена блок-схема алгоритму функціонування каскадної системи керування роботом.

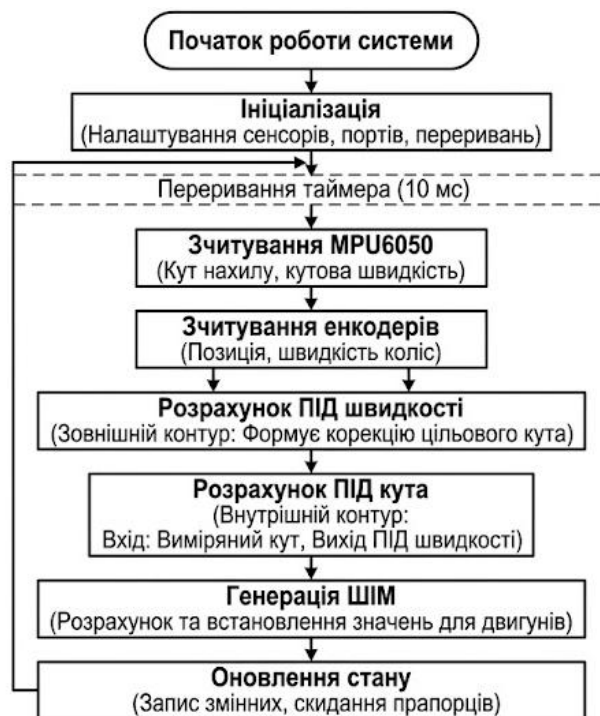


Рисунок 2.3 – Блок-схема алгоритму функціонування каскадної системи керування роботом

Після початкової ініціалізації система входить у головний цикл, який виконується кожні 10 мс за допомогою переривання таймера. Алгоритм наочно демонструє каскадну структуру керування: зовнішній контур ПІД-регулятора швидкості формує корекцію цільового кута, який стає входом для внутрішнього контуру ПІД-регулятора кута. На основі цих розрахунків генеруються ШІМ-сигнали для двигунів, і цикл повторюється після оновлення стану системи.

Фінальним етапом програмного циклу є масштабування розрахованого керуючого впливу $u[k]$ у відповідні значення шпаруватості широтно-імпульсної модуляції (ШІМ) для драйвера двигунів TB6612FNG. Оскільки колекторні двигуни постійного струму мають значну зону нечутливості (deadband) через сухе тертя в редукторах, програмне забезпечення включає блок нелінійної компенсації. Цей блок додає до розрахованого керуючого сигналу базову константу пускової напруги, гарантуючи, що навіть при мінімальних відхиленнях кута двигуни миттєво почнуть генерувати коригуючий крутний момент, запобігаючи виникненню низькочастотних автоколивань корпусу поблизу точки рівноваги [33].

2.4 Розробка підсистеми автономної навігації та планування шляху

Останнім етапом проєктування апаратно-програмного комплексу є розробка підсистеми автономної навігації, яка дозволяє двоколісному роботу не лише стабільно утримувати вертикальне положення, але й безпечно переміщуватися в неструктурованому локальному просторі. Враховуючи використання 8-бітного мікроконтролера Arduino Nano, реалізація ресурсоємних методів глобальної навігації є неможливою. Тому в даній роботі підсистема навігації побудована на базі реактивних алгоритмів локального планування шляху, які базуються на безпосередніх показах далекомірних сенсорів.

Основним алгоритмічним рішенням для планування траєкторії обрано метод штучних потенційних полів (Artificial Potential Fields – APF), адаптований для систем із жорсткими часовими обмеженнями. Математична суть методу

полягає у формуванні віртуального силового вектора, який є результатом суперпозиції сили притягання до цільової точки та сил відштовхування від виявлених перешкод. Оскільки обчислення логарифмічних або експоненціальних функцій потенціалу є обтяжливим для ядра ATmega328P, у програмному забезпеченні використано спрощену поліноміальну апроксимацію функцій поля, що дозволяє розраховувати результуючий вектор руху на кожній 10-й ітерації основного циклу керування (з частотою 10 Гц).

Інтеграція підсистеми навігації в загальну архітектуру керування реалізована через каскадний принцип передачі завдань. Навігаційний алгоритм обробляє дані з ультразвукового датчика та енкодерів, після чого формує необхідний кут повороту та зміщення еталонного кута нахилу θ_{ref} для внутрішнього ПІД-регулятора. Такий підхід забезпечує плавне маневрування: при наближенні до перешкоди сила відштовхування змушує навігаційний контур змістити θ_{ref} у протилежний бік, ініціюючи нахил корпусу і, як наслідок, гальмування або поворот робота.

На рисунку 2.4 показано програмну структуру взаємодії навігаційного контуру та системи стабілізації.

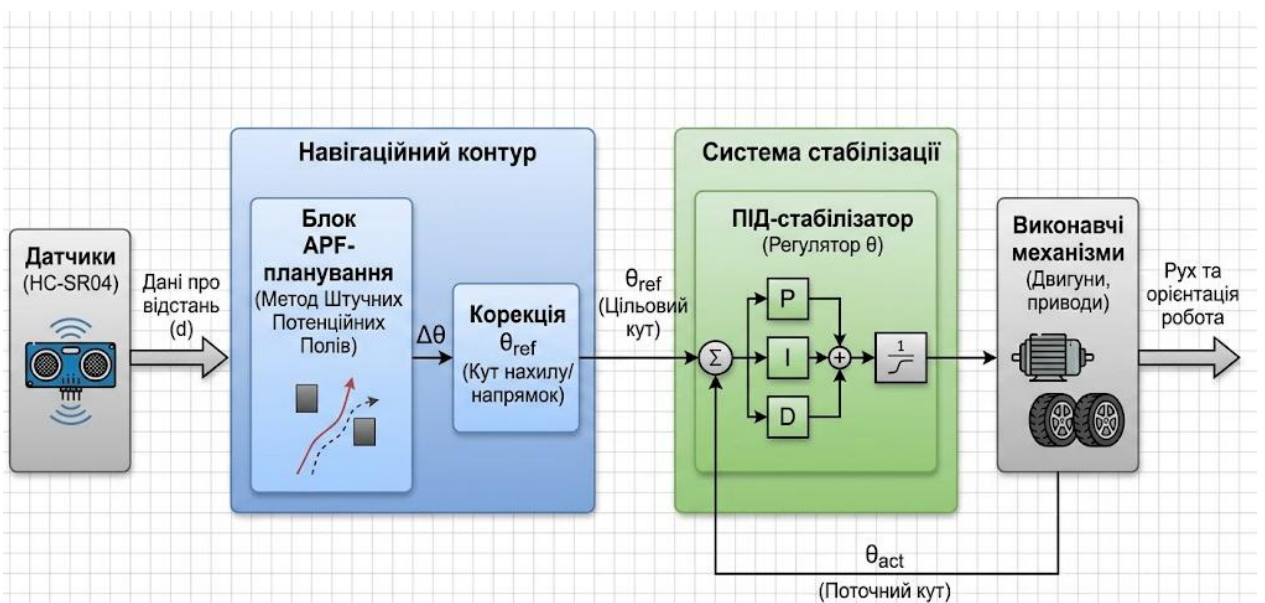


Рисунок 2.4 – Програмна структура взаємодії навігаційного контуру та системи стабілізації.

Для запобігання потраплянню робота в стан «заиклення» у локальних мінімумах (наприклад, перед широкою плоскою стіною) до алгоритму додано блок випадкових збурень, який при нульовій результуючій силі протягом заданого часу ініціює невеликий розворот робота навколо своєї осі. Це дозволяє системі змінити ракурс огляду та знайти вільний коридор для подальшого руху. Розроблена підсистема навігації у поєднанні з оптимізованим програмним кодом забезпечує високий рівень автономності робота при збереженні стабільності динамічного балансування.

На основі аналізу технічних вимог та критерію обчислювальної достатності обґрунтовано вибір 8-бітної мікроконтролерної платформи Arduino Nano, яка завдяки високому рівню детермінованості обробки апаратних переривань забезпечує стабільну частоту циклу керування на рівні 100 Гц.

Для забезпечення прецизійного вимірювання просторової орієнтації інтегровано інерціальний MEMS-модуль MPU6050, використання вбудованого цифрового процесора руху (DMP) якого дозволило мінімізувати обчислювальне навантаження на головний процесор. Вибір виконавчих механізмів у складі колекторних двигунів з редукторами та квадратурними енкодерами у поєднанні з сучасним MOSFET-драйвером TB6612FNG забезпечив необхідну динаміку відгуку системи при збереженні високої енергоефективності акумуляторного живлення.

Програмна реалізація системи базується на каскадній архітектурі ПІД регулювання, де внутрішній контур відповідає за динамічну стабілізацію кута нахилу, а зовнішній контур, оперуючи даними одометрії, компенсує лінійний дрейф та забезпечує точне позиціонування платформи. Розроблена підсистема автономної навігації на основі методу штучних потенційних полів дозволяє роботу в режимі реального часу детектувати перешкоди та адаптувати траєкторію руху, що у сукупності з алгоритмами стабілізації формує цілісну систему керування автономним мобільним об'єктом. Сформована апаратно-програмна база є достатньою для переходу до етапу експериментальної апробації та фінального налаштування коефіцієнтів регулятора

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОТОТИПУ БАЛАНСУЮЧОГО РОБОТА

3.1 Складання апаратної частини прототипу

Складання апаратної частини прототипу балануючого робота виконувалося на основі попередньо обраної елементної бази, до якої входять мікроконтролерна плата керування, інерціальний сенсор, ультразвуковий датчик відстані, два двигуни постійного струму з редукторами, колеса, монтажні пластини, акумуляторний відсік, стійки, гвинти та з'єднувальні провідники. Сукупність цих компонентів (рис. 3.1) забезпечує можливість створення мобільної платформи, здатної підтримувати рівновагу, визначати кут нахилу, керувати рухом коліс та орієнтуватися в навколишньому середовищі.

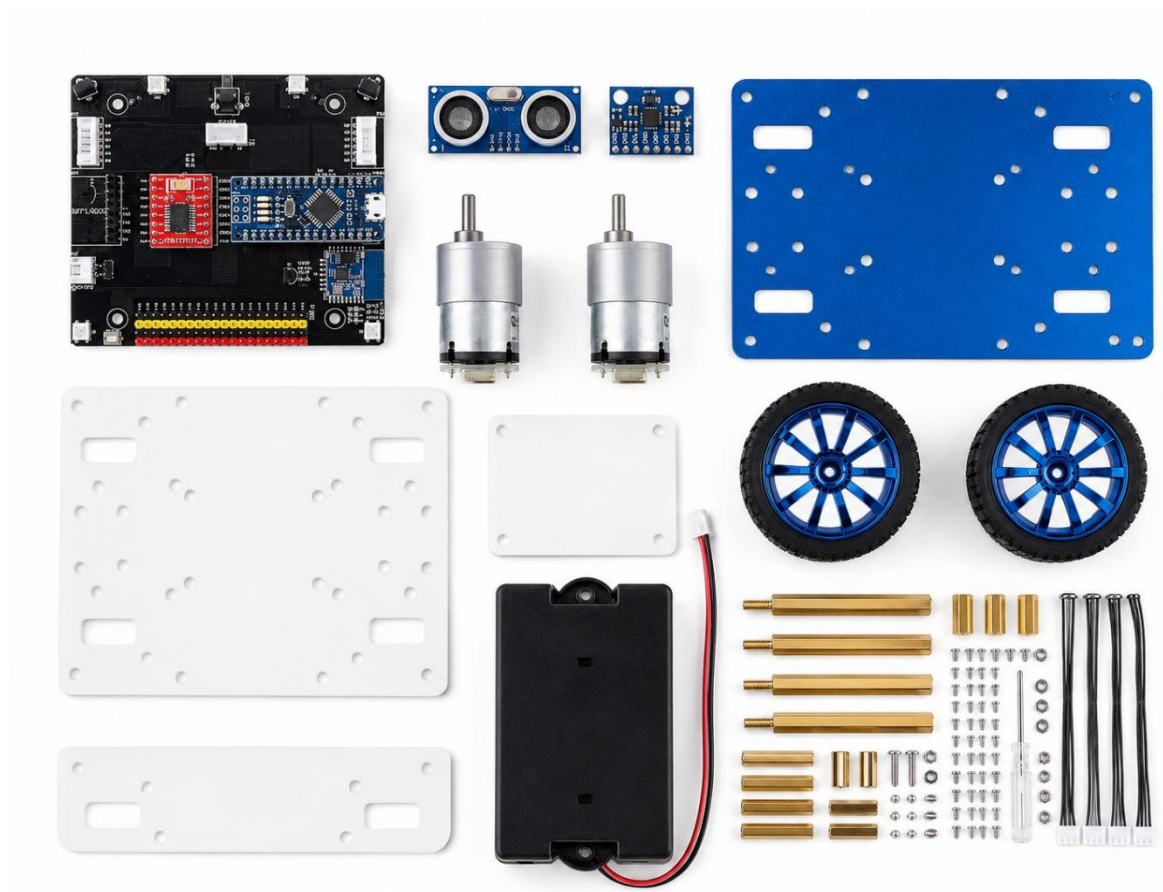


Рисунок 3.1 – Комплект апаратних компонентів для складання балануючого робота

На першому етапі було виконано підготовку основних конструктивних елементів корпусу (рис. 3.2). Для цього використовувалися монтажні пластини, які виконують роль несучої основи для розміщення плати керування, двигунів, сенсорів і блока живлення. Перед початком складання було перевірено наявність необхідних отворів для кріплення, відповідність розташування монтажних місць і можливість симетричного розміщення основних вузлів системи. Такий підхід є важливим, оскільки балансує робот потребує рівномірного розподілу маси відносно центральної вертикальної осі.



Рисунок 3.2 – Підготовка монтажних пластин корпусу балансуєчого робота

Другим етапом стало встановлення двигунів постійного струму з редукторами (рис. 3.3). Двигуни монтувалися на нижній частині конструкції таким чином, щоб їхні вали були розташовані на одній горизонтальній осі. Це забезпечує однакове передавання обертового моменту на обидва колеса та зменшує ризик перекосу під час руху. Надійність кріплення двигунів має важливе

значення, оскільки під час стабілізації вони постійно змінюють швидкість і напрям обертання відповідно до сигналів мікроконтролера.



Рисунок 3.3 – Монтаж двигунів постійного струму з редукторами на основу прототипу

Після встановлення двигунів було виконано монтаж коліс (рис. 3.4). Колеса закріплювалися безпосередньо на валах мотор-редукторів. На цьому етапі перевірялася відсутність люфтів, рівномірність обертання та надійність фіксації коліс. Для балансу роботи якість встановлення коліс є критично важливою, оскільки навіть незначне биття або перекіс можуть спричинити нестабільність під час утримання вертикального положення.



Рисунок 3.4 – Встановлення коліс на вали мотор-редукторів балансуємого робота

Наступним етапом було розміщення основної плати керування (рис. 3.5). Плату доцільно встановлювати ближче до центральної частини корпусу, оскільки таке розташування спрощує підключення сенсорів, двигунів і живлення, а також сприяє більш рівномірному розподілу маси. Плата керування виконує роль центрального обчислювального вузла, який приймає дані від сенсорів, обробляє їх і формує керуючі сигнали для виконавчих елементів.

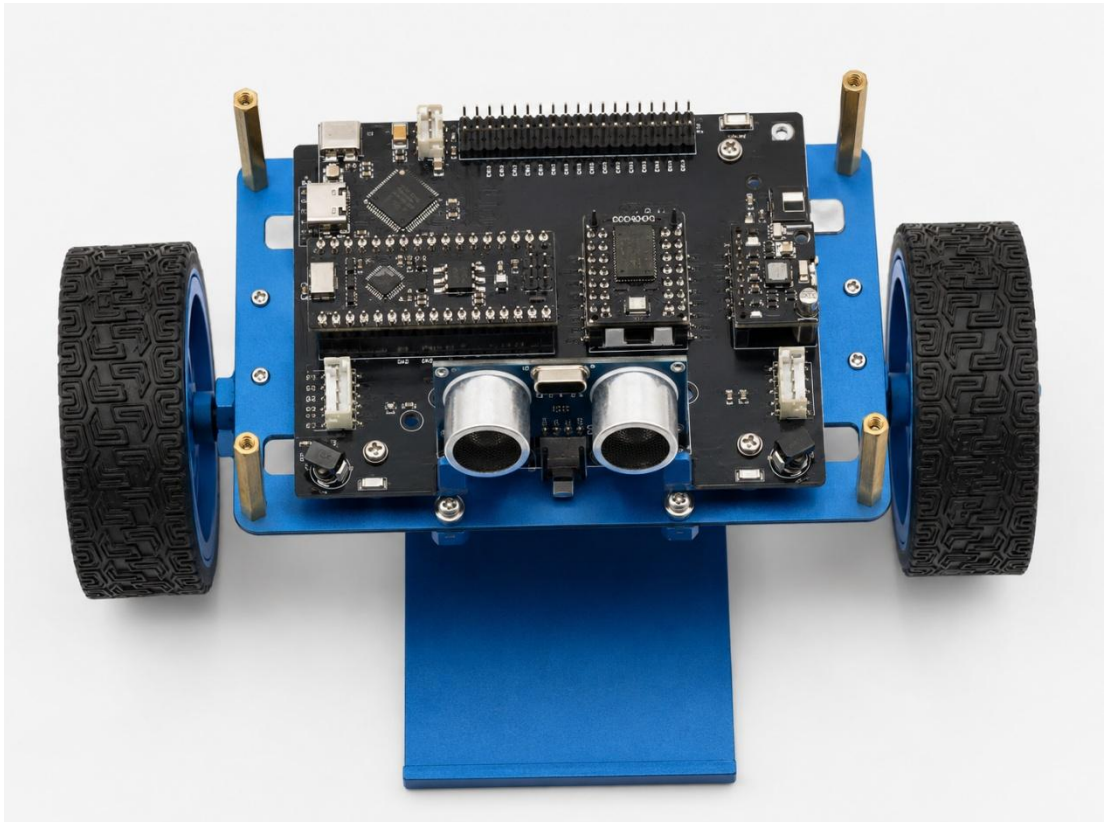


Рисунок 3.5 – Встановлення плати керування на корпус балануючого робота та ультразвукового датчика HC-SR04

Окремим етапом є підключення інерціального сенсора MPU6050, який використовується для визначення кута нахилу та кутової швидкості корпусу робота. Цей модуль містить акселерометр і гіроскоп, тому є основним джерелом даних для алгоритму стабілізації. Сенсор доцільно розміщувати якомога ближче до центру конструкції, щоб отримані вимірювання максимально точно відображали положення всієї платформи, а не окремої її частини.

Для реалізації навігаційної складової було встановлено ультразвуковий датчик відстані HC-SR04. Він розміщується у передній частині робота та використовується для виявлення перешкод у напрямку руху. Принцип його роботи ґрунтується на випромінюванні ультразвукового імпульсу та вимірюванні часу його повернення після відбиття від об'єкта. Отримана інформація дає змогу системі приймати рішення щодо зупинки, зміни напрямку або коригування траєкторії руху.

Далі було змонтовано акумуляторний відсік (рис. 3.6), який забезпечує автономне живлення апаратної частини прототипу. Під час його встановлення враховувалося те, що акумуляторний блок має відносно значну масу, тому його розміщення впливає на центр ваги робота. Найбільш доцільним є встановлення джерела живлення у нижній або центральній частині конструкції, що сприяє покращенню стійкості та зменшує ймовірність різкого перекидання робота під час запуску.

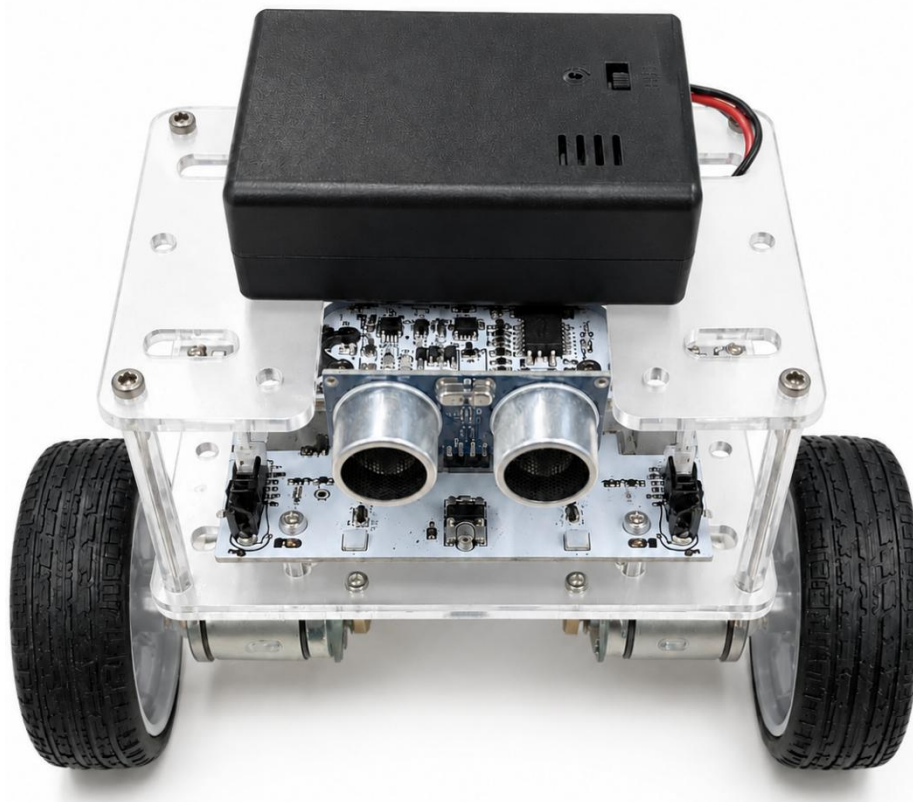


Рисунок 3.6 – Розміщення акумуляторного відсіку в конструкції балансуєчого робота

Після встановлення основних вузлів було виконано електричне з'єднання компонентів. Двигуни підключалися до відповідних виходів плати керування, сенсор MPU6050 – до інтерфейсних виводів мікроконтролера, ультразвуковий датчик – до цифрових входів і виходів, а акумуляторний блок – до силової частини системи. На цьому етапі важливо перевірити правильність полярності живлення, надійність контактів і відсутність коротких замикань.

Завершальним етапом стало формування загального вигляду апаратного прототипу та перевірка правильності складання (рис. 3.7). Після монтажу всіх компонентів було оцінено симетричність конструкції, міцність кріплень, доступність роз'ємів для програмування та можливість подальшого налаштування системи. Отриманий апаратний прототип є основою для подальшої програмної реалізації алгоритмів стабілізації, калібрування сенсорів, керування двигунами та тестування навігаційних функцій.



Рисунок 3.7 – Загальний вигляд зібраного прототипу балансуєчого робота

В результаті виконаного завдання було сформовано апаратну платформу балансуєчого робота, яка поєднує механічну основу, сенсорну підсистему,

мікроконтролерний блок керування, виконавчі елементи та автономне джерело живлення. Така структура забезпечує можливість подальшої реалізації інтелектуальної системи стабілізації та навігації на програмному рівні.

3.2 Розробка алгоритму стабілізації положення робота на основі даних інерціального сенсора

Основним завданням балансуючого робота є утримання вертикального положення під час руху та компенсація відхилень, які виникають унаслідок зміни положення корпусу, нерівномірності поверхні, інерції під час старту або зупинки, а також зовнішніх впливів. Для цього в системі використовується інерціальний сенсор MPU6050, який поєднує в одному модулі акселерометр і гіроскоп. Акселерометр дає змогу оцінювати нахил корпусу відносно осей простору, а гіроскоп визначає кутову швидкість зміни положення. Поєднання цих даних забезпечує більш точне визначення поточного кута нахилу робота, що є основою для подальшого формування керуючого сигналу на двигуни.

У процесі розробки алгоритму стабілізації положення робота враховувалося, що балансуючий робот є нестійкою динамічною системою. На відміну від звичайної колісної платформи, яка має декілька точок опори, балансуючий робот фактично утримується у вертикальному положенні за рахунок постійного коригування руху коліс. Якщо корпус нахилиється вперед, двигуни повинні обертати колеса в напрямку нахилу, щоб повернути центр мас до положення рівноваги. Якщо корпус відхиляється назад, двигуни мають змінити напрям обертання або зменшити швидкість так, щоб компенсувати відхилення. Тому алгоритм стабілізації повинен працювати циклічно, з мінімальною затримкою між зчитуванням даних сенсора та подаванням сигналу на двигуни.

Загальна логіка роботи алгоритму (рис. 3.8) стабілізації полягає у послідовному виконанні таких дій: ініціалізація мікроконтролера та сенсора MPU6050, зчитування значень акселерометра і гіроскопа, фільтрація отриманих

даних, обчислення кута нахилу корпусу, порівняння поточного кута із заданим рівноважним положенням, розрахунок помилки стабілізації, формування керуючого сигналу та передавання його на двигуни. Після цього цикл повторюється, що дає змогу системі безперервно реагувати на зміну положення робота.

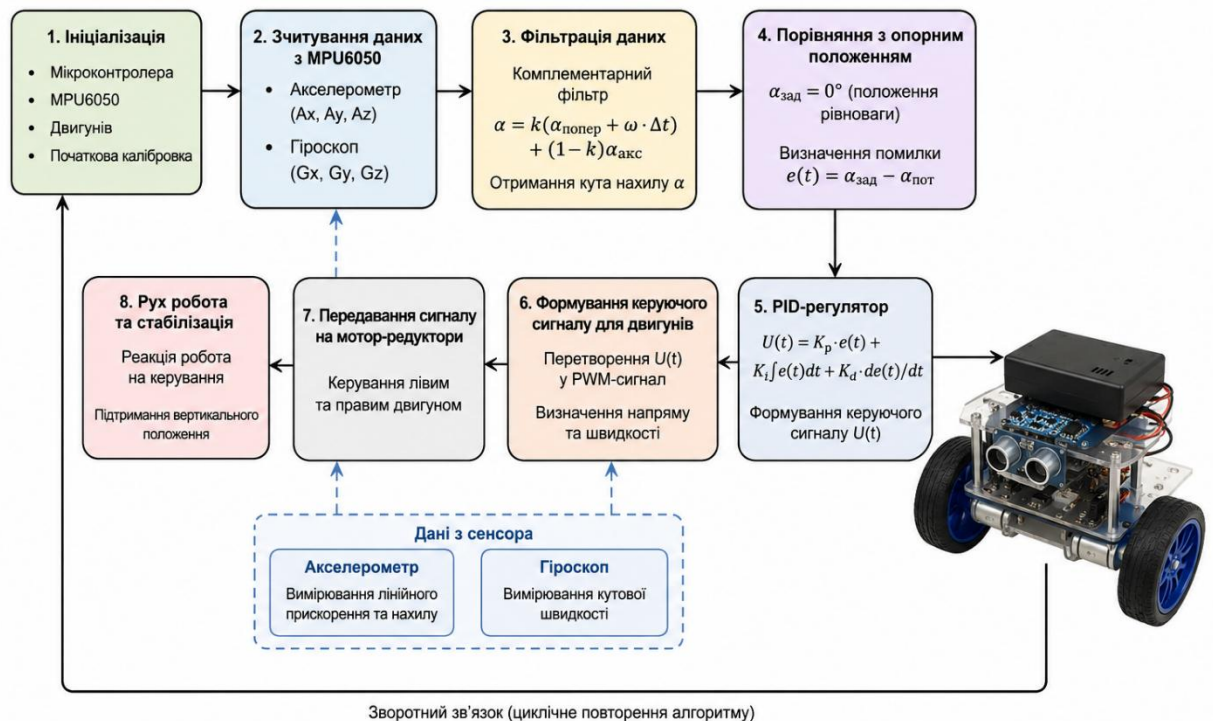


Рисунок 3.8 – Загальна структура алгоритму стабілізації положення балансуючого робота

Першим етапом алгоритму є ініціалізація апаратних і програмних компонентів (рис. 3.9). На цьому етапі мікроконтролер налаштовує інтерфейс обміну даними з інерціальним сенсором, перевіряє його доступність, задає початкові параметри вимірювання та виконує підготовку виходів керування двигунами. Також може здійснюватися початкова калібровка MPU6050, під час якої визначаються нульові зміщення гіроскопа та акселерометра. Це необхідно для зменшення похибки вимірювання, оскільки навіть у нерухомому стані сенсор може передавати незначні відхилення від нульових значень.

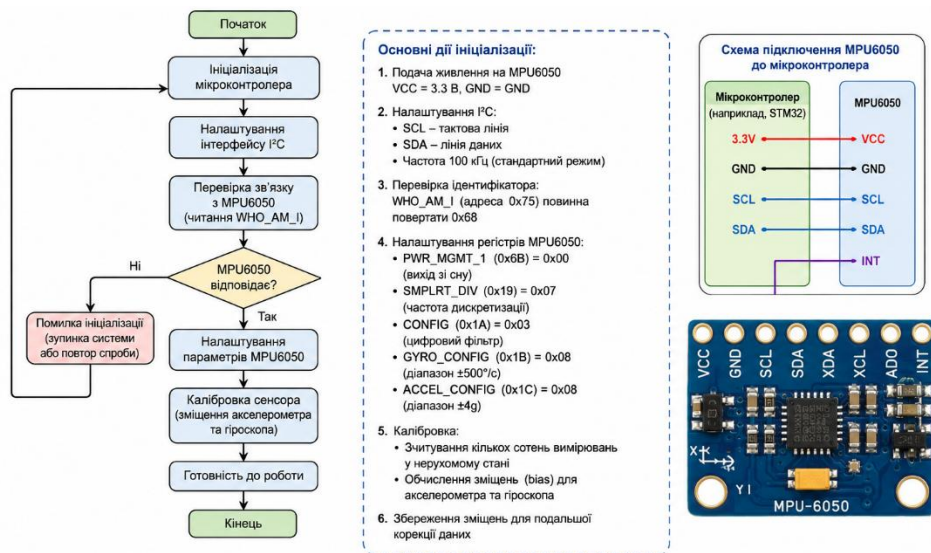


Рисунок 3.9 – Етап ініціалізації мікроконтролера та інерціального сенсора MPU6050

Другим етапом є зчитування даних з інерціального сенсора (рис. 3.10). MPU6050 передає мікроконтролеру значення прискорення вздовж осей X, Y, Z та значення кутової швидкості. Для задачі стабілізації найбільше значення має вісь, відносно якої відбувається нахил корпусу робота вперед або назад. Отримані дані не можна безпосередньо використовувати для керування двигунами, оскільки вони можуть містити шум, вібраційні перешкоди та короточасні коливання, спричинені роботою мотор-редукторів.

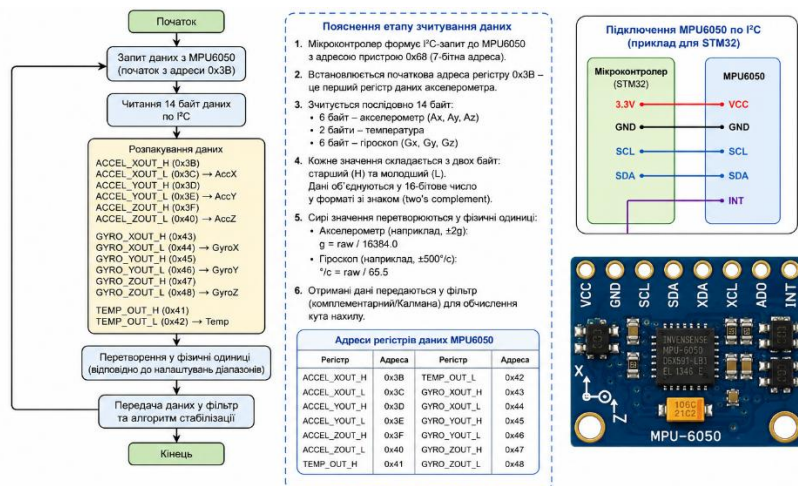


Рисунок 3.10 – Зчитування даних акселерометра та гіроскопа з модуля MPU6050

Для підвищення точності визначення кута нахилу застосовується фільтрація даних. Одним із практичних підходів є використання комплементарного фільтра, який поєднує переваги акселерометра та гіроскопа (рис. 3.11). Гіроскоп добре реагує на швидкі зміни положення, але з часом може накопичувати похибку. Акселерометр, навпаки, дає змогу визначати нахил відносно сили тяжіння, однак є чутливим до вібрацій і прискорень під час руху. Комплементарний фільтр дозволяє об'єднати ці дані та отримати більш стабільне значення кута нахилу.

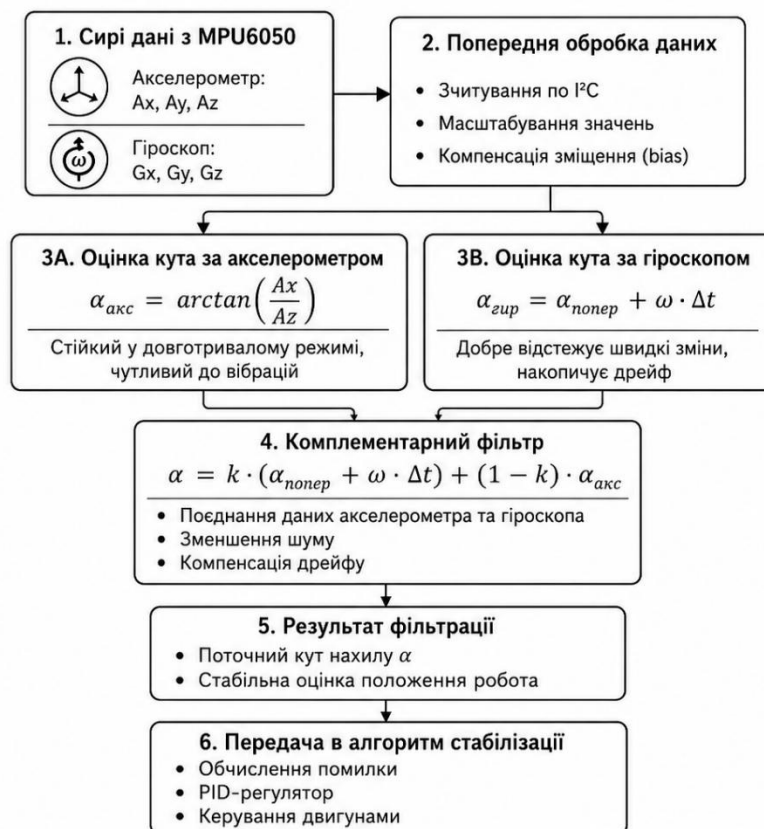


Рисунок 3.11 – Фільтрація даних інерціального сенсора для визначення кута нахилу

Після визначення поточного кута нахилу виконується порівняння з опорним значенням. Для балансуючого робота таким значенням є положення рівноваги, яке відповідає вертикальному розташуванню корпусу. На практиці це не завжди рівно 0° , оскільки реальний центр мас конструкції може бути трохи зміщений через розташування акумулятора, плати керування, сенсорів або

кріпильних елементів. Тому під час налаштування системи визначається робочий нульовий кут, за якого робот найкраще утримує рівновагу.

Для формування керуючого сигналу доцільно використати PID-регулятор, який враховує не лише поточне відхилення, а й швидкість його зміни та накопичену помилку за певний проміжок часу (3.12). Пропорційна складова відповідає за реакцію системи на поточний нахил, інтегральна – за компенсацію тривалого зміщення, а диференціальна - за згладжування різких змін і зменшення коливань. Використання PID-регулювання дає змогу забезпечити більш плавне та стійке балансування робота.

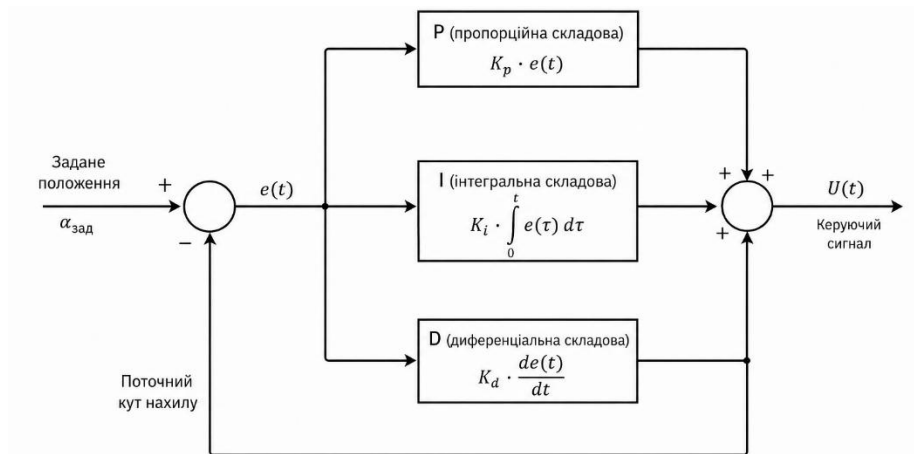


Рисунок 3.12 – Формування керуючого сигналу за допомогою PID-регулятора

На основі розрахованого керуючого сигналу мікроконтролер формує команди для двигунів. Якщо значення сигналу є додатним, двигуни обертаються в одному напрямку, а якщо від’ємним – у протилежному. Величина сигналу визначає швидкість обертання коліс, яка може регулюватися за допомогою широтно-імпульсної модуляції. Чим більшим є кут відхилення корпусу від рівноважного положення, тим інтенсивнішою має бути реакція двигунів. Водночас надто велике значення керуючого сигналу може спричинити ривки та коливання, тому параметри регулятора потрібно налаштовувати поступово (рис. 3.13).

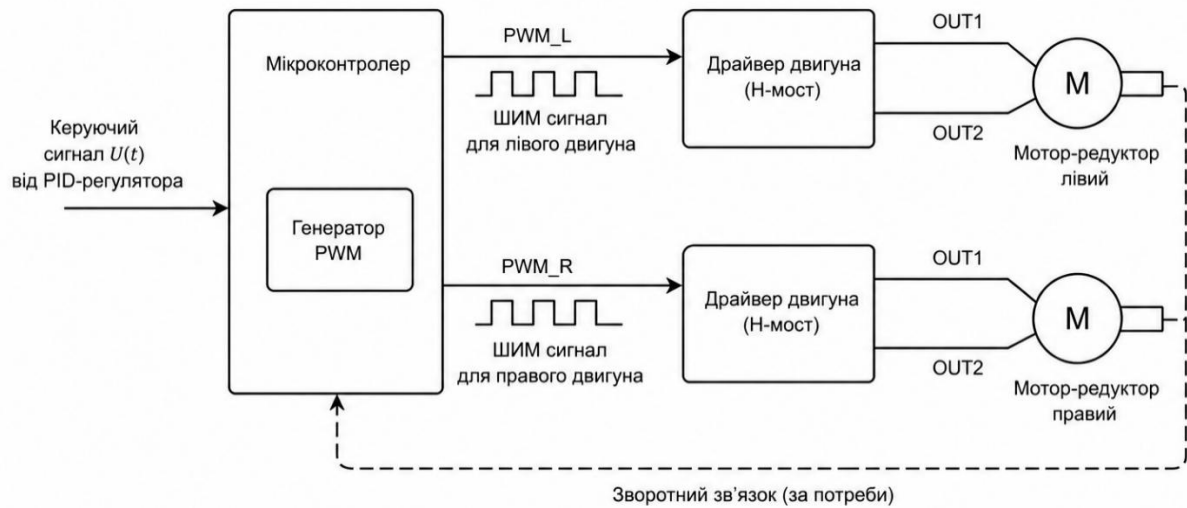


Рисунок 3.13 – Передавання керуючого сигналу від мікроконтролера до мотор-редукторів

Загальний алгоритм стабілізації можна подати у вигляді блок-схеми, наведеної на рисунку 3.14. Вона відображає послідовність основних дій, які виконує система керування для підтримання вертикального положення балансуємого робота. Алгоритм має циклічний характер, оскільки процес стабілізації повинен здійснюватися безперервно в режимі реального часу. Це означає, що мікроконтролер постійно зчитує дані з інерціального сенсора, обчислює поточний кут нахилу, визначає відхилення від заданого положення рівноваги та формує керуючі сигнали для мотор-редукторів.

На початковому етапі виконується ініціалізація системи, під час якої налаштовується мікроконтролер, інтерфейс обміну даними з MPU6050, драйвери двигунів, таймери та параметри PID-регулятора. Після цього проводиться калібрування інерціального сенсора, що передбачає визначення початкових зміщень акселерометра та гіроскопа.

Наступним етапом є зчитування даних з MPU6050. Сенсор передає мікроконтролеру значення прискорення та кутової швидкості, які характеризують просторове положення корпусу робота.

Після визначення кута нахилу система обчислює помилку стабілізації як різницю між заданим кутом рівноваги та поточним кутом нахилу робота.

Сформований керуючий сигнал додатково обмежується для захисту системи від перевищення допустимих значень. Це дозволяє уникнути надмірного навантаження на двигуни та різких ривків під час стабілізації

Завершальна частина алгоритму передбачає перевірку активності системи. Якщо система продовжує працювати, після короткої затримки циклу процес повторюється з етапу зчитування даних з MPU6050. Завдяки такій циклічній структурі робот здатний оперативно реагувати на зміну положення корпусу та підтримувати рівновагу. Якщо система вимикається або отримує команду зупинки, виконання алгоритму завершується.

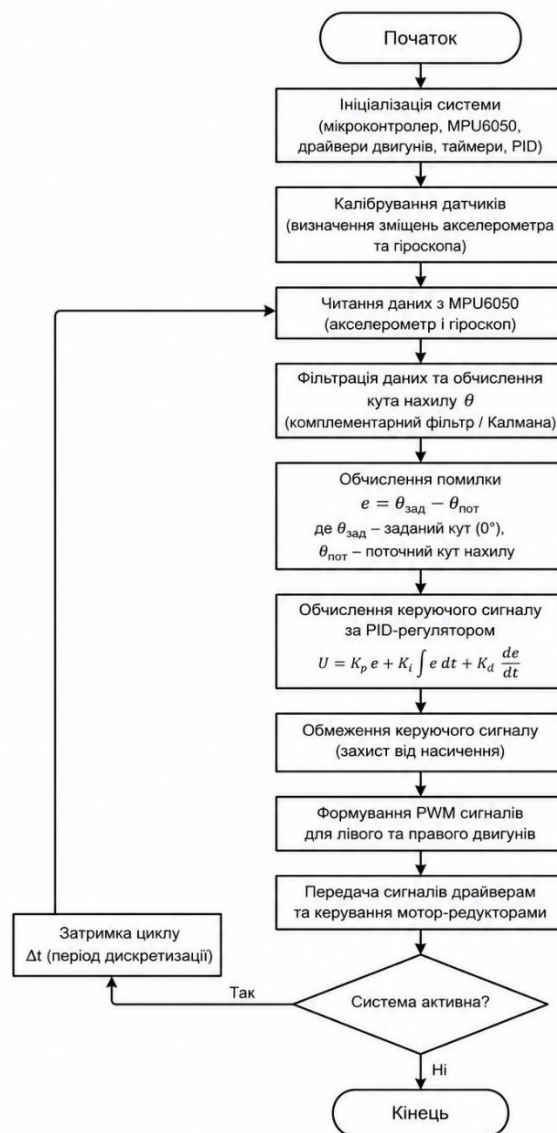


Рисунок 3.14 – Блок-схема алгоритму стабілізації положення балануючого робота

Важливою умовою ефективної роботи алгоритму є правильне налаштування коефіцієнтів PID-регулятора. Якщо коефіцієнт K_p занадто малий, робот буде повільно реагувати на відхилення та може втрачати рівновагу. Якщо K_p занадто великий, система почне різко реагувати на незначні нахили, що може спричинити коливання. Коефіцієнт K_i використовується для усунення сталої похибки, однак його надмірне значення може погіршити стійкість системи. Коефіцієнт K_d допомагає зменшити різкі коливання, оскільки враховує швидкість зміни помилки.

Під час практичного налаштування алгоритму доцільно спочатку встановити нульові значення K_i та K_d , а потім поступово збільшувати K_p до моменту, коли робот почне реагувати на нахил. Після цього додається диференціальна складова K_d для зменшення коливань. Інтегральна складова K_i вводиться обережно, якщо робот має постійне незначне зміщення вперед або назад. Така послідовність дає змогу уникнути надмірно різкої реакції системи та поступово досягти стабільного балансування.

Розроблений алгоритм стабілізації положення робота ґрунтується на безперервному аналізі даних інерціального сенсора MPU6050, обчисленні кута нахилу, визначенні помилки відносно положення рівноваги та формуванні керуючого сигналу для двигунів. Використання фільтрації сенсорних даних і PID-регулювання забезпечує можливість оперативної компенсації нахилу корпусу та підтримання робота у вертикальному положенні. Надалі цей алгоритм може бути поєднаний із навігаційним модулем, що дозволить реалізувати не лише стабілізацію, а й керований рух балансуючого робота в середовищі.

3.3 Програмна реалізація системи керування мікроконтролером

Програмна реалізація системи керування мікроконтролером є одним із ключових етапів створення балансуючого робота, оскільки саме програмне забезпечення забезпечує взаємодію між інерціальним сенсором, обчислювальним модулем, PID-регулятором і виконавчими механізмами.

Основне завдання програмної частини полягає в тому, щоб у режимі реального часу зчитувати дані з MPU6050, визначати поточний кут нахилу корпусу, розраховувати відхилення від рівноважного положення та формувати керуючі PWM-сигнали для мотор-редукторів.

Програмна логіка системи керування побудована за циклічним принципом. Після запуску мікроконтролер виконує ініціалізацію основних апаратних модулів: інтерфейсу I2C для обміну даними з MPU6050, портів введення-виведення, таймерів, PWM-каналів, драйверів двигунів та змінних PID-регулятора. Після цього система переходить до основного циклу роботи, у якому безперервно здійснюється зчитування сенсорних даних, їх фільтрація, обчислення керуючого впливу та передавання сигналів на двигуни (рис. 3.15)

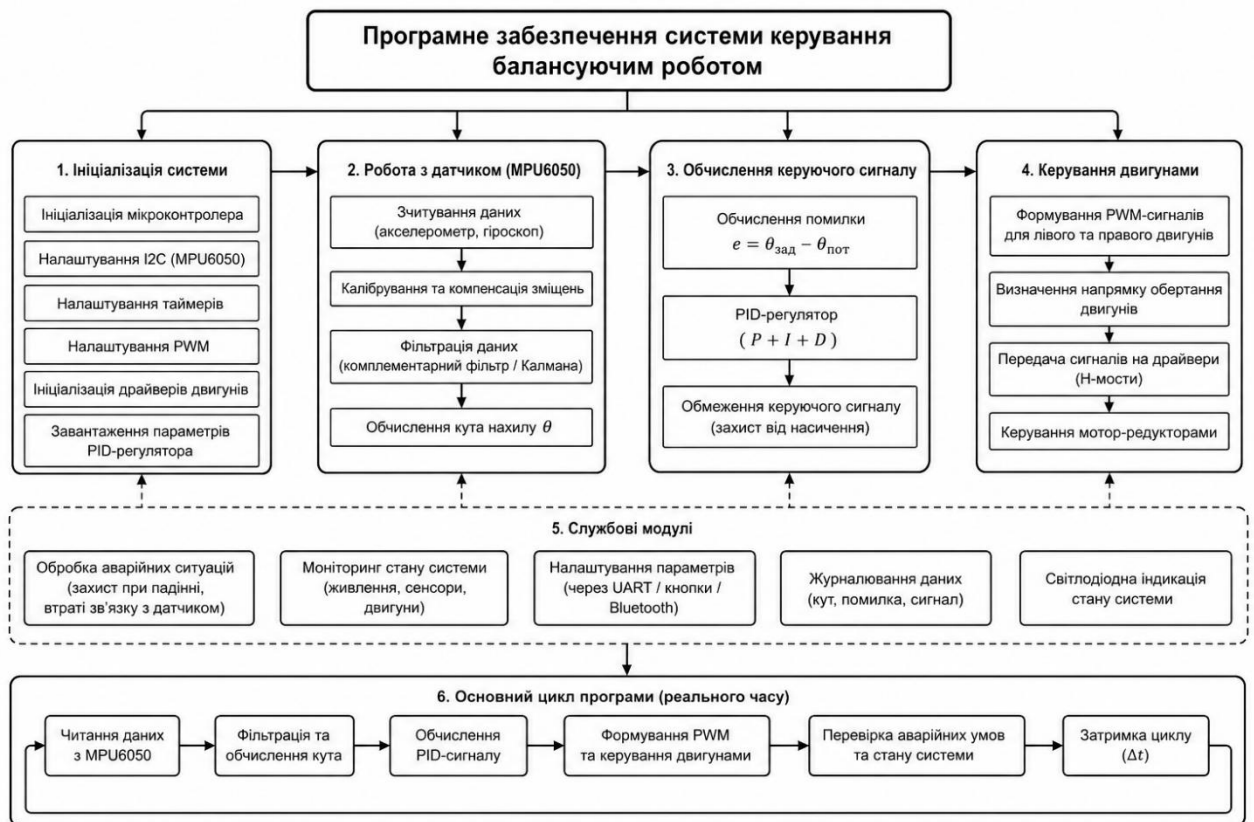


Рисунок 3.15 – Структура програмного забезпечення системи керування балансуючим роботом

На етапі ініціалізації задаються початкові параметри роботи системи

(рис. 3.16). Мікроконтролер встановлює швидкість обміну даними по шині I2C, перевіряє наявність модуля MPU6050, виводить сенсор із режиму сну та налаштовує діапазони вимірювання акселерометра і гіроскопа. Також визначаються початкові значення коефіцієнтів PID-регулятора, обмеження керуючого сигналу, напрямки обертання двигунів і частота формування PWM-сигналів. Таке налаштування є необхідним для узгодженої роботи всіх складових системи.

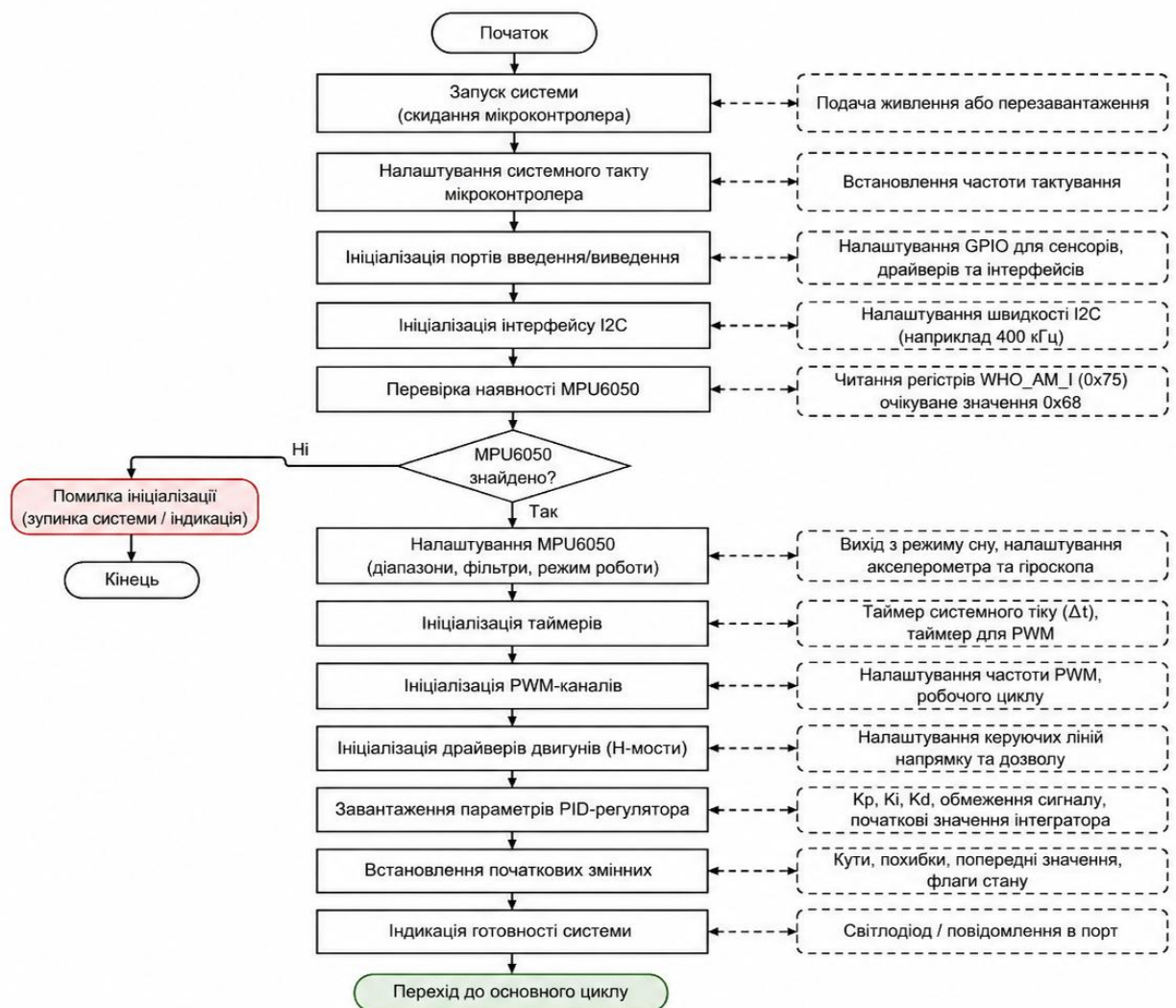


Рисунок 3.16 – Етап ініціалізації програмних модулів системи керування

Після завершення ініціалізації виконується калібрування інерціального сенсора. У процесі калібрування робот має перебувати у нерухомому положенні, щоб мікроконтролер міг визначити початкові зміщення акселерометра та

гіроскопа. Отримані поправки надалі використовуються для корекції вимірних значень. Це дозволяє зменшити систематичну похибку та підвищити точність визначення кута нахилу корпусу робота.

Основний цикл програми починається зі зчитування даних з MPU6050. Мікроконтролер отримує значення прискорення за осями A_x , A_y , A_z та значення кутової швидкості G_x , G_y , G_z . Для задачі стабілізації найбільше значення мають дані, які характеризують нахил робота вперед або назад, оскільки саме в цій площині відбувається основне балансування. Отримані значення не використовуються безпосередньо для керування двигунами, оскільки вони можуть містити шуми, спричинені вібраціями корпусу, роботою двигунів і нерівностями поверхні.

Для отримання стабільного значення кута нахилу в програмі застосовується фільтрація даних. Практично доцільним є використання комплементарного фільтра, який поєднує дані акселерометра та гіроскопа. Гіроскоп забезпечує швидку реакцію на зміну положення, але з часом може накопичувати похибку. Акселерометр дозволяє оцінити нахил відносно сили тяжіння, однак є чутливим до короткочасних прискорень. Поєднання цих двох джерел даних дає змогу отримати більш надійну оцінку поточного кута нахилу.

Після обчислення поточного кута нахилу програма визначає помилку стабілізації. Вона розраховується як різниця між заданим кутом рівноваги та поточним кутом, отриманим після фільтрації даних. Заданий кут рівноваги може бути близьким до 0° , однак у реальному прототипі він може незначно відрізнятись через особливості розташування акумулятора, плати керування, сенсорів і механічних елементів. Тому це значення доцільно уточнювати експериментально під час налаштування робота.

Наступним етапом є програмна реалізація PID-регулятора. Регулятор формує керуючий сигнал на основі поточної помилки, накопиченої помилки та швидкості її зміни. У програмі це реалізується через обчислення пропорційної, інтегральної та диференціальної складових. Пропорційна складова визначає основну реакцію системи на нахил, інтегральна складова компенсує сталі

відхилення, а диференціальна складова зменшує різкі коливання та покращує стійкість робота.

Фрагмент логіки PID-регулятора можна подати як лістинг 3.1.

Лістинг 3.1 – Логіка PID-регулятора

```
error = targetAngle - currentAngle;  
integral = integral + error * dt;  
derivative = (error - previousError) / dt;  
controlSignal = Kp * error + Ki * integral + Kd * derivative;  
previousError = error;
```

кінець лістингу 3.1

Отриманий керуючий сигнал не передається на двигуни безпосередньо. Спочатку він обмежується в межах допустимого діапазону, наприклад від -255 до 255 для PWM-керування. Це потрібно для захисту двигунів, драйверів і джерела живлення від перевантаження. Крім того, обмеження сигналу дозволяє уникнути різких ривків робота під час стабілізації.

Після обмеження керуючого сигналу програма визначає напрям і швидкість обертання двигунів. Якщо сигнал має додатне значення, двигуни обертаються в одному напрямку, компенсуючи нахил уперед. Якщо сигнал має від'ємне значення, двигуни змінюють напрям обертання для компенсації нахилу назад. Абсолютне значення сигналу використовується для формування PWM - сигналу, який визначає швидкість обертання мотор-редукторів.

Узагальнений фрагмент керування двигунами можна подати як лістинг 3.2.:

Лістинг 3.2 – Фрагмент коду керування двигунами

```
if (controlSignal > 0) {  
    setMotorDirection(FORWARD);  
    setMotorSpeed(abs(controlSignal));  
}  
else if (controlSignal < 0) {  
    setMotorDirection(BACKWARD);  
    setMotorSpeed(abs(controlSignal));  
}  
else {  
    stopMotors();  
}
```

кінець лістингу 3.2

Для забезпечення стабільної роботи балансуючого робота важливо підтримувати сталий період виконання основного циклу. Якщо час між вимірюваннями буде змінюватися, це може негативно вплинути на точність обчислення кута нахилу та диференціальної складової PID-регулятора. Тому в програмі доцільно використовувати таймер або фіксовану затримку циклу, яка забезпечує регулярне оновлення даних і керуючих сигналів.

Окрему увагу під час програмної реалізації потрібно приділити обробці аварійних ситуацій. Якщо кут нахилу перевищує допустиме значення, наприклад робот уже фактично впав або втратив стійкість, доцільно тимчасово вимкнути двигуни або перевести систему в безпечний режим. Це дозволяє уникнути неконтрольованого обертання коліс і зменшити ризик пошкодження конструкції. Також бажано передбачити перевірку доступності MPU6050, оскільки втрата зв'язку з інерціальним сенсором робить стабілізацію неможливою.

У результаті програмна реалізація системи керування мікроконтролером забезпечує узгоджену роботу всіх основних елементів балансуючого робота. Мікроконтролер виконує роль центрального керуючого вузла, який отримує дані від інерціального сенсора, обробляє їх, визначає поточний стан робота та формує команди для виконавчих механізмів. Саме програмне забезпечення перетворює апаратну платформу на функціональну інтелектуальну систему, здатну підтримувати рівновагу та реагувати на зміну положення в режимі реального часу.

3.4 Реалізація навігаційного модуля та алгоритму руху робота в середовищі

Реалізація навігаційного модуля балансуючого робота спрямована на забезпечення його здатності не лише утримувати вертикальне положення, а й виконувати керований рух у середовищі з урахуванням наявності перешкод. Якщо система стабілізації відповідає за підтримання рівноваги корпусу, то навігаційний модуль доповнює її функціями просторової орієнтації, визначення

безпечного напрямку руху та формування команд для зміни траєкторії. У межах розробленого прототипу основним сенсорним елементом навігації є ультразвуковий датчик відстані HC-SR04, який використовується для виявлення об'єктів перед роботом.

Навігаційний модуль працює у взаємозв'язку із системою стабілізації. Це означає, що будь-яка команда руху не може формуватися незалежно від поточного положення корпусу робота. Передавання команди на рух уперед, зупинку або зміну напрямку повинно враховувати кут нахилу, швидкість реакції PID-регулятора та здатність мотор-редукторів компенсувати відхилення. Тому навігаційний алгоритм не замінює алгоритм стабілізації, а накладає на нього додаткові керуючі впливи, які визначають бажаний напрямок переміщення.

Основою роботи навігаційного модуля є періодичне вимірювання відстані до перешкоди. Ультразвуковий датчик HC-SR04 формує короткий ультразвуковий імпульс і приймає відбитий сигнал. Мікроконтролер вимірює час проходження імпульсу до об'єкта і назад, після чого обчислює відстань до перешкоди. Отримане значення використовується для прийняття рішення щодо подальшого руху робота. Якщо відстань є достатньою, робот може продовжувати рух. Якщо перешкода розташована занадто близько, система повинна зменшити швидкість, зупинити рух або змінити напрямок (рис. 3.17).

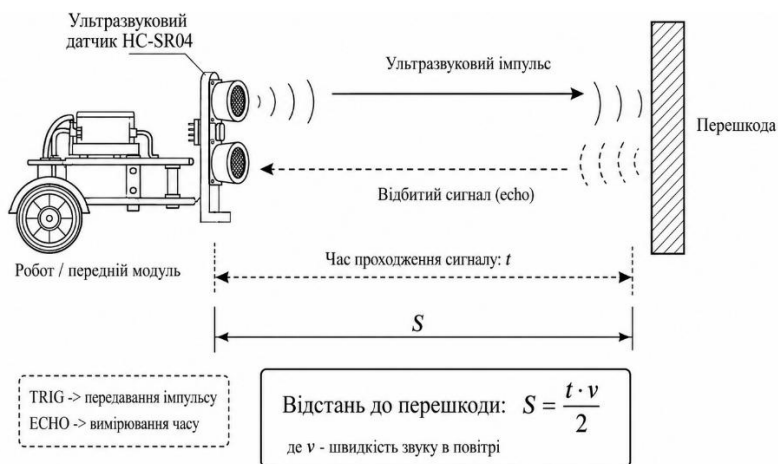


Рисунок 3.17 – Принцип визначення відстані до перешкоди за допомогою ультразвукового датчика HC-SR04

У програмній реалізації для наближених розрахунків швидкість звуку може прийматися як 343 м/с за нормальних умов. Однак для навчального прототипу абсолютна точність вимірювання не є визначальною, оскільки головним завданням є виявлення факту наближення до перешкоди та своєчасна реакція системи керування.

Алгоритм руху робота в середовищі побудований за принципом порогового аналізу відстані (рис. 3.18). Для цього в програмі задається мінімально допустима відстань до перешкоди, наприклад 20-30 см. Якщо виміряна відстань перевищує встановлений поріг, робот продовжує рух уперед. Якщо відстань стає меншою за порогове значення, мікроконтролер формує команду зупинки або зміни напрямку. Такий підхід є простим, але достатньо ефективним для базового прототипу балансуєчого робота, який має обмежену кількість сенсорів.

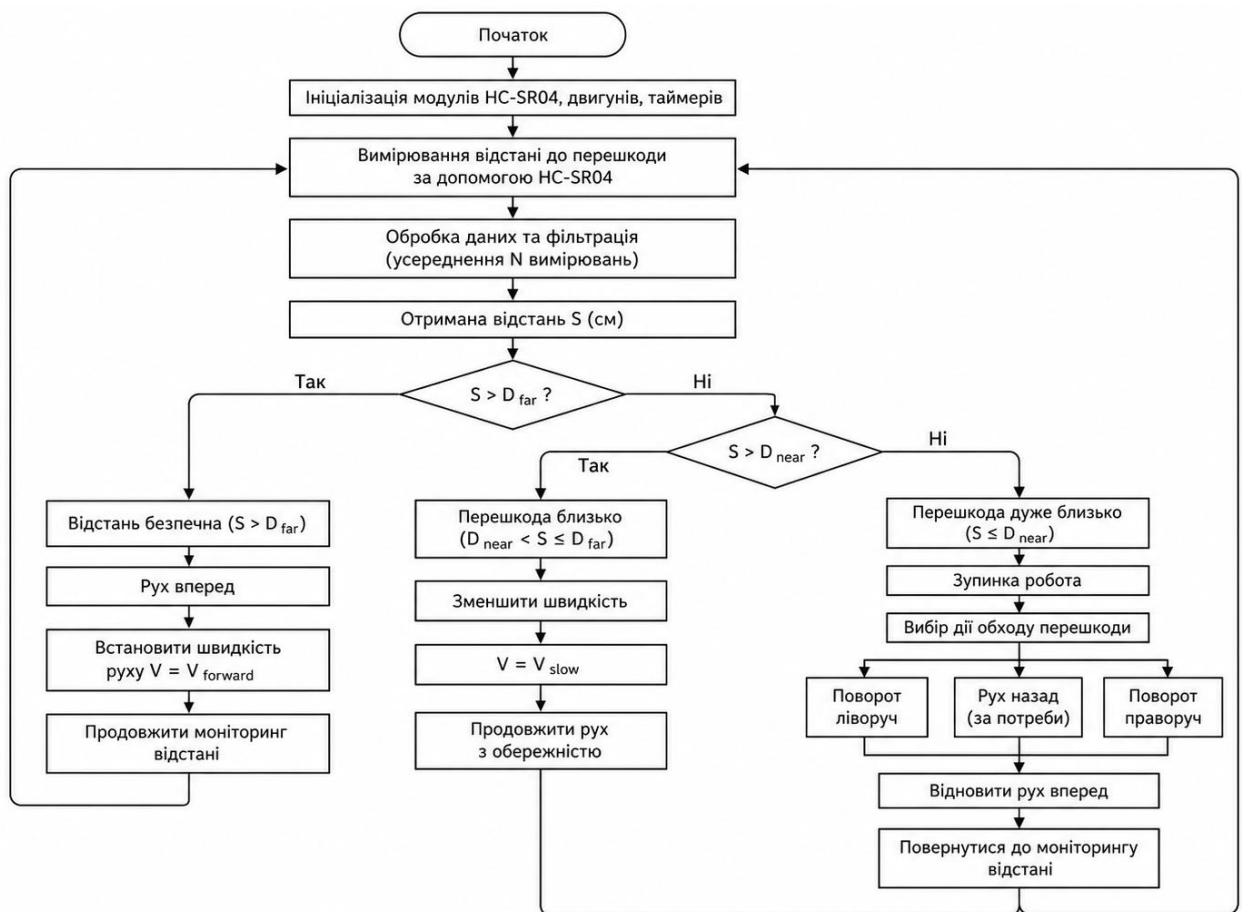


Рисунок 3.18 – Блок-схема алгоритму прийняття рішення щодо руху робота залежно від відстані до перешкоди

У загальному вигляді логіка руху робота може бути описана так: спочатку система стабілізує корпус у вертикальному положенні, після чого навігаційний модуль зчитує відстань до перешкоди. Якщо простір перед роботом вільний, формується команда руху вперед. Якщо датчик виявляє перешкоду, робот зменшує швидкість і переходить у режим зупинки. Після цього можливе виконання маневру зміни напрямку, наприклад короткий рух назад або поворот за рахунок різної швидкості обертання лівого та правого коліс.

Для реалізації повороту використовується принцип диференціального керування колесами. Якщо ліве і праве колесо обертаються з однаковою швидкістю в одному напрямку, робот рухається прямо. Якщо швидкість одного колеса зменшується або змінюється напрям його обертання, робот починає повертати. У балансуєчому роботі такий маневр потрібно виконувати обережно, оскільки різка зміна швидкості одного з двигунів може порушити рівновагу. Тому команди повороту повинні формуватися плавно, з обмеженням максимального керуючого впливу.

Програмна реалізація навігаційного алгоритму передбачає поєднання двох контурів керування: контуру стабілізації та контуру навігації (рис. 3.19).

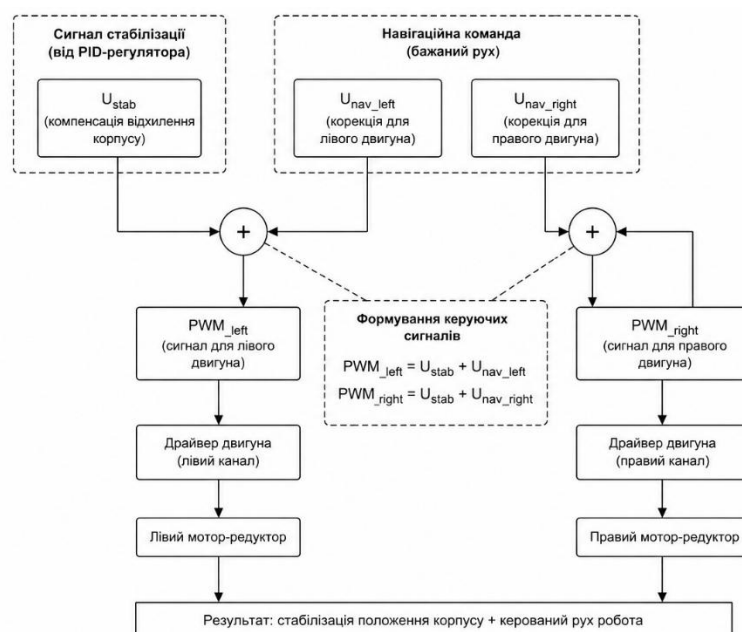


Рисунок 3.19 – Поєднання сигналу стабілізації та навігаційної команди для керування двигунами

Особливу увагу під час реалізації навігаційного алгоритму потрібно приділити обмеженню швидкості руху. Балансуючий робот не повинен різко прискорюватися або зупинятися, оскільки це може спричинити втрату рівноваги. Тому в програмі доцільно використовувати плавну зміну навігаційної команди. Наприклад, замість миттєвого встановлення максимальної швидкості система може поступово збільшувати значення PWM-сигналу до заданого рівня. Аналогічно, під час зупинки швидкість повинна зменшуватися поступово.

У процесі реалізації навігаційного модуля також враховується зона невизначеності вимірювань ультразвукового датчика. Якщо перешкода має складну форму, нахилену поверхню або погано відбиває ультразвук, показники можуть бути нестабільними. Для зменшення впливу випадкових похибок доцільно виконувати кілька послідовних вимірювань і використовувати середнє значення або відкидати явно некоректні результати. Це підвищує надійність прийняття рішень і зменшує ймовірність помилкової зупинки робота.

Загальний алгоритм роботи навігаційного модуля можна описати так. Після запуску системи виконується ініціалізація ультразвукового датчика та задання порогових значень відстані. Далі в основному циклі програми мікроконтролер зчитує відстань до перешкоди, порівнює її з допустимим порогом і визначає режим руху. Якщо перешкоди немає, робот рухається вперед із заданою швидкістю. Якщо перешкоду виявлено, робот зменшує швидкість, зупиняється або виконує маневр обходу. Після цього цикл повторюється.

Для оцінювання роботи навігаційного модуля доцільно використовувати такі критерії: правильність вимірювання відстані, своєчасність реакції на перешкоду, плавність зміни швидкості, відсутність різких коливань корпусу під час маневру та здатність системи повернутися до стабільного руху після зупинки або повороту. Основні параметри роботи навігаційного модуля наведено в таблиці 3.1.

Таблиця 3.1 – Основні параметри роботи навігаційного модуля балансуєчого робота

Параметр	Призначення	Орієнтовне значення
Мінімальна допустима відстань до перешкоди	Визначає момент зупинки або зміни напрямку	20-30 см
Період вимірювання відстані	Визначає частоту оновлення даних з HC-SR04	50-100 мс
Максимальна навігаційна поправка	Обмежує вплив навігаційної команди на двигуни	10-30 % від PWM
Кількість вимірювань для усереднення	Зменшує вплив шуму сенсора	3-5 вимірювань
Час затримки перед маневром	Запобігає різким змінам руху	100-300 мс

Реалізація навігаційного модуля забезпечує балансуєчому роботу можливість орієнтуватися в середовищі та реагувати на перешкоди. Ультразвуковий датчик HC-SR04 виконує роль основного засобу визначення відстані, а мікроконтролер аналізує отримані дані та формує відповідні команди руху. Поєднання навігаційної логіки з алгоритмом стабілізації дає змогу забезпечити не лише підтримання рівноваги, а й кероване переміщення робота в межах простого середовища.

ВИСНОВКИ

У кваліфікаційній роботі було розроблено інтелектуальну систему стабілізації та навігації балансуєчого робота на базі мікроконтролера. У процесі виконання роботи було послідовно розглянуто теоретичні, апаратні, програмні та експериментальні аспекти створення такої робототехнічної системи.

У роботі проаналізовано теоретичні основи побудови балансуєчих роботів та принципи їхньої динамічної рівноваги. Встановлено, що балансуєчий робот є нестійкою мобільною платформою, яка потребує постійного контролю кута нахилу та оперативного коригування положення за допомогою електродвигунів. Розглянуто модель робота як зворотного маятника, що дало змогу пояснити особливості його поведінки під час руху та утримання вертикального положення.

У межах дослідження було охарактеризовано методи стабілізації мобільних роботизованих систем. Основну увагу приділено PID-регулюванню, яке дозволяє враховувати поточне відхилення, швидкість його зміни та накопичену похибку. Такий підхід є доцільним для балансуєчого робота, оскільки забезпечує швидке формування керувального впливу на мотор-редуктори та сприяє підтриманню рівноваги в режимі реального часу.

Також було розглянуто сенсорні засоби, які використовуються для визначення кута нахилу та просторової орієнтації робота. Зокрема, обґрунтовано використання інерціального модуля MPU6050, який поєднує акселерометр і гіроскоп. Дані з цього сенсора використовуються для визначення поточного положення корпусу, обчислення кута нахилу та подальшого формування сигналів стабілізації.

У роботі обґрунтовано вибір основних апаратних компонентів прототипу: мікроконтролерної плати, інерціального сенсора, мотор-редукторів, драйвера двигунів, джерела живлення та ультразвукового датчика HC-SR04. Обрані компоненти забезпечують можливість реалізації стабілізації положення, керування рухом і базового виявлення перешкод у локальному середовищі.

Було розроблено структурну схему інтелектуальної системи стабілізації та навігації балансуємого робота. У ній поєднано сенсорну підсистему, мікроконтролерний блок обробки даних, виконавчі механізми, підсистему живлення та навігаційний модуль. Така структура відображає взаємозв'язок між апаратними компонентами та програмними алгоритмами керування.

Описано алгоритм обробки сенсорних даних і формування керувальних сигналів для двигунів. Він передбачає зчитування даних з MPU6050, фільтрацію вимірювань, обчислення кута нахилу, визначення помилки стабілізації та формування PWM-сигналів для мотор-редукторів. Це дозволяє системі оперативно реагувати на зміну положення корпусу та компенсувати відхилення від вертикального стану.

Програмна реалізація системи керування виконана за циклічним принципом. Програма забезпечує ініціалізацію модулів, калібрування інерціального сенсора, обробку даних, роботу PID-регулятора, керування двигунами та аналіз відстані до перешкод. Навігаційна складова реалізована на основі ультразвукового датчика HC-SR04, що дає змогу визначати перешкоди перед роботом і змінювати режим руху.

Проведене тестування підтвердило працездатність створеного прототипу. Було перевірено роботу інерціального сенсора, правильність визначення кута нахилу, реакцію двигунів на керувальні сигнали, ефективність PID-регулювання та здатність навігаційного модуля виявляти перешкоди. Отримані результати засвідчили, що розроблена система може забезпечувати стабілізацію балансуємого робота та виконання базових навігаційних дій.

Створений прототип балансуємого робота може використовуватися як навчально-дослідна платформа для вивчення мікроконтролерних систем, алгоритмів стабілізації, PID-регулювання, сенсорної обробки даних і автономної навігації мобільних роботів. Подальше вдосконалення системи може бути пов'язане з точнішим налаштуванням регулятора, покращенням фільтрації даних, використанням додаткових сенсорів і розширенням можливостей автономного руху.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Control Strategies for Two-Wheeled Self-Balancing Robotic Systems: A Comprehensive Review. *Robotics*. URL: <https://doi.org/10.3390/robotics14080101> (date of access: 15.01.2026).
2. Fractional-Order PID Control of Two-Wheeled Self-Balancing Robots with Optimisation-Based Parameter Tuning. *Mdpi*. URL: <https://www.mdpi.com/2504-3110/9/10/619> (date of access: 20.01.2026).
3. Abdelgawad A., Shohdy T., Nada A. Model- and Data-Based Control of Self-Balancing Robots. *IFAC-PapersOnLine*. URL: <https://doi.org/10.1016/j.ifacol.2024.07.399> (date of access: 20.01/2026),
4. Adaptive Nonlinear PD Controller of Two-Wheeled Self-Balancing Robot. *Engineering Science and Technology, an International Journal*. 2024. URL: <https://www.sciencedirect.com/science/article/pii/S1546221824007859> (date of access: 20.01.2026).
5. Clonal selection algorithm based control for two-wheeled self-balancing mobile robot. *Engineering Applications of Artificial Intelligence*. URL: <https://doi.org/10.1016/j.engappai.2022.104940> (date of access: 20.01.2026).
6. Design and Implementation of a Two-Wheeled Self-Balancing Car Based on Dual Closed-Loop PID Control and Filtering Algorithms. *Applied Sciences*. URL: <https://www.mdpi.com/2076-3417/14/12/5296> (date of access: 20.01.2026).
7. Development of Proportional-Integral-Derivative Based Self-Balancing Robot. *Engineering Proceedings*. URL: <https://www.mdpi.com/2673-4591/92/1/24> (date of access: 20.01.2026).
8. Mobile Robot with Bluetooth Control: Design, Implementation, and Performance Analysis. *Robotics*. URL: <https://www.mdpi.com/2673-4052/6/3/42> (date of access: 10.02.2026).
9. Unified Control Framework for Self-Balancing Robots. *Sensors*. URL: <https://www.mdpi.com/1424-8220/25/23/7144> (date of access: 10.02.2026).

10. Modified Dual Hierarchical Terminal Sliding Mode Control for a Two-Wheeled Self-Balancing Robot. *Electronics*. URL: <https://www.mdpi.com/2079-9292/14/13/2692> (date of access: 10.02.2026).

11. Balance Control Method for Bipedal Wheel-Legged Robots Based on Friction Feedforward Linear Quadratic Regulator. *Sensors*. URL: <https://doi.org/10.3390/s25041056> (date of access: 10.02.2026).

12. JuMBRo: A model predictive control based two-wheeled self-balancing robot capable of jumping over obstacles. *Control Engineering Practice*. URL: <https://doi.org/10.1016/j.conengprac.2025.106620> (date of access: 10.02.2026).

13. Papadimitriou K. D., Tzes A., Kanoulas D. Genetic Algorithm-based Control of a Two-Wheeled Self-Balancing Mobile Robot. *Journal of Intelligent & Robotic Systems*. URL: <https://doi.org/10.1007/s10846-025-02236-1> (date of access: 20.02.2026).

14. Classic and Fuzzy Parallel Hybrid Controller of PD-PI Type for a Two-Wheeled Self-Balancing Robot. *Robotics*. URL: <https://www.mdpi.com/2673-4052/7/2/49> ((date of access: 20.02.2026)).

15. Zhang J., Zhao T., Guo B., Dian S. Fuzzy fractional-order PID control for two-wheeled self-balancing robots on inclined road surface. *Systems Science & Control Engineering*. 2022. Vol. 10. P. 289-299.

16. Gad O. M. M., Saleh S. Z. M., Bulbul M. A., Khadraoui S. Design and control of two wheeled self balancing robot. *Proceedings of the 2022 Advances in Science and Engineering Technology International Conferences (ASET)*. Dubai, United Arab Emirates, 2022. P. 1-6.

17. Chen J.-X., He N., He L. Adaptive double fuzzy anti-integral saturation PID control for self-balancing robot with reaction wheel. *Proceedings of the 2022 IEEE 17th International Conference on Control & Automation (ICCA)*. Naples, Italy, 2022. P. 56-61.

18. A deep reinforcement learning algorithm to control a two-wheeled scooter. *Engineering Applications of Artificial Intelligence*. 2023. URL:

<https://www.sciencedirect.com/science/article/abs/pii/S0952197623011259> (date of access: 20.03.2026).

19. Research on Stability Control System of Two-Wheel Heavy-Duty Self-Balancing Vehicle. *Applied Sciences* URL: <https://www.mdpi.com/2076-3417/14/17/7682> (date of access: 20.03.2026).

20. Mechatronic Design Procedure for Self-Balancing Vehicles. *Machines*. URL: <https://www.mdpi.com/2075-1702/13/9/826> (date of access: 20.03.2026).

21. Localized Path Planning for Mobile Robots Based on a Subarea-Artificial Potential Field Model. *Sensors*. URL: <https://doi.org/10.3390/s24113604> (date of access: 20.03.2026).

22. Path Planning Method and Control of Mobile Robot with Uncertain Dynamics Based on Improved Artificial Potential Field and Its Application in Health Monitoring. *Mathematics*. URL: <https://doi.org/10.3390/math12192965> (date of access: 20.03.2026).

23. Tao S. Improved artificial potential field method for mobile robot path planning. *Applied and Computational Engineering*. 2024. Vol. 33, No. 1. P. 157-166. DOI: <https://doi.org/10.54254/2755-2721/33/20230259>.

24. Improved artificial potential field method based on robot obstacle avoidance path planning. *International Journal of Advanced Robotic Systems*. URL: <https://doi.org/10.1177/17298806241278172> (date of access: 20.03.2026).

25. An improved dynamic window approach algorithm for dynamic obstacle avoidance in mobile robot formation. *Decision Analytics Journal*. URL: <https://doi.org/10.1016/j.dajour.2024.100471> (date of access: 20.03.2026).

26. Enhancing Obstacle Avoidance in Dynamic Window Approach for Mobile Robots. *Actuators*. URL: <https://www.mdpi.com/2076-0825/14/5/207> (date of access: 20.03.2026)

27. Design and Implementation of an Ultrasonic Sensor-Based Obstacle Avoidance System for Arduino Robots. *Proceedings of the 2023 International Conference on Information and Communication Technology for Sustainable*

Development. URL: <https://doi.org/10.1109/ICICT4SD59951.2023.10303550> (date of access: 20.03.2026).

28. Arduino Nano. *Arduino Documentation*. URL: <https://docs.arduino.cc/hardware/nano> (date of access: 20.03.2026).

29. Arduino Nano Datasheet. *Arduino Documentation*. URL: <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf> (date of access: 20.03.2026).

30. ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash Datasheet. *Microchip Technology Inc.* URL: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (date of access: 20.03.2026).

31. MPU-6050: IMU Inertial Measurement Unit. *TDK InvenSense Product Center*. URL: https://product.tdk.com/en/search/sensor/motion-inertial/imu/info?part_no=MPU-6050 (date of access: 20.04.2026).

32. MPU-6000 and MPU-6050 Register Map and Descriptions. *InvenSense*. URL: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf> (date of access: 20.04.2026).

33. TB6612FNG Dual Motor Driver Carrier. *Pololu*. URL: