

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроенерготехніки та автоматики

(повне найменування інституту, факультету)

Кафедра автоматизації електромеханічних систем та електроприводу

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ Сергій ПЕРЕСАДА
(підпис)

“ ____ ” _____ 20__ р.

Дипломний проект

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Електромеханічні системи
автоматизації, електропривод та електромобільність»
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»
(код і назва)

на тему: Система керування рухом транспортного засобу _____

Виконав (-ла): студент (-ка) IV курсу, групи ЕП-71
(шифр групи)

Чинкин Ярослав Олегович
(прізвище, ім'я, по батькові)

Керівник старший викладач, к.т.н. Димко Сергій Сергійович
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по-батькові)

Консультант _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ім'я, по-батькові)

Рецензент доцент, Гераскін Олександр Анатолійович
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по-батькові)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2021 р.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЕКТУ

[illegible]

				141.7117.029.БР		
	ПБ	Підп.	Дата			
Розробн.	Чинкин Я.О.			Відомість дипломного проекту	Лист	Листів
Керівн.	Димко С.С.				2	74
Консульт.					КПІ ім. Ігоря Сікорського	
Н/контр.	Теряєв В.І.				Каф. АЕМС-ЕП	
Зав.каф.	Пересада С.М.				Гр. ЕП - 71	

Пояснювальна записка до дипломного проекту

на тему: Система керування рухом транспортного засобу

Київ – 2021 р.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет електроенерготехніки та автоматики
(повна назва)

Кафедра автоматизації електромеханічних систем та електроприводу
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 141 Електроенергетика, електротехніка та електромеханіка
(код і назва)

Освітньо-професійна програма – Електромеханічні системи автоматизації, електропривод та електромобільність

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій ПЕРЕСАДА
(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ
на дипломний проект студенту
Чинкин Ярослав Олегович
(прізвище, ім'я, по батькові)

1. Тема проекту Система керування рухом транспортного засобу _____

керівник проекту старший викладач, к.т.н. Димко Сергій Сергійович _____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Термін подання студентом проекту 10.06.2021р.

3. Вихідні дані до проекту Точність позиціонування ± 5 см, рух до точки, рух по лінії.

4. Зміст пояснювальної записки Аналітичний огляд системи керування транспортним засобом.; Кінематика транспортного засобу; Алгоритми керування транспортним засобом; Дослідження алгоритмів керування

транспортним засобом; Практична реалізація алгоритмів керування через Robot Operating System (ROS).

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) Результати досліджень(A1), Результати експерименту (A1), Кінематичні моделі транспортного засобу

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв


* Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломного проекту.

7. Дата видачі завдання «5» квітня 2021 р.

Календарний план

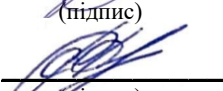
№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Аналітичний огляд системи керування транспортним засобом (Особливості поведінки автомобіля при поворотах; геометрія Аккермана; кінематична модель велосипеда).	22.04.21	
2	Кінематика транспортного засобу; кінематичне моделювання за принципом Аккермана; приведення до кінематичної моделі велосипеда.	02.05.21	
3	Огляд алгоритмів керування транспортним засобом, таких як алгоритм руху до точки та слідування лінії.	13.05.21	
4	Дослідження алгоритмів руху до точки та слідування лінії шляхом математичного моделювання в середовищі MatLab	23.05.21	
5	Розгляд програмного пакету Robot Operating System (ROS) та проведення експериментальних досліджень з його допомогою на реальному прототипі транспортного засобу.	31.05.21	
	Оформлення пояснювальної записки та графічної частини. Підготовка презентації	05.06.21	

Студент


(підпис)

Ярослав ЧИНКИН

Керівник проекту


(підпис)

Сергій ДИМКО

РЕФЕРАТ

Бакалаврська робота складається із вступу, п'яти розділів, висновка, списку використаної літератури і додатка (74 сторінки пояснювальної записки, 60 рисунків, 13 використаних джерел).

Мета роботи – дослідження систем керування рухом транспортного засобу.

Об'єкт дослідження – системи керування рухом транспортних засобів.

Предмет дослідження – алгоритми керування рухом та позиціонування транспортного засобу.

В першому розділі було проведено аналітичний огляд систем керування транспортним засобом, розглянуто особливості поведінки авто при поворотах, розглянуто перехід до моделі Аккермана, виконано огляд кінематичної моделі велосипеда.





В другому розділі розглянуто кінематичну модель чотириколісного транспортного засобу та приведення її до моделі велосипеда.

В третьому розділі виконано огляд алгоритмів керування транспортним засобом таких як рух до заданої точки та слідування лінії.

В четвертому розділі було проведено дослідження алгоритмів керування, розглянутих у попередньому пункті, шляхом математичного моделювання.

В п'ятому розділі було розглянуто особливості роботи та налаштування Robot Operating System (ROS), для подальшого використання при практичній реалізації алгоритмів керування транспортним засобом. Виконано дослідження алгоритмів керування на реальному прототипі транспортного засобу експериментальним шляхом.

МОДЕЛЬ АККЕРМАНА, АЛГОРИТМИ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОБОМ, КІНЕМАТИЧНА МОДЕЛЬ ВЕЛОСИПЕДА, КОЛІСНА БАЗА, КУТ ПОВОРОТУ, МОБІЛЬНИЙ РОБОТ, ПОЗИЦІОНУВАННЯ

					141.7117.029.БР				
					Система керування рухом транспортного засобу	Літ		Маса	Масштаб
							Т		
Зм.	Лист	№ докум.	Підпис	Дата		Лист 6		Листів 74	
Розроб.		Чинкин Я.О.				КПП ім. Ігоря Сікорського Каф. АЕМС-ЕП Гр. ЕП -71			
Перевір.		Димко С.С							
Г.контр.									
Реценз.		Гераскін О.А.							
Н. контр.		Геряєв В.І.							
Ватверд.		Пересада С.М.							

SUMMARY

The bachelor's thesis consists of an introduction, five chapters, a conclusion, a list of references and an appendix (74 pages of explanatory note, 60 figures, 13 sources used).

The purpose of the work is to examine the vehicle steering control system.

The object of research is steering control system.

The subject of research is the algorithm for steering and positioning of the vehicle.

In the first section there is an analytical review of vehicle steering control system, transition to the Ackermann model, kinematic bicycle model.


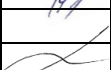
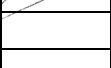
The second section considers the kinematic model of a four-wheeled vehicle and bringing it to the model of a bicycle.

The third section provides an overview of vehicle control algorithms such as moving to a given point and following a line.

In the fourth section, a study of control algorithms discussed in the previous paragraph was carried out by mathematical modeling.

The fifth section discusses the features of the operation and configuration of the Robot Operating System (ROS), for further use in the practical implementation of vehicle control algorithms. The research of control algorithms on the real prototype of the vehicle is performed experimentally

AKKERMANN MODEL, VEHICLE CONTROL ALGORITHMS, KINEMATIC BICYCLE MODEL, WHEEL BASE, TURNING ANGLE, MOBILE ROBOT, POSITIONING

					<div>141.7117.029.БР</div>						
					Vehicle steering control system			Let		Mass	Scale
Ch.	Sheet	№ докум.	Підпис	Дата					T		
Develop		Чинкин Я.О.									
Check		Димко С.С.									
G.contr								Sh.7		Sheets 74	
Review		Гераскін О.А.						Igor Sikorsky KPI, FEA, AEMS-ED, EP-71			
N.contr		Геряев В.І.									
Approv.		Пересада С.М.									

ЗМІСТ

ВСТУП.....	10
1 АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМИ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОБОМ.....	11
1.1 Особливості поведінки автомобіля при поворотах	11
1.2 Перехід до геометрії Аккермана.....	13
1.3 Кінематична модель велосипеда для чотириколісного транспортного засобу	18
1.4 Розгляд кінематичної моделі велосипеда	18
Висновки до розділу 1	21
2 КІНЕМАТИКА ТРАНСПОРТНОГО ЗАСОБУ	22
2.1 Припущення щодо задачі кінематики транспортного засобу.....	22
2.2 Система координат.....	22
2.3 Кінематичний опис	24
2.4 Кінематичне моделювання за принципом Аккермана	25
2.5 Кінематична модель, що враховує вплив похибки	26
2.6 Приведення до кінематичної моделі велосипеда	27
2.7 Інші варіанти реалізації кінематичної моделі	30
2.8 Обмеження кінематичної моделі велосипеда.....	31
Висновки до розділу 2	33
3 АЛГОРИТМИ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОБОМ.....	34
3.1 Рух до точки призначення.	34
3.1.1 Формулювання задачі.....	34
3.1.2 Модель машини.....	34
3.1.3 Геометрія руху до точки та кінематичні рівняння	36
3.1.4 Кінематика руху моделі.....	37
3.2 Слідування по лінії.....	37
3.2.1 Формулювання задачі	37
3.2.2 Кінематика руху	38
3.2.3 Опис алгоритму.....	38
Висновки до розділу 3	39
4 ДОСЛІДЖЕННЯ АЛГОРИТМІВ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОБОМ.....	40
4.1 Алгоритм руху до заданої точки	41

4.2 Алгоритм слідування заданої лінії з заданої позиції	51
Висновки до розділу 4	59
5 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ КЕРУВАННЯ ЧЕРЕЗ ROBOT OPERATING SYSTEM (ROS).....	60
5.1 Що таке ROS?	60
5.2 Переваги використання ROS.....	60
5.3 Основні поняття	61
5.4 Інструменти.....	62
5.5 Вибір дистрибутиву	63
5.6 Інсталяція	63
5.7 Налаштування.....	64
5.8 Запуск програми в ROS	64
5.9 Створення власної програми в ROS2	65
5.10 З MatLab до ROS	65
5.11 Проведення експерименту на реальному прототипі	66
Висновки до розділу 5	71
ВИСНОВКИ.....	72
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	73

ВСТУП

З розвитком технологій люди все частіше стараються автоматизувати певні процеси праці та життєдіяльності. Одним з таких процесів є водіння різноманітного транспорту, що, в свою чергу, виступає доволі небезпечним та відповідальним заняттям.

Для цього при проектуванні транспортних засобів іноді встановлюють різного роду системи керування руху транспортним засобом, які встановлюються від агропромислового транспорту до транспорту цивільного.

Фактичне впровадження систем керування рульовим механізмом є однією з дисциплін у галузі інтелектуальних автомобілів, якій на даний момент часу приділяється більше уваги зі сторони таких компаній як Tesla.

Основною метою використання систем управління рульовим керуванням є підвищення безпеки та зменшення як нещасних випадків, так і робочого навантаження водіїв.

1 АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМИ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОБОМ

1.1 Особливості поведінки автомобіля при поворотах

Поворот – це є зміна напрямку руху транспортного засобу відмінного від прямого.

Тобто, коли передні колеса знаходяться в прямому положенні, то, відповідно, автомобіль рухається по прямій лінії, а коли вони почнуть повертатись на будь-який кут, тоді автомобіль почне виконувати поворот. Радіус повороту автомобіля обернено пропорційно залежить від величини кута повороту передніх його коліс. Тобто чим більше кут – тим менше радіус.

Так як поворот коліс виконується за допомогою механізму рульового управління, то прийнято говорити не про кути повороту передніх коліс, а про поворот керма. Навіть незначний поворот керма відведе автомобіль з прямолінійною траєкторії.

Якщо повернути кермо, зафіксувати його в одному положенні і приступити до руху, то автомобіль опише коло. Центр кола, уздовж якого повертає автомобіль, завжди лежить на продовженні задньої осі автомобіля, а радіус повороту автомобіля змінюється кутом повороту передніх коліс.

Машина залишається керованою, коли колеса мають контакт з дорогою достатній для того, щоб колеса не ковзали. Зі збільшенням швидкості цей контакт слабшає, і в залежності від якості або стану дорожнього покриття може зникнути зовсім. Ця властивість помітно проявляється на слизькій поверхні дороги.

Автомобіль спирається на дорогу чотирма колесами. Система підвіски допомагає долати нерівності дороги і забезпечує водієві та пасажирам певний комфорт, але зчеплення з дорожнім покриттям автомобілю забезпечують тільки чотири зони контакту його коліс.

Коли автомобіль стоїть на місці, вага автомобіля рівномірно розподілена по всіх його колесах.

При початку руху та набору швидкості вага автомобіля зміщується назад. А при уповільненні і зупинці, вага переміщується вперед.

Коли автомобіль рухається прямолінійно при постійній тязі, його вага відносно рівномірно розподіляється по всіх колесах. І якщо полотно дороги рівне та наявне гарне зчеплення коліс з полотном, то автомобіль на дорозі стійкий та добре керований.

Оскільки будь-який автомобіль має вагу і має інерцію, то коли автомобіль починає виконувати поворот, то центр маси автомобіля прагне за межі кола повороту з силою, яку дає йому прискорення. Вектор сили «F» спрямований перпендикулярно радіусу повороту (рис.1.1).

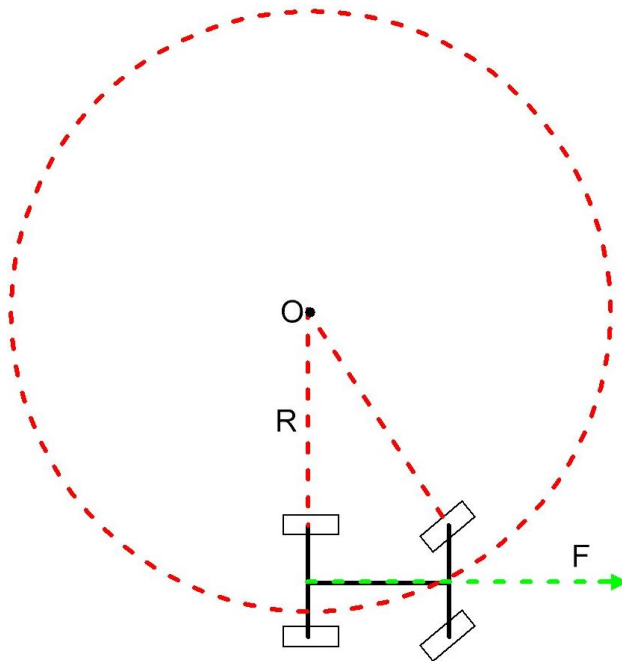


Рисунок 1.1 – Зображення вектору сили при повороті авто

Вірогідність вийти за межі дуги повороту тим сильніше, чим менше радіус повороту та чим вище швидкість автомобіля в повороті. Саме з цієї причини швидкість необхідно знизити ще до входу в поворот, а безпосередньо дугу повороту необхідно пройти на постійній швидкості.

Наслідки переміщення ваги в процесі повороту автомобіля помітно впливають на якість зчеплення коліс з дорожнім покриттям. Тому як в процесі руху автомобіля в повороті відбувається перерозподіл його ваги на кожному з його чотирьох коліс.

Тобто при повороті автомобіля зчеплення з дорогою у всіх коліс не є рівномірним. Тоді найбільш навантаженим колесом стає зовнішнє переднє колесо відносно дуги повороту (на правому повороті цим колесом стає переднє ліве, а на лівому повороті - переднє праве).

Відповідно ці колеса на повороті матимуть найсильніше зчеплення з дорожнім полотном, тоді як у решти коліс зчеплення з дорогою на поворотах слабшає. Це відбувається безперервно з кожним поворотом керма.

Автомобіль залишається керованим та слухається повороту керма тільки завдяки гарному зчепленню коліс з дорожнім покриттям. Хороше зчеплення - це коли колеса автомобіля не ковзають в незалежності від умов дорожнього покриття.

Вирішальним фактором залишається швидкість. Якщо автомобіль починає втрачати зчеплення з дорогою, то це означає, його швидкість в даний момент є занадто високою.

Різниця між особливостями руху на літній і зимовій дорозі полягає лише в тому, що на зимовій слизькій дорозі автомобіль втратить зчеплення з дорогою на значно меншій швидкості, ніж на літній сухій дорозі при протилежних умовах руху.

На одному і тому ж дорожньому полотні під час руху автомобіля на повороті його колеса заковзають швидше, ніж в процесі прямолінійного руху. Чим вище швидкість на дугі повороту, тим вище ймовірність зриву коліс з поверхні дорожнього покриття.

Перерозподіл ваги по колесах автомобіля відбувається при кожному повороті керма, в тому числі і при коригуванні руху на прямій дорозі.

1.2 Перехід до геометрії Аккермана

Розглянемо історичний перехід до використання геометрії Аккермана в поворотних механізмах чотириколісних транспортних засобів.

Для початку розглянеться один із перших поворотних механізмів, який колись використовувався ще в каретах, де для того щоб виконати поворот, потрібно було повертати всю вісь (рис.1.2).

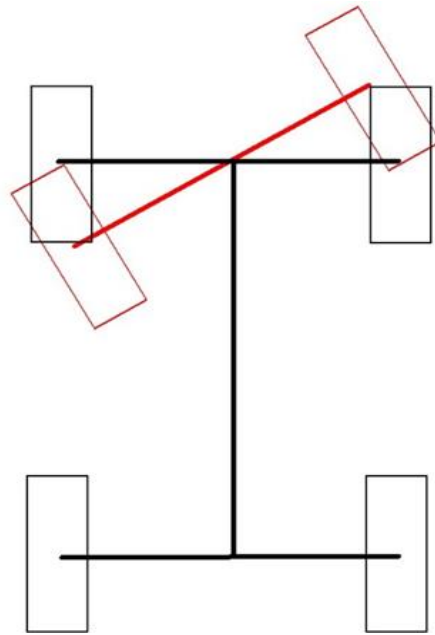


Рисунок 1.2 – Наглядне зображення автомобіля з поворотною передньою віссю

Від осей коліс проведено перпендикуляри, які, в свою чергу, при пересіченні утворюють точку «О», навколо якої авто виконуватиме поворот. Тобто створено центр обертання. Якщо кут повороту осі зменшити, то, як видно на рис.1.3, відстань до точки, навколо якої виконується поворот «О1» змінить своє положення та здвинеться далі.

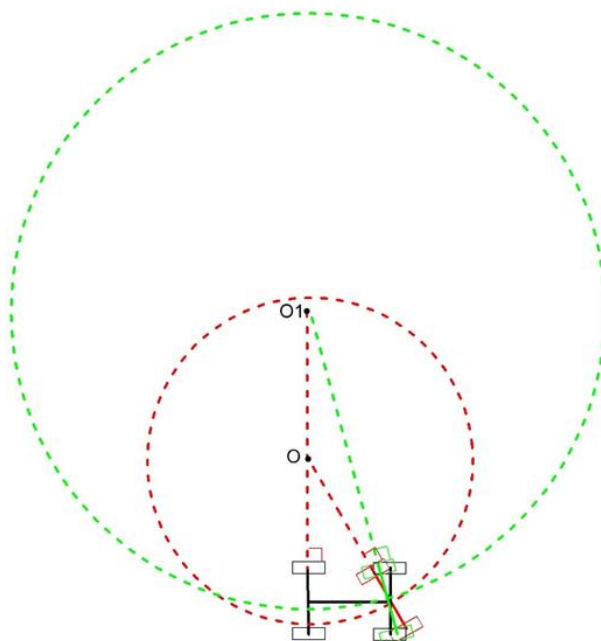


Рисунок 1.3 – Зображення зміни радіусу повороту при зміні кута повороту передньої осі

Варто наголосити, що виконати поворот, повертаючи всю вісь, не є досить ефективним рішенням з механічної точки зору. Так як, через специфіку конструкції, колеса умовної карети повинні бути досить маленькими, щоб вісь могла нормально прокручуватися під корпусом транспортного засобу. З цієї причини передня частина карети підстрибувала над різними перешкодами на шляху, що створювало постійні удари які завдавали шкоди як коням, так і власне кареті. Більші передні колеса були б кращими, але це не є гарним рішенням через специфіку конструкції.

Другою вадою була небезпека перекидання. У звичайній кареті з обертовою передньою віссю основа, на якій спирається карета, змінюється з прямокутника на трикутник, якщо поворот серйозний. Це добре видно на схемі, зображеній нижче (рис.1.4).

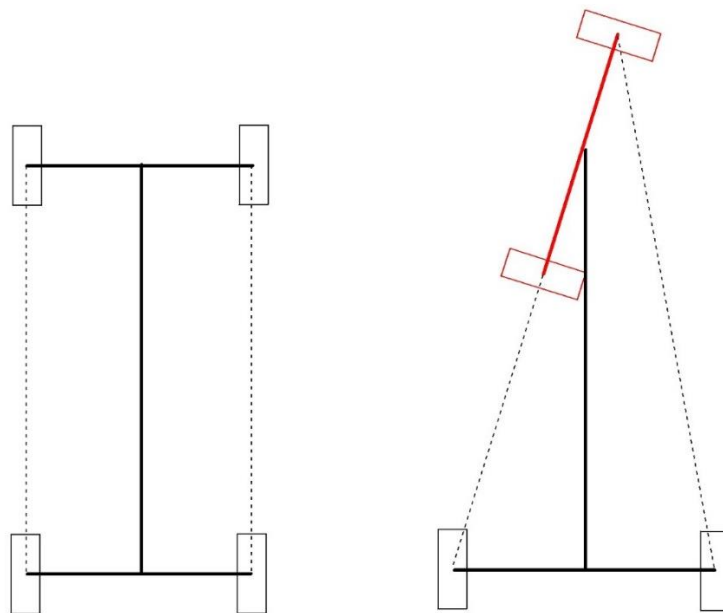


Рисунок 1.4 – Наглядне зображення зміни форми основи транспортного засобу при повороті передньої осі, що може призвести до його перекидання

Отже використання такого роду поворотного механізму є не дуже вигідним та доволі небезпечним. В такому випадку більш правильним рішенням поставленої задачі буде повертати кожне колесо окремо.

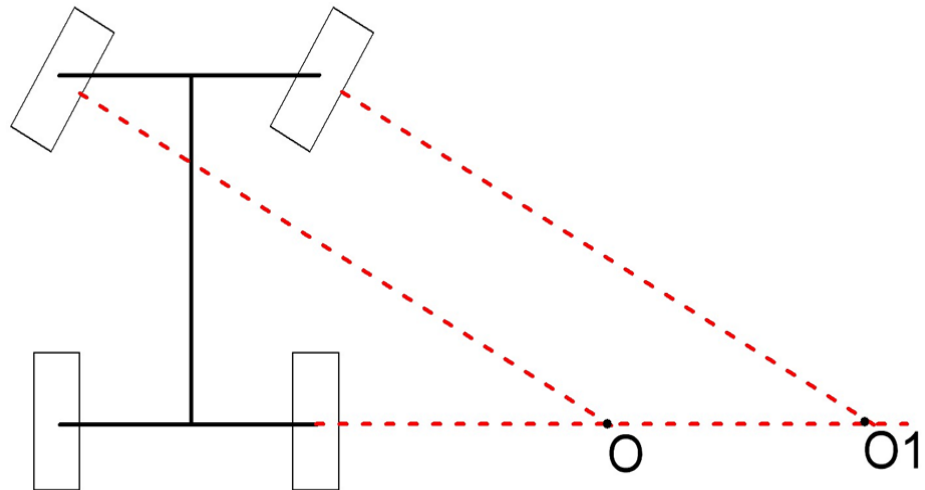


Рисунок 1.5 – Схема автомобіля, де кожне колесо повертається окремо

Припущено, що обидва колеса повернуті на однаковий кут. Такий підхід був би простіше, але при проекції перпендикулярів, отримано два центри обертання для кожного колеса окремо. Тобто при повороті принаймні одне колесо, як не обидва, почне ковзати, що негативно вплине на процес зносу шин. Тому цього явища варто уникати.

В такому випадку в якості вирішення проблеми потрібно повертати обидва колеса так, щоб у них був спільний центр обертання, але при цьому не повертати всю спільну вісь. Рішенням виступає поворот кожного колеса на свій кут таким чином, щоб їх перпендикуляри при пересіченні сходились в один єдиний центр обертання.

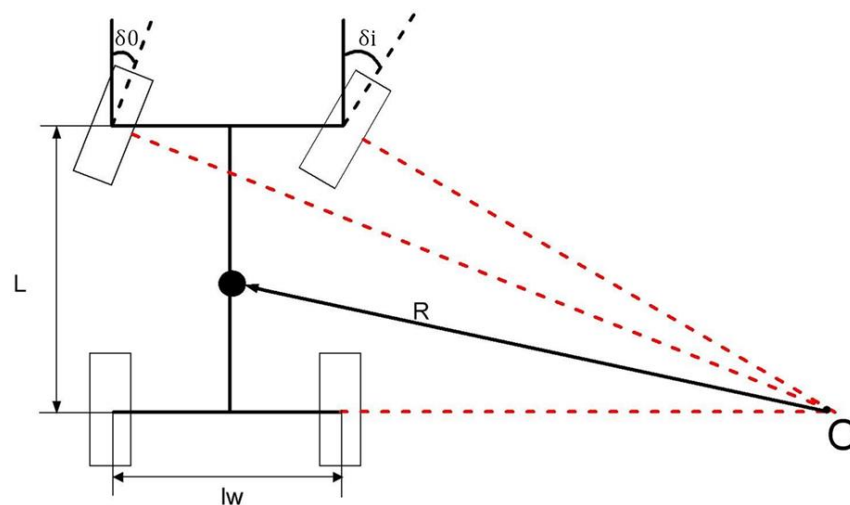


Рисунок 1.6 – Схематичне зображення чотириколісного транспортного засобу при різних кутах повороту коліс

Одним із шляхів досягнення того, щоб два колеса з різними кутами обертання утворювали один центр обертання є керування Аккермана [3] (англ. Ackermann steering). Що виступає спеціальним механізмом, який в свою чергу наближує процес керування авто майже до ідеалу(але не робить його ідеальним).

Таким чином, різниця в кутах повороту двох передніх коліс пропорційна квадрату середнього кута повороту. Таке диференціальне кермо можна отримати за допомогою трапецієподібного пристрою тяги, як показано на рис.1.7. Як видно з рисунку, як для лівого, так і для правого повороту внутрішнє колесо завжди виконує поворот на більший кут.

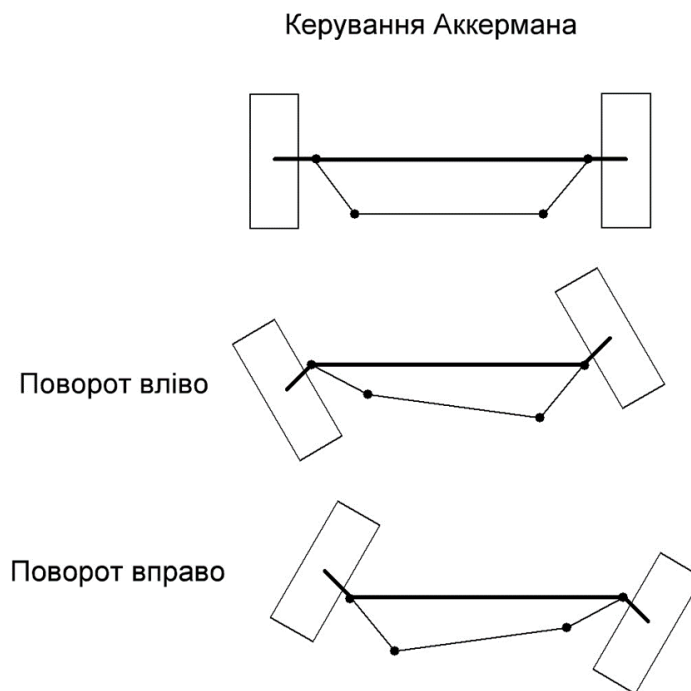


Рисунок 1.7 – Диференціальне рульове управління від трапецієподібної конструкції тяги

Призначення геометрії Аккермана полягає у тому, щоб уникнути необхідності шин ковзати вбік при проходженні по кривій. Геометричне рішення цього полягає в тому, щоб усі колеса мали свої осі, розташовані як радіуси кіл із загальною центральною точкою. Оскільки задні колеса закріплені, ця центральна точка повинна знаходитися на лінії, витягнутій із задньої осі. Перетин осей передніх коліс на цій лінії також вимагає повороту

внутрішнього переднього колеса при рульовому керуванні на більший кут, ніж зовнішнє.

Кожне колесо отримало власне поворотне коло, близьке до власної маточини (ступиці). Це було досягнуто за рахунок зчеплення не простого паралелограма, а завдяки зменшенню довжини колії (рухомої ланки між маточинами), а не довжини осі.

1.3 Кінематична модель велосипеда для чотириколісного транспортного засобу

Доступно багато моделей динамічних транспортних засобів різної складності та точності. Однією з найпростіших і найбільш загальноживаних моделей є модель велосипеда (рис.1.8). Термін «велосипед» обумовлений тим, що обидва передні колеса прийняті як єдине ціле, це також стосується і обох задніх коліс, що робить його двоколісною моделлю.

Кінематична модель виведена з використанням не законів руху Ньютона, а Аккермана. Оскільки кінематична модель отримана з припущенням, що відсутній кут ковзання шини, вона є надійною, коли вектор швидкості кожного колеса знаходиться в напрямку колеса. Це є причиною того, чому кінематична модель використовується для ситуацій з обмеженою низькою швидкістю, таких як розумна система допомоги при паркуванні SPAS (smart parking assist system).

1.4 Розгляд кінематичної моделі велосипеда

При розробці контролерів для транспортних засобів та в процесі побудови математичної моделі для автомобіля також дуже часто використовується спрощений варіант моделі. Причиною цього є те, що «модель велосипеда» значно спрощує розробку контролера. Але, в свою чергу, ця модель все ще відображає динаміку автомобіля достатньо добре, що дозволяє вести надійну розробку контролерів.

В процесі розробки контролери замість цілого транспортного засобу, використовують модель велосипеда, як математичну модель для опису входів і виходів автомобіля.

Отже, іншими словами, замість того, щоб будувати математичну модель для автомобіля з чотирма колесами, двома ззаду і двома спереду доцільніше будувати двоколісну модель. Для чого два передні колеса автомобіля об'єднуються у одне переднє колесо, а два задніх відповідно – у заднє. Після чого визначається центр мас (на рис.1.8 «Ц.М.»), відповідно до якого в подальшому виведенні буде поставлено референсну точку.

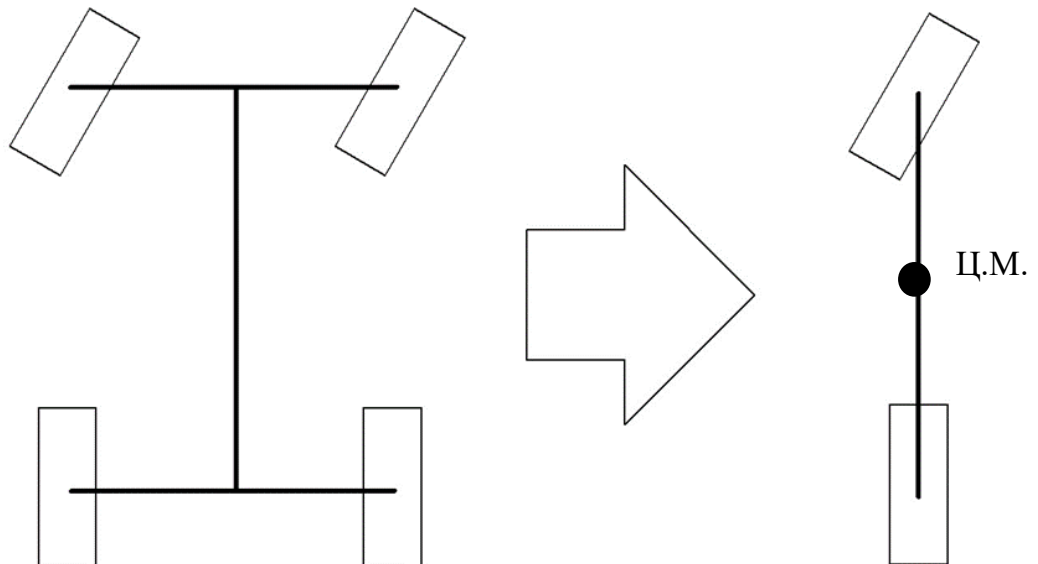


Рисунок 1.8 – Наглядне представлення перетворення до кінематичної моделі велосипеда

Якщо розглянути детальніше модель велосипеда (рис.1.9). Проведенням перпендикулярів від кожного з коліс знаходиться центр обертання. Точкою пересічення проведених прямих і буде миттєвий центр обертання (назвемо його «О»), навколо якого виконуватиме поворот наш транспортний засіб.

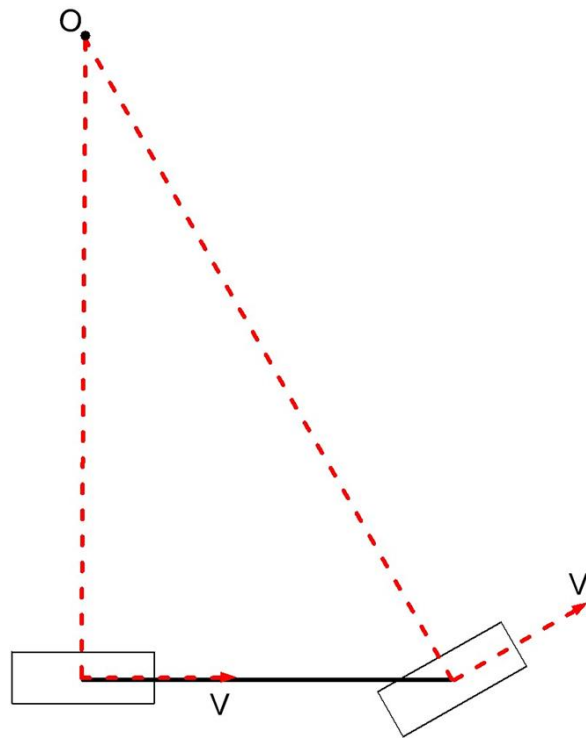


Рисунок 1.9 – Схема напрямів швидкостей кожного з коліс

Варто наголосити, що при повороті машини у неї присутня деяка поперечна швидкість. Тобто у кожного колеса моделі присутня повздовжня швидкість в напрямку його обертання.

При повороті переднього колеса, вектор його швидкості повертається на кут δ , тобто на кут повороту колеса. Якщо взяти будь-яку точку на кузові авто та провести пряму, до точки обертання, та перпендикуляр до проведеної прямої, то буде отримано кут, який покаже напрямок поперечної швидкості тієї чи іншої точки кузова (див.рис.2.10). Чим ближче така точка розміщуватиметься до переднього колеса, тим більше кут вектора швидкості θ буде наближатися до кута повороту колеса. Якщо провести цю ж саму операцію для точки центру мас авто, то буде отримано кут β , що називається кутом ковзання. Це кут між напрямком повороту колеса та дійсним напрямком руху автомобіля.

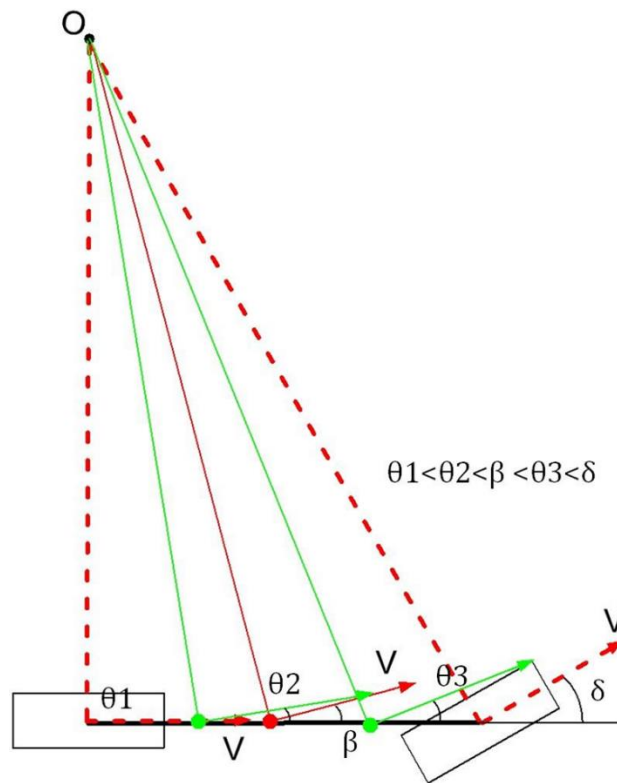


Рисунок 1.10 – Схема напрямів швидкостей декількох точок кінематичної моделі велосипеда

При фіксованому куті повороту керма автомобіль рухається по круговій дузі. З цієї причини криві на дорогах – це кругові дуги або клотоїди, що полегшує життя водієві, оскільки постійний або плавно змінюваний кут нахилу керма дозволяє машині йти по дорозі.

Висновки до розділу 1

В даному розділі було розглянуто особливості поведінки автомобіля на дорозі при виконанні повороту.

Оглянуто історичний перехід до використання геометрії Аккермана в поворотних механізмах чотириколісних транспортних засобів, причини цього переходу. Висунуто недоліки варіантів поворотних механізмів при повороті осі, та кожного колеса окремо на один і той самий кут.

Виконано огляд кінематичної моделі велосипеда для чотириколісного транспортного засобу, її використання та особливості поведінки.

2 КІНЕМАТИКА ТРАНСПОРТНОГО ЗАСОБУ

2.1 Припущення щодо задачі кінематики транспортного засобу

Припущення щодо проектування:

- 1) Усі колеса не є гнучкими і не містять гнучких деталей.
- 2) Осі, з'єднані двома колесами, паралельні поверхні.
- 3) Всі колеса перпендикулярні поверхні, без бічного прогину.

Припущення щодо процесу водіння:

- 4) Транспортний засіб рухається по площині.
- 5) Контакт між колесом і поверхнею є точковим.
- 6) Колеса обертаються в русі, це чисто кочення, без бокового ковзання, гальмування та ковзання.

Припущення 1 передбачає, що весь транспортний засіб - це жорстка механіка кузова на землі. Припущення 2 і 3 вказані для зменшення дальності автомобіля, яку повинна враховувати методологія, обмежуючи складність кінематичної моделі. Припущення 4 нехтує нерівностями на дійсній поверхні землі, на якій рухається транспортний засіб. Оскільки транспортний засіб, що розглядається майже не їздить на стандартних магістралях, можна нехтувати грубими, вибоїстими або кам'янистими поверхнями. Припущення 5 забезпечує придатність теоретичних кінематичних властивостей колеса в умовах кочення [6]. Припущення 6 ілюструє, що при повороті колесо повинно обертатися навколо точок дотику.

2.2 Система координат

Спочатку призначається система координат як для транспортного засобу, так і для глобальних обставин [8]. Автомобіль має чотири колеса, два передні поворотні колеса, та два задні ведучі. $X_l O Y_l$ - глобальна координата, а $X_R M Y_R$ - локальна система координат. M_r - середня точка задньої осі транспортного засобу, і якщо припустити, що це оригінал $X_R M Y_R$. У $X_l O Y_l$, M_r визначається (x_r, y_r) , $M_f(x_f, y_f)$ - середня точка передньої осі. l - відстань між задніми і передніми осями; φ - кут повороту передніх коліс [9]; d - відстань між двома колесами, як показано на рис.2.1.

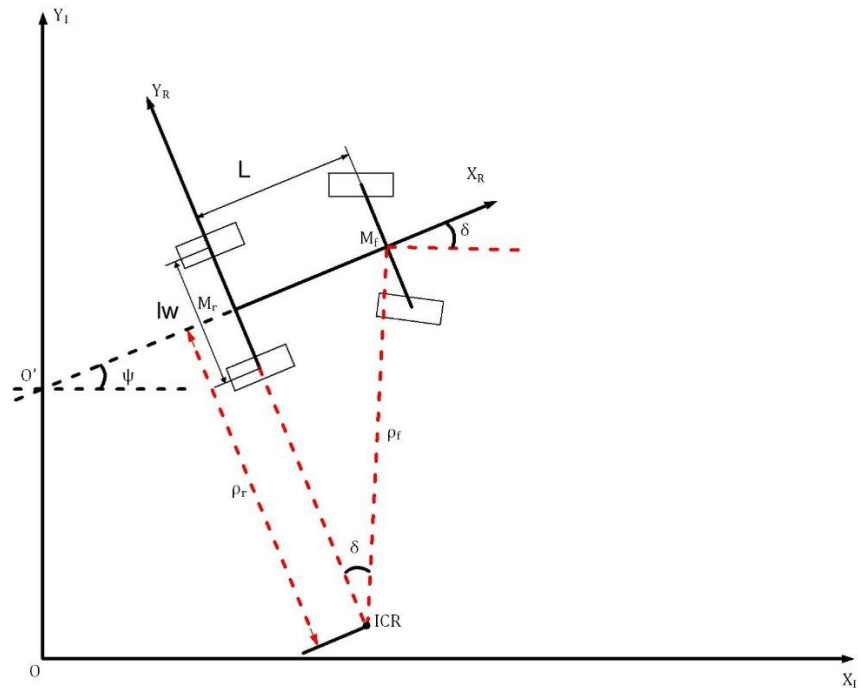


Рисунок 2.1 – Локальна та глобальна координати транспортного засобу

Відношення між M_r та M_f :

$$\begin{cases} x_f = x_r + l \cdot \cos \psi \\ y_f = y_r + l \cdot \sin \psi \end{cases} \quad (2.1)$$

У рівнянні (2.1) ψ - кут між $X_I O Y_I$ та $X_R M Y_R$ і він відображає напрямок руху транспортного засобу в глобальній координатній системі [10]. Рух транспортного засобу в $X_R M Y_R$ можна описати трьома компонентами x , y та ψ .

Нехай $\zeta^T = [x \ y \ \psi]^T$ - вектор пози руху в $X_I O Y_I$.

Якщо припустити, що транспортний засіб обертається навколо Миттєвого центру обертання (Instantaneous Centre of Rotation (ICR)) за надзвичайно короткий час, коли він змінює напрямок [6], то шлях руху можна описати як коло [7], а ρ_r і ρ_f є радіусами кола ICR.

$R(\psi)$ - матриця перетворення, яка може перетворити координату точки в $X_I O Y_I$ в точку в $X_R M Y_R$, тобто $\zeta^R = R(\psi) \zeta^I$, навпаки, $\zeta^I = R(\psi)^{-1} \zeta^R$ існує.

$R(\psi)^{-1}$ визначається наступним рівнянням:

$$R(\psi)^{-1} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

2.3 Кінематичний опис

Нехай швидкість руху всього транспортного засобу запишеться як v_r , напрямком вектору якого співпадає з X_R ; v_f - швидкість передніх коліс, напрямком вектору яких спрямовано вперед. Відношенням між v_r та v_f виступає $v_f = \frac{v_r}{\cos\delta}$.

За ультракороткий час Δt транспортний засіб рухається навколо ICR, передня вага X_R задніх коліс $v_r dt$, а вага руху за напрямком Y_R відсутня. Передня вага X_R передніх коліс дорівнює $v_f \cos\phi dt$, а в напрямку Y_R передніх коліс $v_f \cdot \sin\phi dt$. Якщо ϕ не змінюється, то шлях транспортного засобу - це коло. За Δt воно переміщує Δs , тоді існує $ds = r d\psi$, це означає $r \cdot d\psi = v \cdot dt$.

Крім того, в $\Delta M_r M_f ICR$, $\rho_r = \frac{L}{\tan\delta}$ і $\rho_f = \frac{L}{\sin\delta}$ існують. Тоді можна довести,

що $d\psi = \frac{v_r \cdot dt}{\rho_r} = \frac{v_r \cdot \tan\phi}{L} dt$, тоді рівняння (2.3) та (2.4) можуть бути доведені:

$$\frac{d\psi}{dt} = \frac{\tan\delta}{L} \cdot v_r \quad (2.3)$$

$$\frac{d\psi}{dt} = \frac{\sin\delta}{L} \cdot v_f \quad (2.4)$$

Тоді можна отримати вектор руху пози $\dot{\zeta}_r^R$ та $\dot{\zeta}_f^R$ у $X_R M Y_R$, так як показано у рівняннях (2.5) та (2.6) :

$$\dot{\zeta}_r^R = [v_r \quad 0 \quad \frac{\tan\delta}{L} \cdot v_r]^T \quad (2.5)$$

$$\dot{\zeta}_f^R = [v_f \cdot \cos\delta \quad v_f \cdot \sin\delta \quad \frac{\sin\delta}{L} \cdot v_f]^T \quad (2.6)$$

Тоді рівняння M_r стану позиції в координатній площині $X_I O Y_I$ описується рівнянням (2.7), що можна довести за допомогою (2.2) та (2.5).

$$\zeta_r^I = R(\psi)^{-1} \cdot \zeta_r^R = [v_r \cdot \cos \psi \quad v_r \cdot \sin \psi \quad \frac{v_r}{L} \cdot tg \delta]^T \quad (2.7)$$

Рівняння стану позиції M_f в координатній площині $X_I O Y_I$ описано рівнянням (2.8), що може бути доведено за допомогою (2.2) та (2.6).

$$\zeta_f^I = [v_r \cdot \frac{\cos(\psi + \delta)}{\cos \delta} \quad v_r \cdot \frac{\sin(\psi + \delta)}{\cos \delta} \quad \frac{v_r}{L} \cdot tg \delta]^T \quad (2.8)$$

Рівняння (2.7) та (2.8) є ідеальними кінематичними рівняннями транспортних засобів.

2.4 Кінематичне моделювання за принципом Аккермана

Зазвичай транспортні засоби мають кермовий механізм Аккермана між двома передніми колесами, і цей принцип приблизно забезпечує критерій зчеплення приводу, забезпечуючи правильні кути нахилу колеса, щоб уникнути ковзання з боку колеса.

Нехай δ_i – кут повороту для внутрішнього колеса, а δ_0 – для зовнішнього. Для того, щоб забезпечити поворот транспортного засобу без поперечного ковзання, чотири колеса повинні обертатися паралельно ICR, крім того, δ_0 і δ_i повинні відповідати рівнянню (2.9)

$$\begin{cases} ctg \delta_0 = ctg \delta + \frac{lw}{2 \cdot L} \\ ctg \delta_i = ctg \delta - \frac{lw}{2 \cdot L} \end{cases} \quad (2.9)$$

Потім можна отримати рівняння (2.10)

$$ctg \delta_0 - ctg \delta_i = \frac{\rho_r + lw/2}{L} - \frac{\rho_r - lw/2}{L} = \frac{lw}{L} \quad (2.10)$$

Нехай ρ_f^l - радіус ICR лівого переднього колеса, ρ_f^r - радіус ICR правого переднього колеса, ρ_r^l - радіус ICR лівого заднього колеса і ρ_r^r - радіус ICR правого заднього колеса, тоді $\rho_f^l = L / \sin \delta_0$, $\rho_f^r = L / \sin \delta_i$, $\rho_r^l = L / tg \delta + lw/2$ і $\rho_r^r = L / tg \delta - lw/2$. В свою чергу Δr буде $\rho_f^l - \rho_r^l$, коли транспортний засіб повертає праворуч (при повороті ліворуч Δr дорівнює $\rho_f^r - \rho_r^r$).

2.5 Кінематична модель, що враховує вплив похибки

Нехай e_r^R - вектор похибки заднього колеса, тоді може бути визначене $\sigma \cdot [x_e \ y_e \ \delta_e]$. $x_e(t)$, $y_e(t)$ та $\delta_e(t)$ змінюються з часом t . σ - похибка кута повороту φ .

Тоді рівняння (2.5) приймає наступну форму (див. р-н.(2.11)):

$$\dot{\zeta}_r^R = [v_r + \dot{x}_e \ \dot{y}_e \ \frac{tg(\delta + \delta_e)}{L} \cdot v_r]^T \quad (2.11)$$

А рівняння (2.7) прийме вигляд рівняння (2.12):

$$\begin{aligned} \zeta_r^I = & [v_r \cdot \cos \psi \ v_r \cdot \sin \psi \ \frac{v_r}{L} \cdot tg \delta]^T + \\ & + [\dot{x}_e \cdot \cos \psi - \dot{y}_e \cdot \sin \psi \ \dot{x}_e \cdot \sin \psi + \dot{y}_e \cdot \cos \psi \ \delta_e]^T \end{aligned} \quad (2.12)$$

Величина ϕ_e визначена рівнянням (2.13)

$$\phi_e = \frac{v_r}{L} \cdot \frac{tg \delta_e \cdot (1 + tg^2 \delta)}{1 - tg \delta \cdot tg \delta_e} \quad (2.13)$$

Коли помилки та інші фактори руйнують неголономічне обмеження транспортного засобу, дальність може бути оголошена k_x та k_y . $k_x = x_e \cdot \cos \psi - y_e \cdot \sin \psi$, $k_y = x_e \cdot \sin \psi + y_e \cdot \cos \psi$, яка відображає помилку позиції від локальної системи координат до загальної системи координат. Нехай $e_r^I = [k_x \ k_y]^T$ не має відношення до стану автомобіля, але знаходиться під впливом різного роду похибок. Якщо не вжити належних заходів для контролю накопичення помилок e_r^I , то це призведе до втрати контролю над транспортним засобом.

2.6 Приведення до кінематичної моделі велосипеда

Розглядається опис кінематичної моделі велосипеда [3] (рис.2.2) рівняннями.

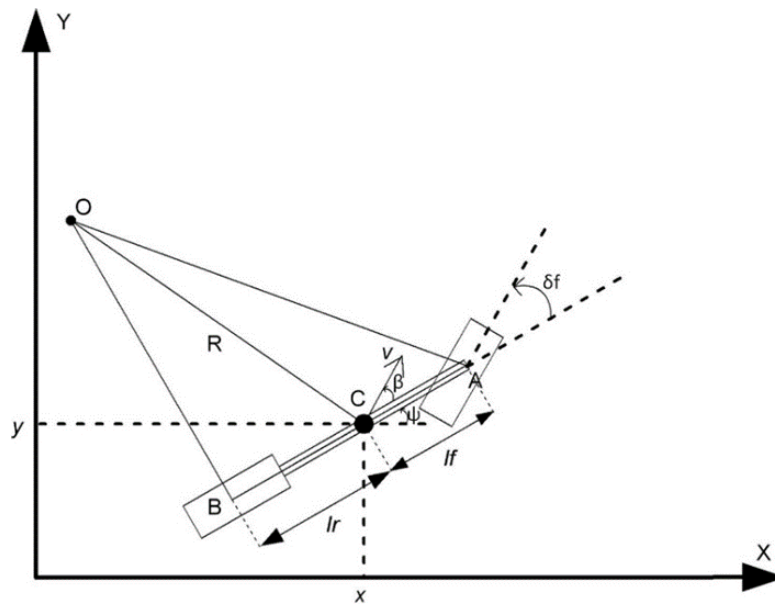


Рисунок 2.2 – Кінематична двокісна модель транспортного засобу

Нехай:

Кути повороту: δ_f δ_r

Припускається, що передніми і задніми колесами можна керувати.

Для випадку, коли тільки переднє колесо можна повертати $\delta_r = 0$

C – центр мас

$L = l_f + l_r$: колісна база транспортного засобу

ψ – кут орієнтації транспортного засобу відносно глобальної осі x

v – швидкість центра мас

β – кут ковзання

R – круговий радіус

O – центр повороту

Кут ходу транспортного засобу $\gamma = \psi + \beta$

Розглянемо величини у вигляді трикутників OCA та OCB на рис.2.2.

Маємо трикутник OCA, та наступні співвідношення в ньому :

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin(\frac{\pi}{2} - \delta_f)}{R} \quad (2.14)$$

Або

$$\frac{\sin \delta_f \cdot \cos \beta - \sin \beta \cdot \cos \delta_f}{l_f} = \frac{\cos \delta_f}{R} \quad (2.15)$$

або

$$\operatorname{tg} \delta_f \cdot \cos \beta - \sin \beta = \frac{l_f}{R} \quad (2.16)$$

Тоді параметри трикутника ОСВ можемо описати наступним чином:

$$\frac{\sin(\beta - \delta_r)}{l_r} = \frac{\sin(\frac{\pi}{2} + \delta_r)}{R} \quad (2.17)$$

Або

$$\frac{\sin \beta \cdot \cos \delta_r - \sin \delta_r \cdot \cos \beta}{l_r} = \frac{\cos \delta_r}{R} \quad (2.18)$$

або

$$\sin \beta - \operatorname{tg} \delta_r \cdot \cos \beta = \frac{l_r}{R} \quad (2.19)$$

Співвіднівши рівняння параметрів обох трикутників буде отримано наступний вираз:

$$(\operatorname{tg} \delta_f - \operatorname{tg} \delta_r) \cdot \cos \beta = \frac{l_f + l_r}{R} \quad (2.20)$$

З отриманих рівнянь виведеться вираз для кута ковзання β . Для цього розглянеться рівняння (2.16) та перемножаться всі його члени на l_r , звідки буде отримано:

$$l_r \cdot \operatorname{tg} \delta_f \cdot \cos \beta - l_r \cdot \sin \beta = \frac{l_r \cdot l_f}{R} \quad (2.21)$$

Виконавши подібну операцію для наступного виразу, тобто для (2.19), відповідно виведеться рівняння:

$$l_f \cdot \sin \beta - l_f \cdot \operatorname{tg} \delta_r \cdot \cos \beta = \frac{l_r \cdot l_f}{R} \quad (2.22)$$

Об'єднавши отримані вище рівняння вийде наступний вираз:

$$(l_f \cdot \operatorname{tg} \delta_r + l_r \cdot \operatorname{tg} \delta_f) \cdot \cos \beta - (l_f + l_r) \cdot \sin \beta = 0 \quad (2.23)$$

Нехай дане рівняння запишеться інакше:

$$(l_f \cdot \operatorname{tg} \delta_r + l_r \cdot \operatorname{tg} \delta_f) \cdot \cos \beta = (l_f + l_r) \cdot \sin \beta \quad (2.24)$$

Звідки, використовуючи тригонометричну тотожність, виведеться рівняння для $\operatorname{tg} \beta$:

$$\operatorname{tg} \beta = \frac{l_f \cdot \operatorname{tg} \delta_r + l_r \cdot \operatorname{tg} \delta_f}{l_f + l_r} \quad (2.25)$$

Звідки

$$\beta = \arctg\left(\frac{l_f \cdot \operatorname{tg} \delta_r + l_r \cdot \operatorname{tg} \delta_f}{l_f + l_r}\right) \quad (2.26)$$

Кут ковзання представлений кутами повороту та колісною базою

$$(\operatorname{tg} \delta_f - \operatorname{tg} \delta_r) \cdot \cos \beta = \frac{l_f + l_r}{R} \quad (2.27)$$

Кутова швидкість при цьому:

$$\dot{\psi} = \frac{v}{R} \quad (2.28)$$

Тому буде отримано наступну систему рівнянь для кінематичної моделі велосипеда:

$$\begin{aligned} \dot{\psi} &= \frac{v \cdot (\operatorname{tg} \delta_f - \operatorname{tg} \delta_r) \cdot \cos \beta}{l_f + l_r} \\ \dot{X} &= v \cdot \cos(\psi + \beta) \\ \dot{Y} &= v \cdot \sin(\psi + \beta) \end{aligned} \quad (2.29)$$

2.7 Інші варіанти реалізації кінематичної моделі

В представленому раніше прикладі було показано кінематичну двоколійну модель транспортного засобу при референсній точці в центрі тяги самого транспортного засобу. Але можливі і інші варіанти розташування референсної точки.

Якщо точку відрахування розташувати на задньому колесі двоколійної кінематичної моделі, вийде наступна картинка:

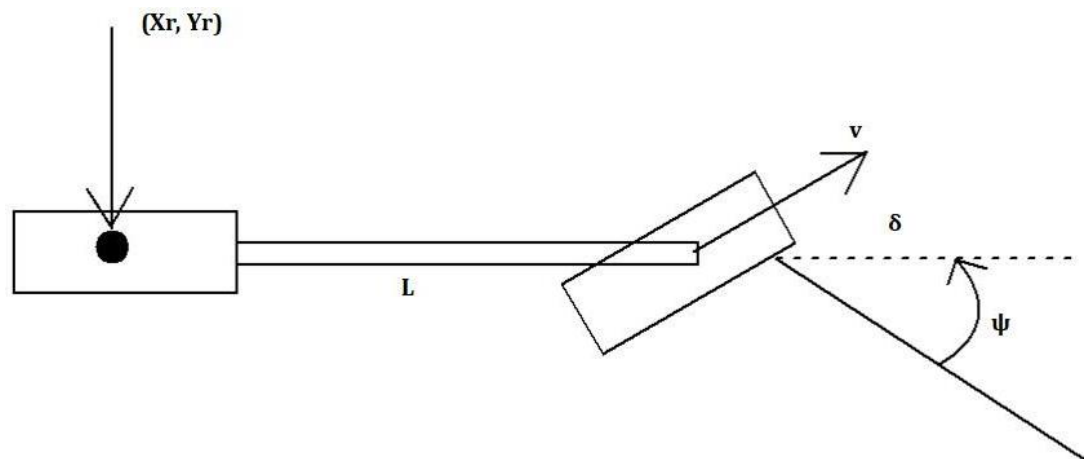


Рисунок 2.6 – Кінематична двоколійна модель з розташуванням точки на задньому колесі

В такому випадку, виходячи з попереднього математичного виведення, кінематична модель «велосипеда» прийме наступний вигляд:

$$\begin{aligned}\dot{\psi} &= \frac{v \cdot \tan \delta}{L} \\ \dot{X}_r &= v \cdot \cos(\psi) \\ \dot{Y}_r &= v \cdot \sin(\psi)\end{aligned}\tag{2.30}$$

Також можливий ще один варіант, коли референсна точка знаходиться на передньому колесі, яке власне і виконує поворот, що матиме наступний вигляд:

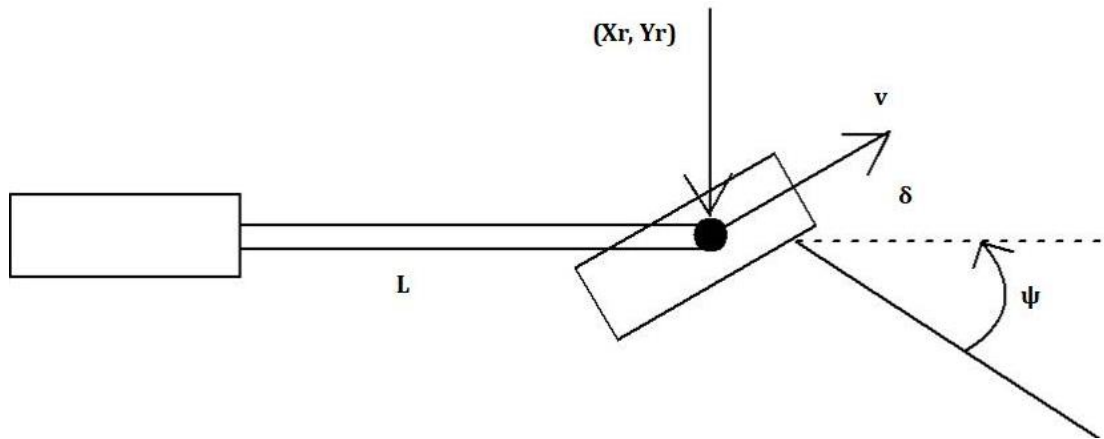


Рисунок 2.7 – Розташування референсної точки на передньому колесі

Виходячи з варіанту розташування референсної точки та з математичного приведення, розписаному вище двоколісна кінематична модель прийме наступний вигляд:

$$\begin{aligned}\dot{\psi} &= \frac{v \cdot \sin \delta}{L} \\ \dot{X}_f &= v \cdot \cos(\psi + \delta) \\ \dot{Y}_f &= v \cdot \sin(\psi + \delta)\end{aligned}\quad (2.31)$$

2.8 Обмеження кінематичної моделі велосипеда

Обмеженням моделі велосипеда у випадку моделі Аккермана [3] можна вважати те, що як ліве “ δ_o ”, так і праве “ δ_i ” переднє колеса були представлені одним переднім колесом у моделі велосипеда. Слід зазначити, що лівий і правий кути повороту в цілому будуть приблизно рівними, але не зовсім такими. Це пояснюється тим, що радіус шляху кожного з цих коліс різний (рис.1.6).

Нехай l_w - ширина колії автомобіля, а δ_o і δ_i - зовнішній та внутрішній кути повороту відповідно. Нехай колісна база $L = l_f + l_r$ буде невеликою порівняно з радіусом R.

В моделі велосипеда

$$\dot{\psi} = \frac{v \cdot (tg\delta_f - tg\delta_r) \cdot \cos\beta}{l_f + l_r}\quad (2.32)$$

Якщо кут ковзання β малий, $\delta_r = 0$

$$\dot{\psi} = \frac{v \cdot \delta_f}{l_f + l_r} = \frac{v \cdot \delta_f}{L} \quad (2.33)$$

з іншої сторони

$$\dot{\psi} = \frac{v}{R} \quad (2.34)$$

Оскільки радіус на внутрішньому та зовнішньому колесах різний виконається наступна умова:

$$\begin{aligned} \delta_0 &= \frac{L}{R + \frac{l_w}{2}} \\ \delta_i &= \frac{L}{R - \frac{l_w}{2}} \end{aligned} \quad (2.35)$$

Середній кут повороту переднього колеса приблизно дається

$$\delta_f = \frac{\delta_0 + \delta_i}{2} = \frac{L}{R} \quad (2.36)$$

Різниця між δ_0 і δ_i становить:

$$\delta_i - \delta_0 = \frac{L}{R^2} \cdot l_w = \delta^2 \cdot \frac{l_w}{L} \quad (2.37)$$

Але у розгляд береться лише одне переднє колесо та одне заднє. Адже хоч і кути різні, їх різниця досить мала та для більшості випадків вона не є важливою.

Висновки до розділу 2

В процесі виконання даного розділу було виконано наступні дії:

- Розглянуто ряд припущень щодо спрощення створення кінематичної моделі транспортного засобу.
- Виконано кінематичне моделювання чотириколісного транспортного засобу за принципом Аккермана.
- Розглянуто моделювання з врахуванням похибки.
- Виконано приведення до кінематичної моделі велосипеда.
- Розглянуто інші варіанти кінематичної моделі велосипеда в залежності від розташування референсної точки.
- Розглянуто обмеження кінематичної моделі велосипеда.

3 АЛГОРИТМИ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОБОМ

При проектуванні систем керування транспортними засобами зазвичай розглядаються певні алгоритми переміщення розгляданої машини, наприклад рух до певної заданої позиції або слідування заданій траєкторії.

В цьому ж розділі буде розглянуто алгоритми переміщення машини (математична модель якої подано в попередньому розділі) такі як рух до заданої точки та слідування лінії з певної початкової позиції.

3.1 Рух до точки призначення.

3.1.1 Формулювання задачі

В процесі розгляду поставленої задачі керування припускається, що транспортний засіб рухається у відкритій прямокутній області в підпросторі Евкліда, що відповідає горизонтальній площині.

Метою даної роботи є опис алгоритму для транспортного засобу з метою руху до заданої нерухомої точки. Досягнення заданої точки призначення означає приблизну відповідність позицій машини та точки за певний проміжок часу.

3.1.2 Модель машини

Положення транспортного засобу [4] в декартовій системі координат повністю визначається потрійним вектором $q_R = (\dot{X}, \dot{Y}, \psi)^T$, де \dot{X} та \dot{Y} представляють координати центру мас машини, а ψ - положення машини відносно осі X. Кінематичними рівняннями руху виступатимуть:

$$\begin{aligned}\dot{\psi} &= \frac{v \cdot \operatorname{tg} \delta}{L} \\ \dot{X}_v &= v \cdot \cos(\psi) \\ \dot{Y}_v &= v \cdot \sin(\psi)\end{aligned}\tag{3.1}$$

Розглядається представлення полярних координат рівнянь кінематики транспортного засобу. Полярне представлення машини отримують, враховуючи наступну зміну змінної

$$\begin{aligned} x &= r \cdot \cos(\lambda) \\ y &= r \cdot \sin(\lambda) \end{aligned} \quad (3.2)$$

де r - радіальна координата, а λ – кутова координата, як показано на рис.3.1

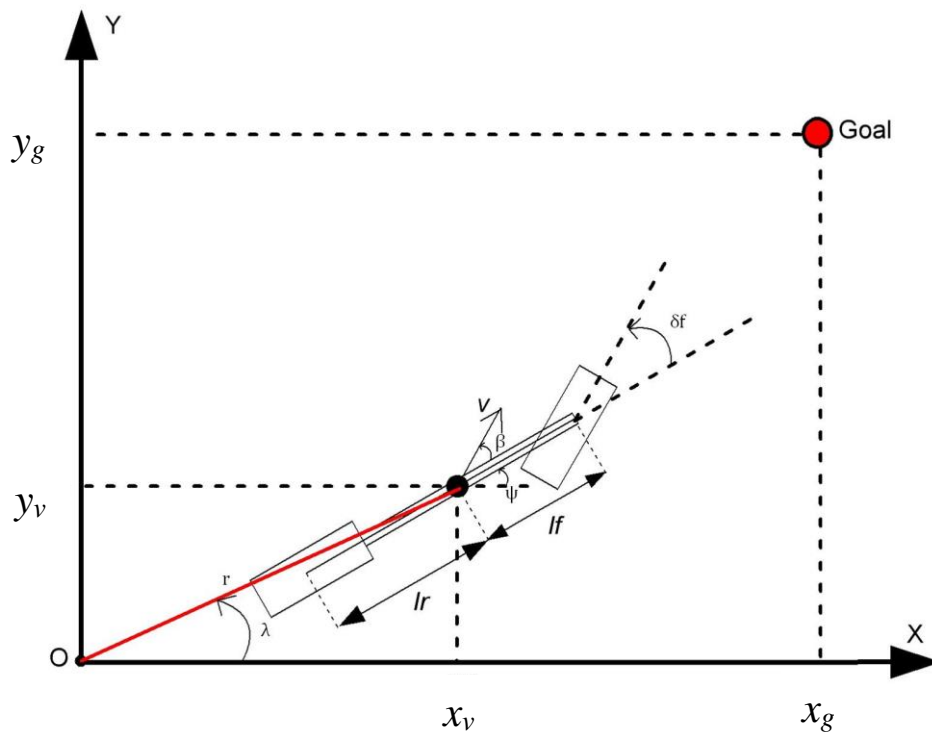


Рисунок 3.1 – Декартове та полярне зображення робота та точки
Беручи похідну від часу для r і λ , буде отримано наступні рівняння:

$$\begin{aligned} \dot{r} &= \frac{\dot{x} \cdot x + \dot{y} \cdot y}{r} \\ \dot{\lambda} &= \frac{\dot{y} \cdot x - \dot{x} \cdot y}{r^2} \end{aligned} \quad (3.3)$$

Нехай v_r і v_λ - компоненти вектора швидкості робота вздовж та поперек лінії OR. Вирази для v_r та v_λ можна отримати, співвіднівши рівняння кінематики (3.1) та зміну змінної (3.2), що призводить до наступної системи:

$$\begin{aligned} v_r &= \dot{r} = v \cdot \cos(\psi - \lambda) \\ v_\lambda &= r \cdot \dot{\lambda} = v \cdot \sin(\psi - \lambda) \end{aligned} \quad (3.4)$$

3.1.3 Геометрія руху до точки та кінематичні рівняння

Геометрія поставленої задачі в горизонтальній площині показана на рис.3.2. Досліджуваний транспортний засіб позначено як R, а точка, до якої потрібно прямувати – G. Позиції машини та точки призначення в інерційній системі відліку координат задаються векторами $r_R = OR$ та $r_G = OG$, де O є початком системи відліку. Очевидно, що вектор швидкості для машини можна записати наступним чином.

$$v = \dot{r} = v_r \cdot e_r + v_\lambda \cdot e_\lambda = \dot{X}_V \cdot e_x + \dot{Y}_V \cdot e_y \quad (3.5)$$

де e_r та e_λ – одиничні вектори вздовж та поперек ліній OR або OG, а e_x і e_y – одиничні вектори в декартовій системі відліку. З рис.3.2 ми визначаємо такі геометричні величини:

- Лінія зору LG - це пряма лінія, яка починається в опорній точці робота і спрямована на ціль.
- Евклідова дистанція від машини до заданої точки позначається r_{GR}

Проекція відносної відстані на осі X та Y дає нам наступне:

$$\begin{aligned} x_d &= X_G - X_V \\ y_d &= Y_G - Y_V \\ r_{GV} &= \sqrt{(X_G - X_V)^2 + (Y_G - Y_V)^2} \end{aligned} \quad (3.6)$$

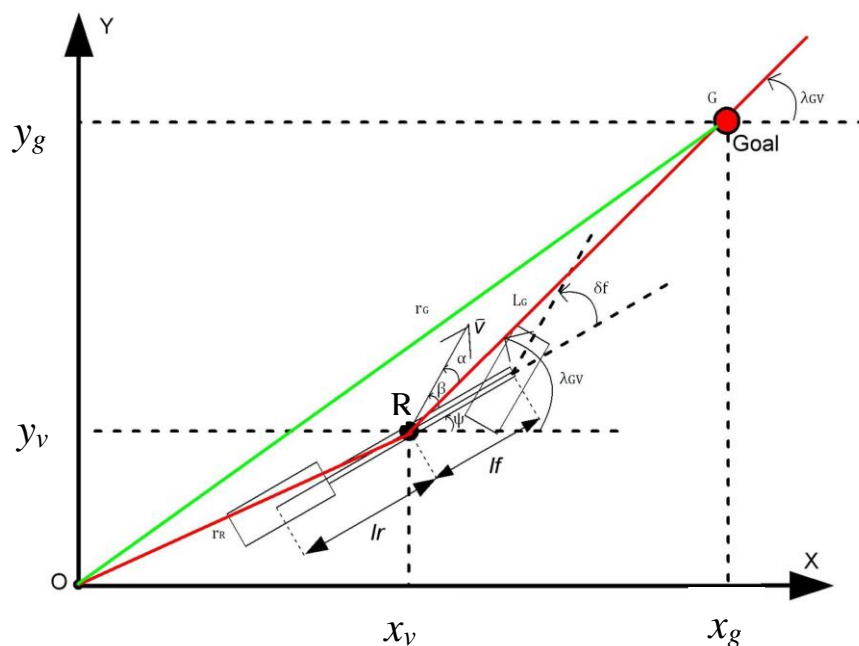


Рисунок 3.2 – Геометрія поставленої задачі

- Кут лінії зору авто-точка - це кут від позитивної осі x до лінії зору. Цей кут позначається:

$$\begin{aligned} \operatorname{tg} \lambda_{GR} &= \frac{y_d}{x_d} \\ \lambda_{GR} &= \operatorname{arctg} \frac{y_d}{x_d} \end{aligned} \quad (3.7)$$

- Кут α – ведучий кут лінії зору до вектора швидкості.

Кут лінії зору не визначений для $r_{GV} = 0$, оскільки положення машини і точки призначення збігаються, коли $r_{GV} = 0$.

3.1.4 Кінематика руху моделі

Відносну швидкість \dot{r}_V можна розкласти на радіальну та тангенціальну складові. Розглядаючи отриману модель кінематики для транспортного засобу та цілі:

$$\begin{aligned} v_r &= \dot{r}_V = v \cdot \cos \alpha \\ v_\lambda &= r_V \cdot \dot{\lambda} = v \cdot \sin \alpha \end{aligned} \quad (3.8)$$

Де

$$\alpha = \psi - \lambda_{GV} \quad (3.9)$$

Модель відносної кінематики в полярних координатах буде використана в цій роботі для моделювання задачі навігації. Кінематична модель у рівнянні (3.8) враховує лінійну та кутову швидкість транспортного засобу відносно точки призначення, а також кут огляду, що представляє важливу геометричну величину. Ця модель полегшує виведення та аналіз навігаційної стратегії.

3.2 Слідування по лінії

3.2.1 Формулювання задачі

В цій частині розділу потрібно розглянути та описати алгоритм руху транспортного засобу відносно заданої лінії від початкової точки розташування.

3.2.2 Кінематика руху

Отже завдання алгоритму [5] полягає в тому, щоб керувати транспортним засобом слідуючи по поточній прямій лінії, таким чином зменшуючи відстань до лінії, позначеної як d , до нуля. Похибка напрямку визначається як $\beta = \psi - \psi_L$, і вона представляє різницю між напрямком транспортного засобу $\psi \in [-\pi, \pi)$ та орієнтацією на опорну лінію ψ_L . Швидкість транспортного засобу дається як v .

Кінематична модель лінії, що йде в горизонтальній площині, дається як:

$$\begin{aligned}\dot{d} &= v \cdot \sin \beta \approx v \cdot \beta \\ \dot{\beta} &= r\end{aligned}\tag{3.10}$$

де символ “ \approx ” позначає лінійне наближення для невеликих значень β . З урахуванням усіх вищезазначених спрощень моделі через рух транспортного засобу в горизонтальній площині, положення та орієнтація транспортного засобу $[x \ y \ \psi]$ в координатній площині «е» виражаються як:

$$\begin{aligned}\dot{\psi} &= r \\ \dot{X}_v &= v \cdot \cos(\psi) + v_x \\ \dot{Y}_v &= v \cdot \sin(\psi) + v_y\end{aligned}\tag{3.11}$$

де v_x, v_y - x та y складові поточної швидкості відповідно, а r - швидкість похибки.

3.2.3 Опис алгоритму

Притримування лінії, визначеної як $a \cdot x + b \cdot y + c = 0$, потребує два регулятори для регулювання рульового керування[1]. Один з них визначається як:

$$\alpha_d = -K_d \cdot d, K_d > 0\tag{3.12}$$

Він повертає машину до лінії, з ціллю мінімізувати відстань d до неї.

$$d = \frac{a \cdot x + b \cdot y + c}{\sqrt{a^2 + b^2}}\tag{3.13}$$

Другий контролер регулює орієнтацію автомобіля ψ так, щоб він був паралельним лінії.

$$\psi^* = \arctg\left(-\frac{a}{b}\right) \quad (3.14)$$

Для цього використовується пропорційний регулятор α_h :

$$\alpha_h = K_h \cdot (\psi^* - \psi), K_h > 0 \quad (3.15)$$

Комбінований закон керування γ обертає кермо таким чином, щоб вести транспортний засіб до лінії та рухатися вздовж нього.

$$\gamma = -K_d \cdot d + K_h \cdot (\psi^* - \psi) \quad (3.16)$$

Висновки до розділу 3

В даному розділі було виконано аналітичний огляд алгоритмів керування чотириколісним транспортним засобом, а саме:

- рух до точки, де розглянуто модель машини, виведення геометрії руху до заданої точки, а також кінематика моделі руху, звідки вияснилось, що дана модель враховує лінійну та кутову швидкість транспортного засобу відносно точки призначення, а також кут огляду, що в свою чергу сприяє на полегшення виведення та аналізу навігаційної стратегії;
- слідування лінії з заданого положення, а саме кінематику руху, звідки виведено положення та орієнтацію транспортного засобу в координатній площині, а також виконано опис самого алгоритму, розписано використовувані в алгоритмі регулятори рульового керування.

4 ДОСЛІДЖЕННЯ АЛГОРИТМІВ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОБОМ

Задачею даного розділу виступає дослідження алгоритмів керування транспортним засобом шляхом математичного моделювання в середовищі Simulink програмного пакету MatLab.

Основним предметом дослідження алгоритмів керування виступатиме точність позиціонування транспортного засобу за заданими конфігураціями початкової та кінцевої (у випадку дослідження алгоритму руху до точки) позицій транспортного засобу на площині. А також вплив коефіцієнтів регуляторів на точність руху та позиціонування транспортного засобу.

В якості досліджуваного транспортного засобу виступатиме машина приведена до кінематичної моделі велосипеда, до якої будуть застосовуватися описані у минулому розділі алгоритми руху до заданої точки та слідування по лінії.

Блок моделі транспортного засобу було взято з бібліотеки Robotic Toolbox середовища Simulink. Внутрішній вигляд блоку можна побачити на рисунку 4.1.

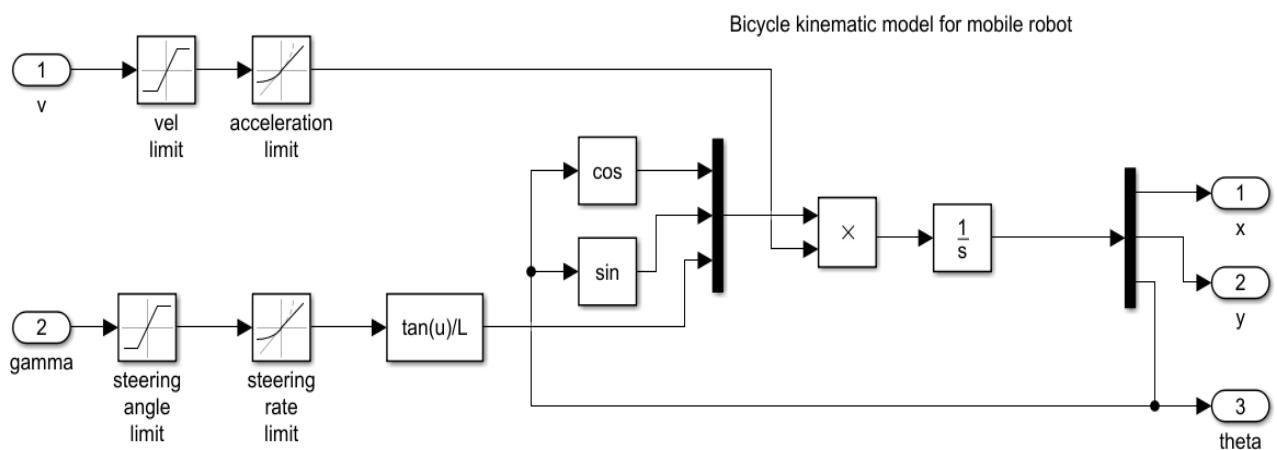


Рисунок 4.1 – Зображення блоку кінематичної моделі велосипеда для транспортного засобу

4.1 Алгоритм руху до заданої точки

Для початку буде досліджено алгоритм руху до заданої точки, модель якого виглядає наступним чином (рис.4.2)

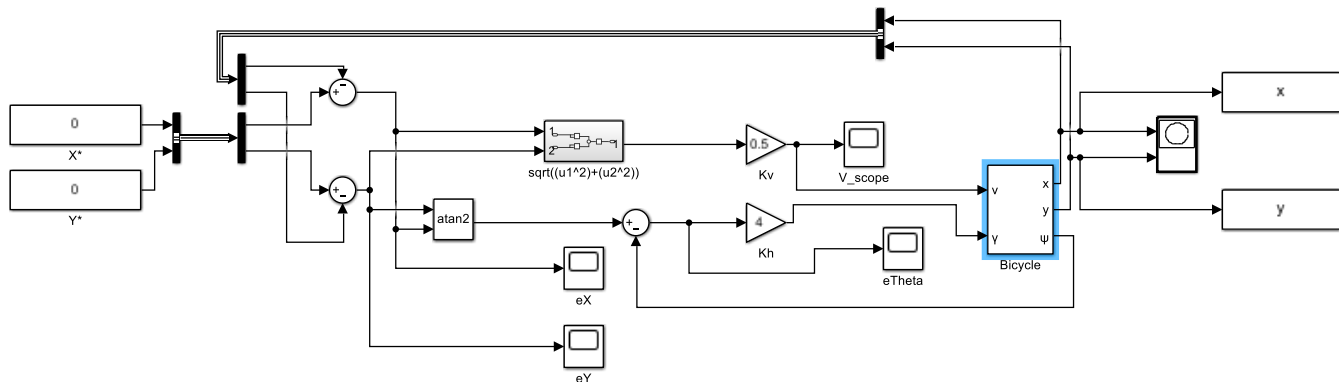


Рисунок 4.2 – Математична модель алгоритму руху до заданої точки керування транспортним засобом

Для початку встановлено початкову та кінцеву позицію руху транспортного засобу на площині. Нехай початковою точкою розташування транспортного засобу виступатиме (8, 5), при цьому повернено транспортний засіб буде в напрямку осі Y та паралельно ній. А кінцевою точкою позиціонування виступатиме (0, 0).

Для першого дослідження коефіцієнтам K_h та K_v було присвоєно наступні значення: $K_h=4$, а $K_v=0.5$, так як при даних значеннях досліджуваний алгоритм має відпрацьовуватись правильно. Тобто похибки повинні прямувати до нуля, швидкість знижуватись з приближенням транспортного засобу до точки призначення, а траєкторія правильно відпрацьовуватись.

Виконавши моделювання для вище заданих параметрів, було отримано наступні результати роботи.

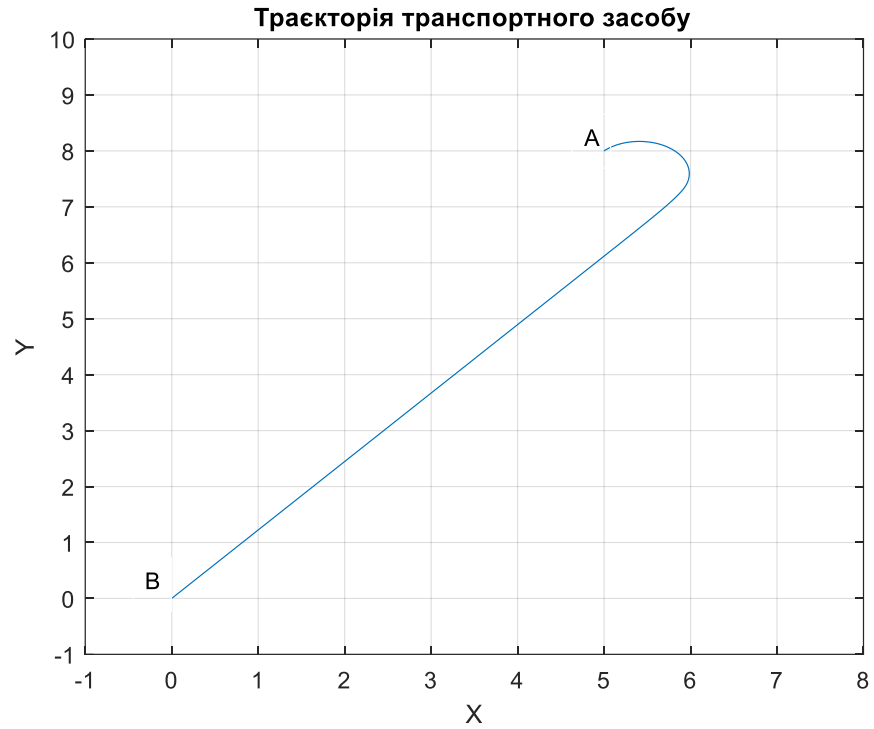


Рисунок 4.3 – Траєкторія руху транспортного засобу на площині від точки А до точки В

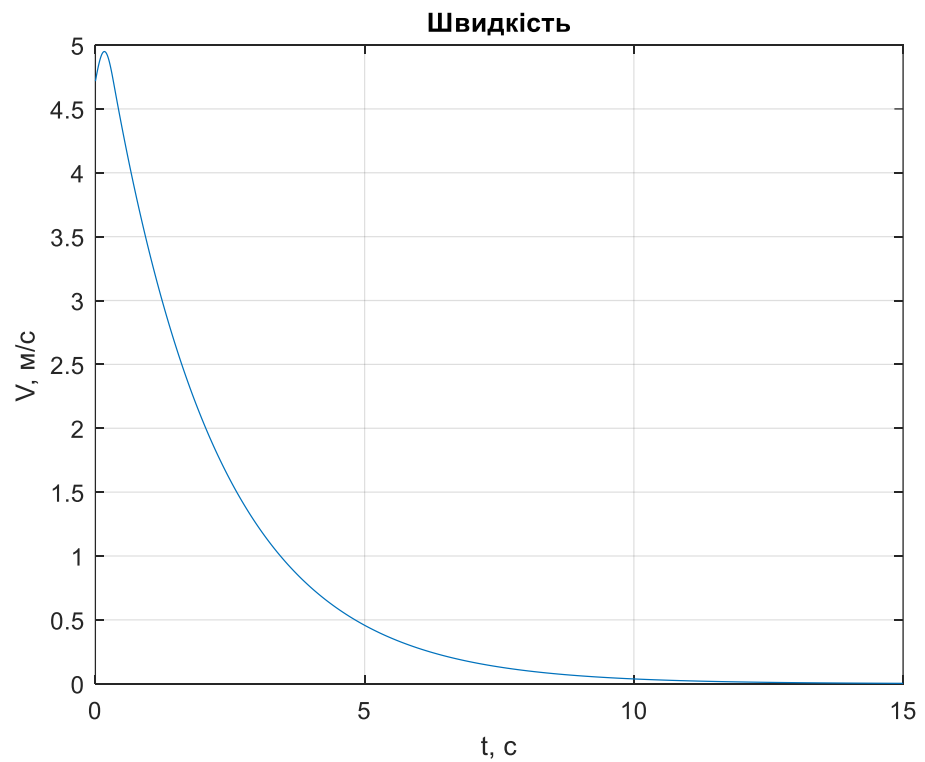


Рисунок 4.4 – Зміна швидкості транспортного засобу з часом, при виконанні алгоритму керування руху до точки при заданих умовах

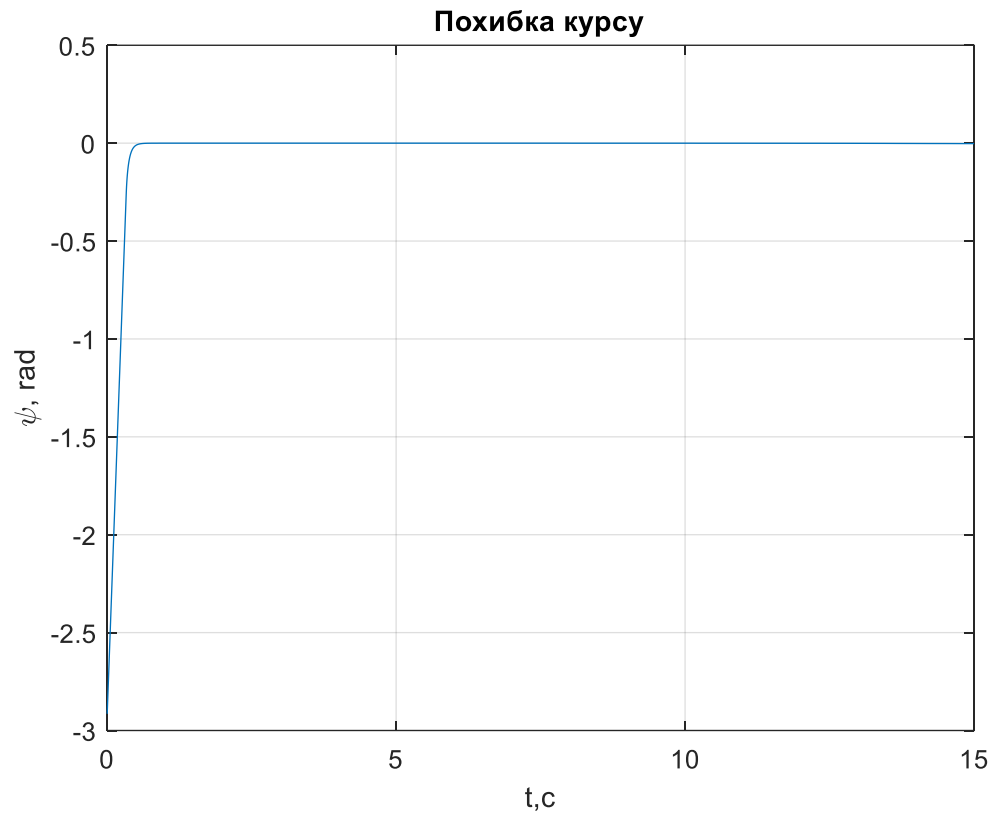


Рисунок 4.5 – Похибка курсу руху транспортного засобу

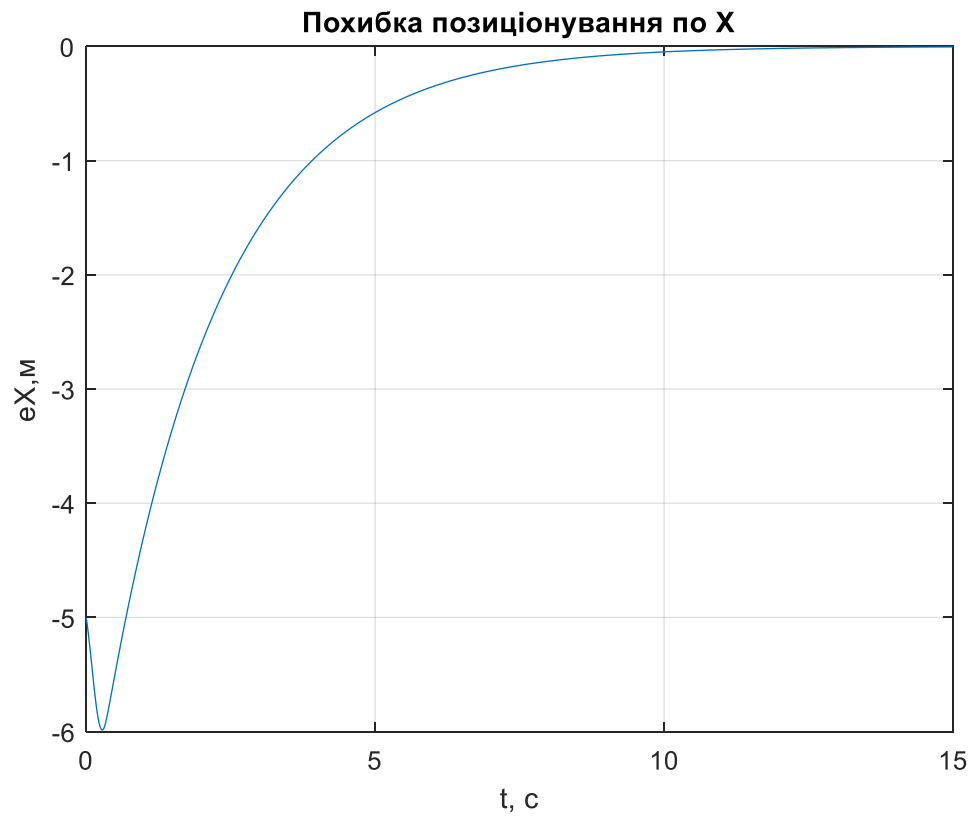


Рисунок 4.6 – Похибка між заданим та реальним значенням X

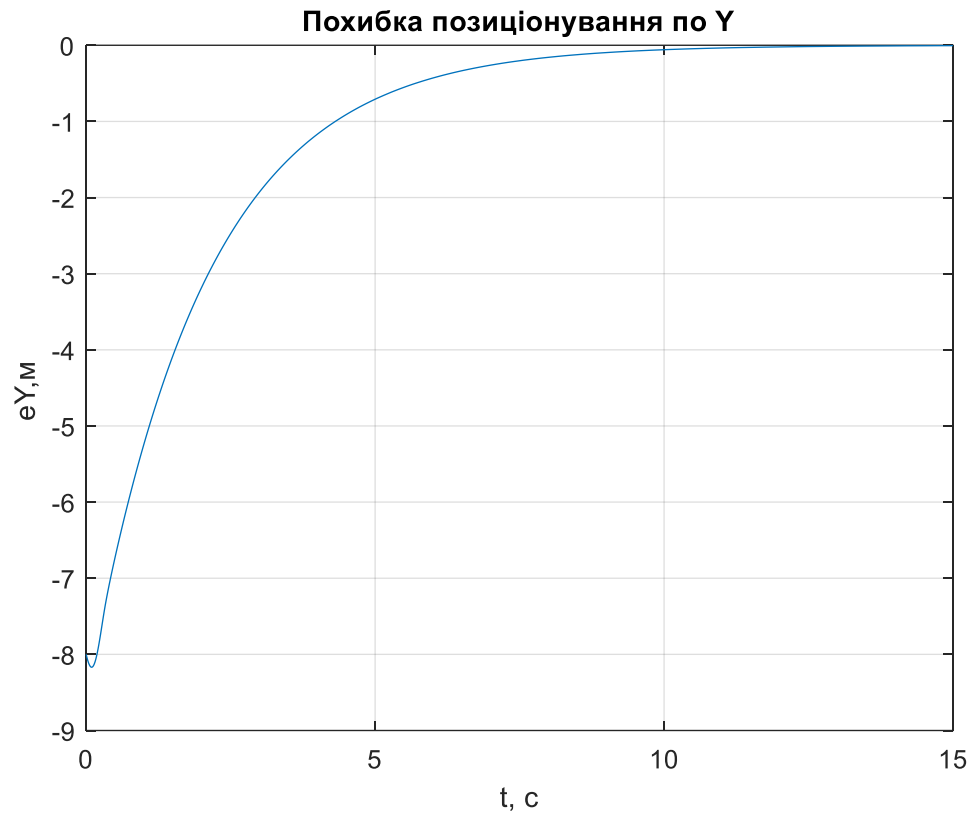


Рисунок 4.7 – Похибка між заданим та реальним значенням Y

Як видно з графіків перехідних процесів, відпрацювання алгоритму руху до заданої точки із заданої позиції відбувається стабільно, транспортний засіб рухається до точки призначення, а похибки приймають допустимі значення.

Для подальшого аналізу даного алгоритму було значно зменшено коефіцієнт регулятора K_h до значення в 0.3. Після чого було отримано наступні результати роботи.

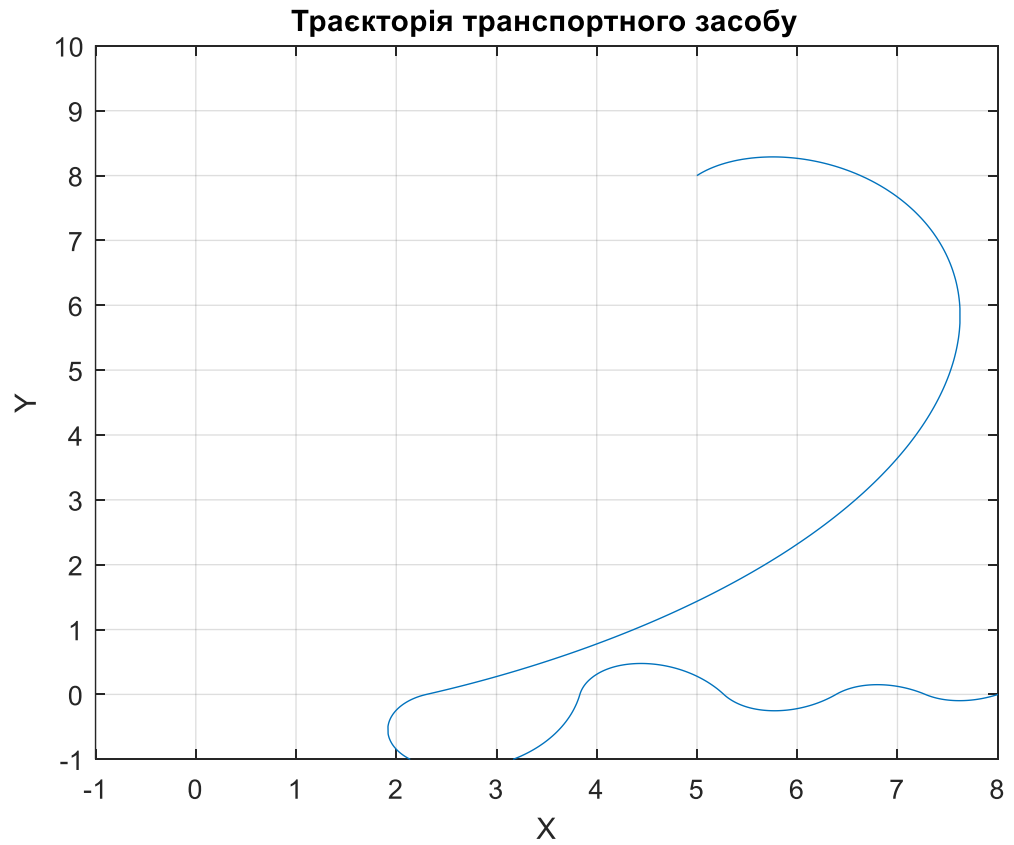


Рисунок 4.8 – Траєкторія руху транспортного засобу на площині

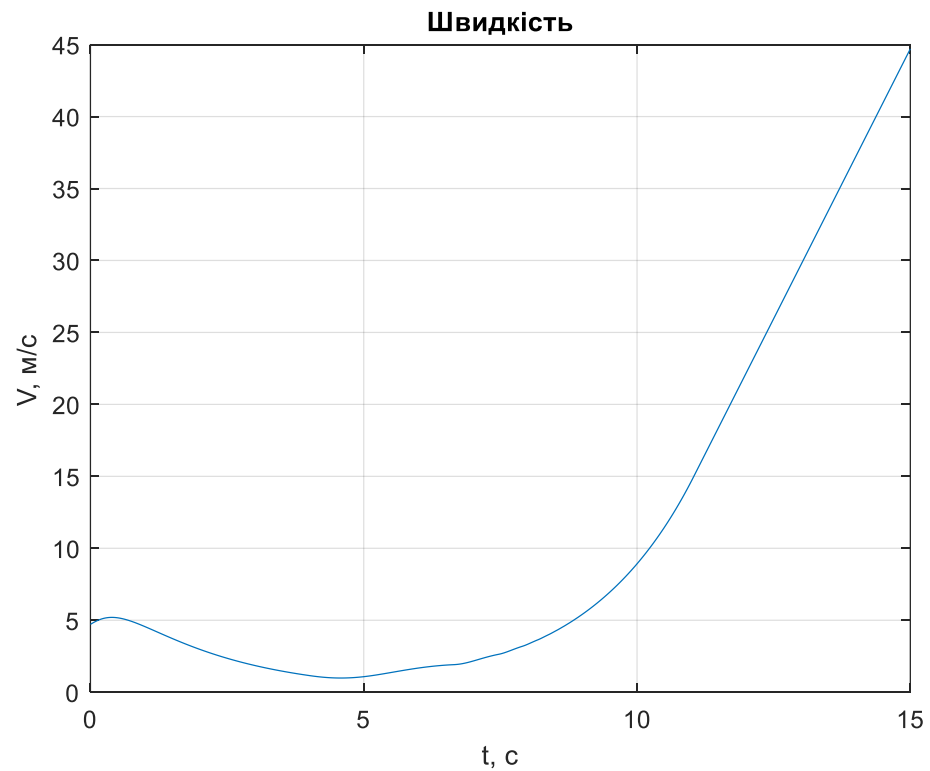


Рисунок 4.9 – Зміна швидкості транспортного засобу з часом, при виконанні алгоритму керування руху до точки при заданих умовах

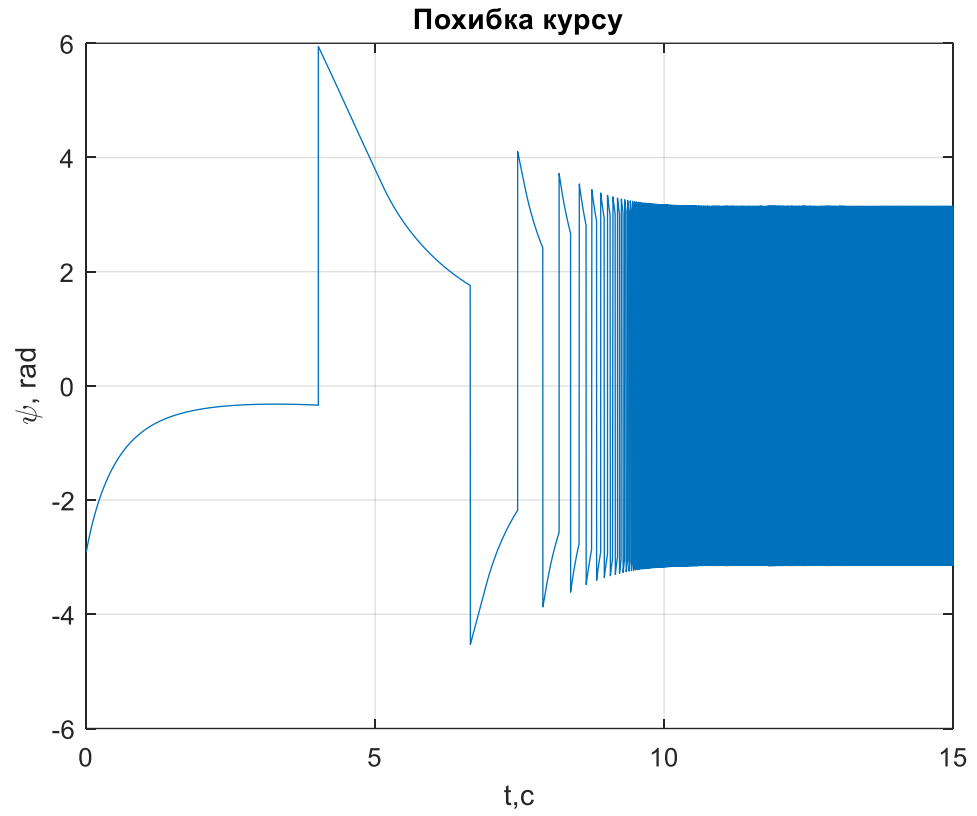


Рисунок 4.10 – Похибка курсу руху транспортного засобу

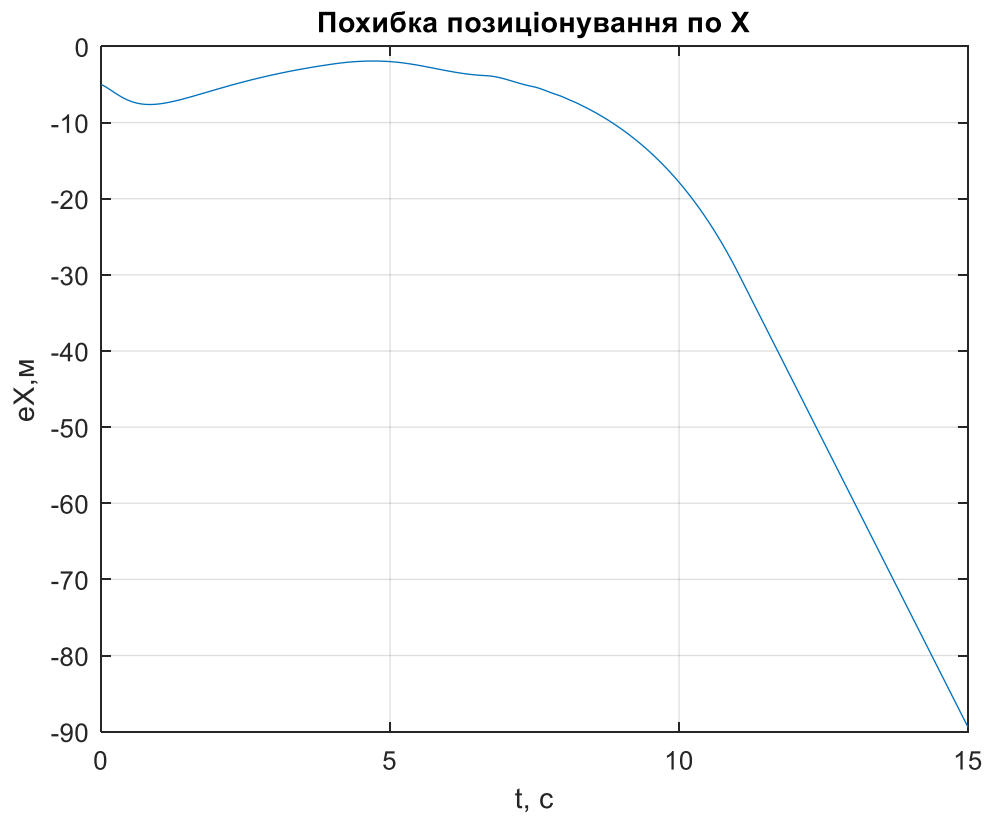


Рисунок 4.11 – Похибка між заданим та реальним значенням X

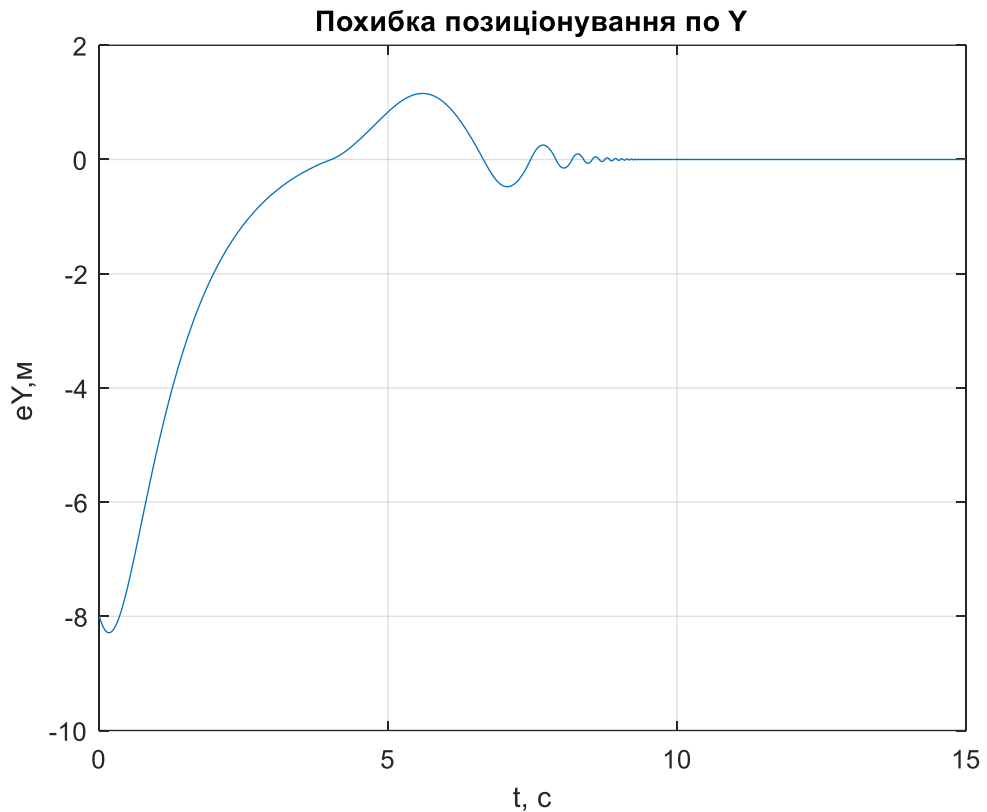


Рисунок 4.12 – Похибка між заданим та реальним значенням Y

Як видно з отриманих графіків перехідних процесів, при коефіцієнті $K_h=0.3$, алгоритм руху автомобіля виконується дуже нестабільно. Транспортний засіб не досягає заданої точки призначення і рухається в протилежному напрямку від неї. Швидкість машини починає постійно зростати, а похибки по курсу та заданих координатах зашкалюють. За винятком координати Y, адже відносно цієї осі транспортний засіб проїхав точніше.

Після виконання цього дослідження коефіцієнт K_h було нормалізовано, тобто поставлено значення 4. Та значно збільшено коефіцієнт K_v . Його значення було встановлено на 3.

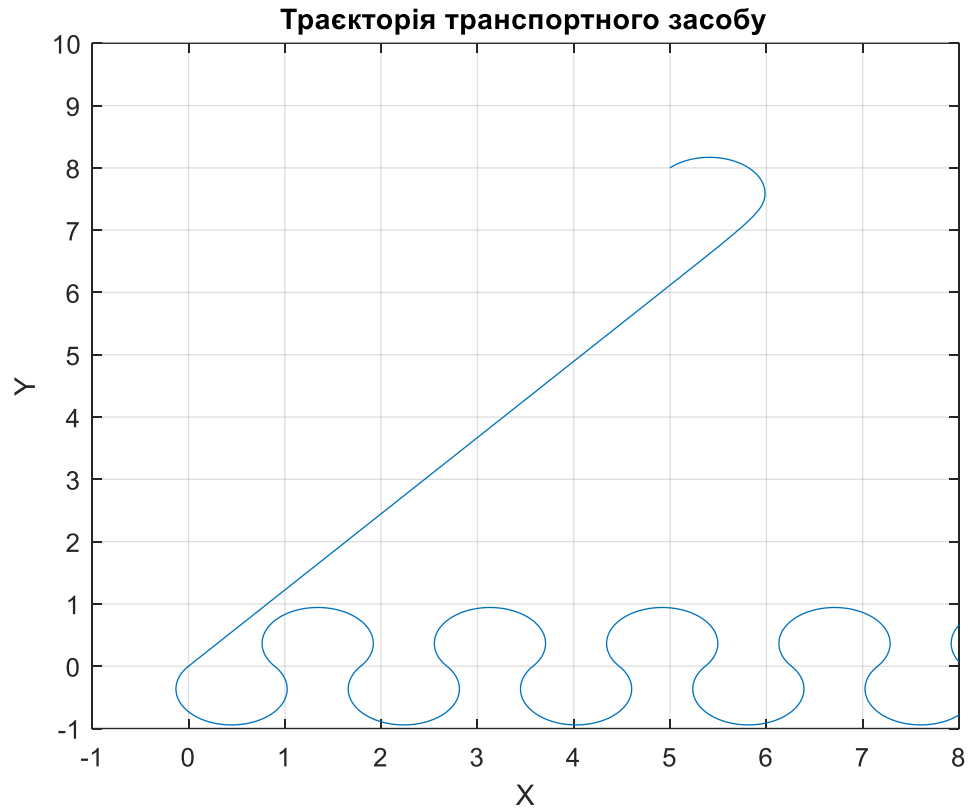


Рисунок 4.13 – Траєкторія руху транспортного засобу на площині

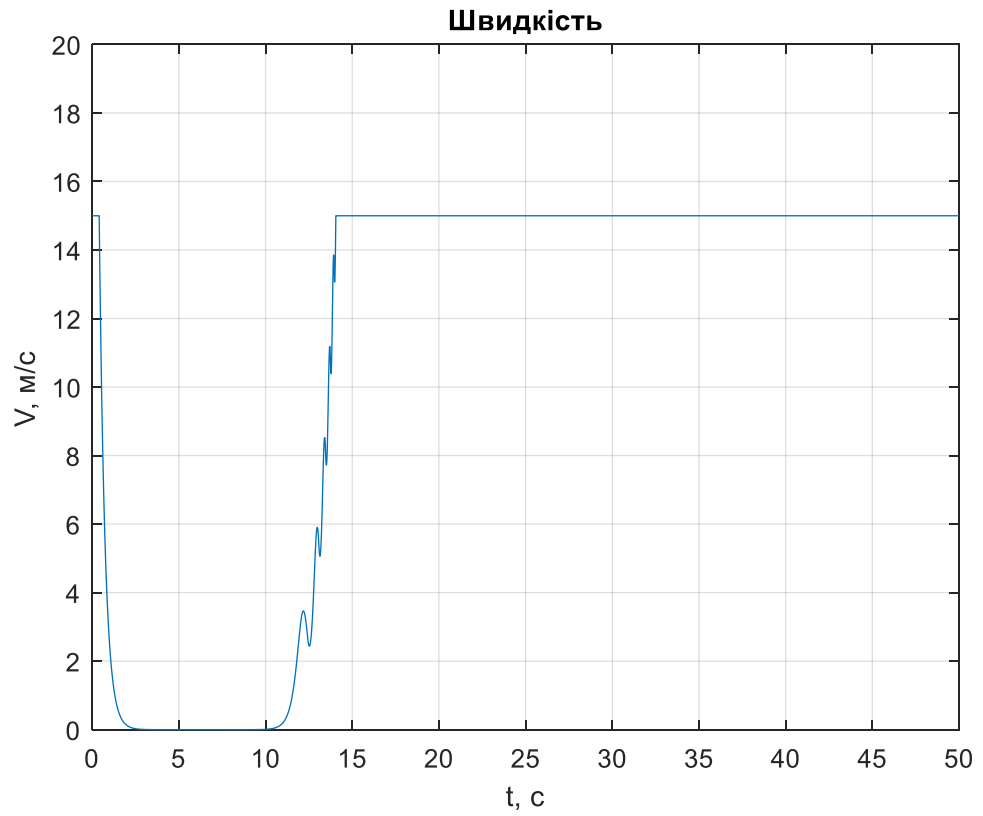


Рисунок 4.14 – Зміна швидкості транспортного засобу з часом, при виконанні алгоритму керування руху до точки при заданих умовах

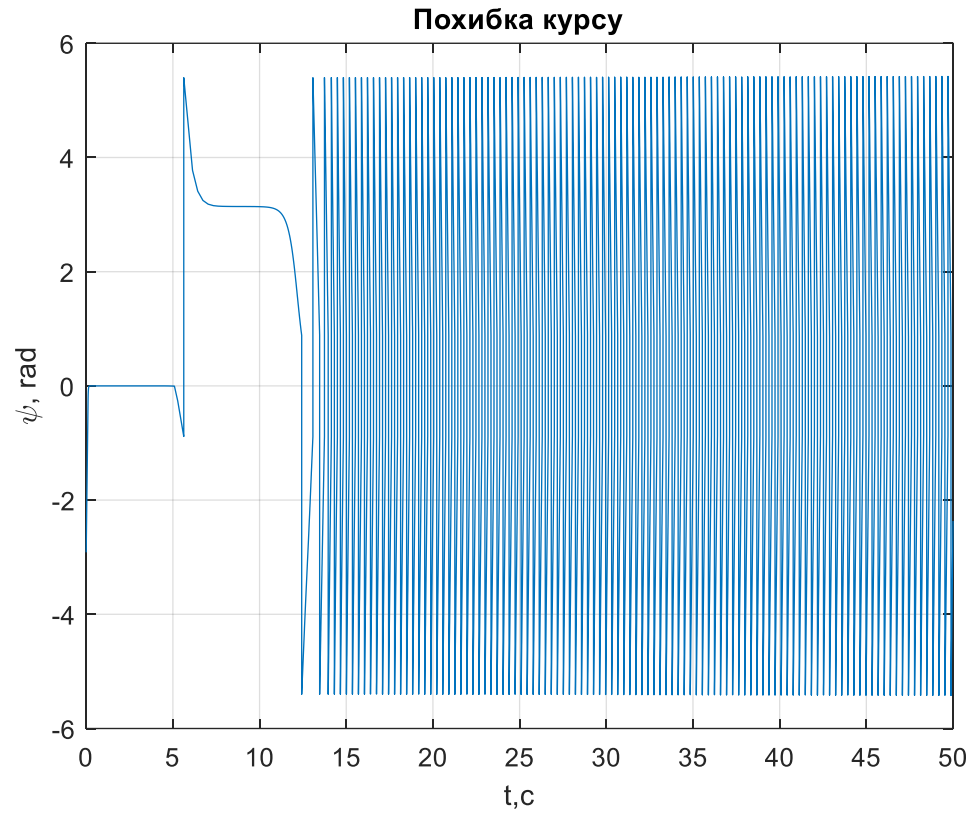


Рисунок 4.15 – Похибка курсу руху транспортного засобу

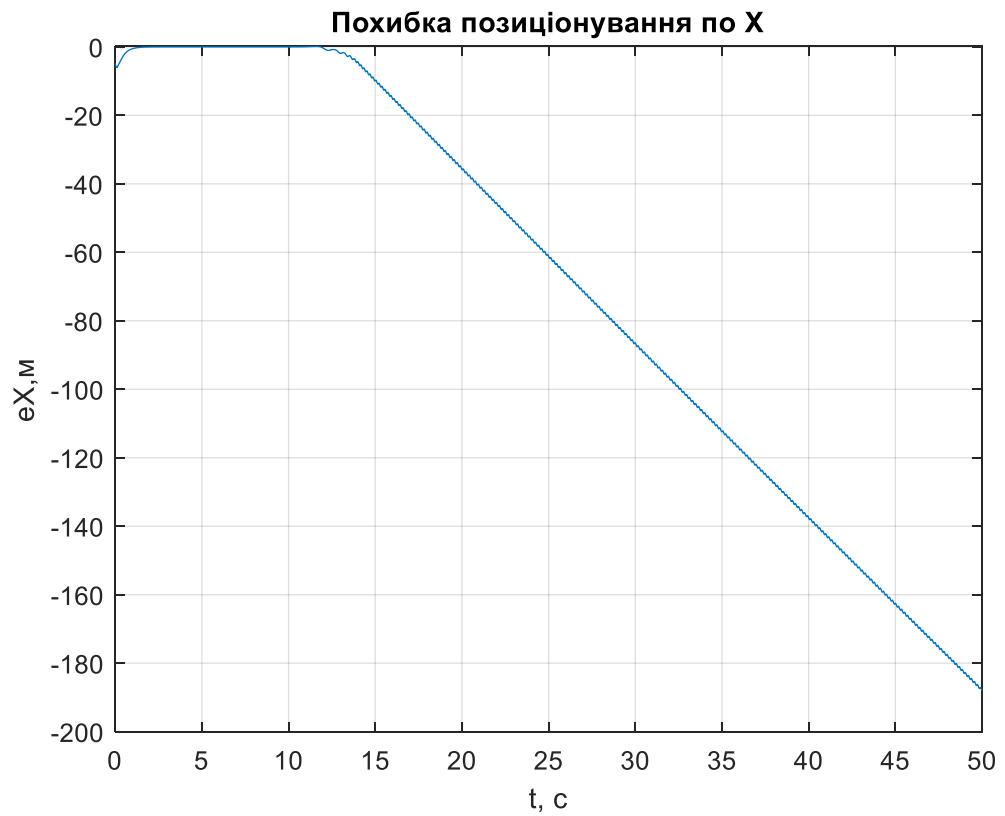


Рисунок 4.16 – Похибка між заданим та реальним значенням X

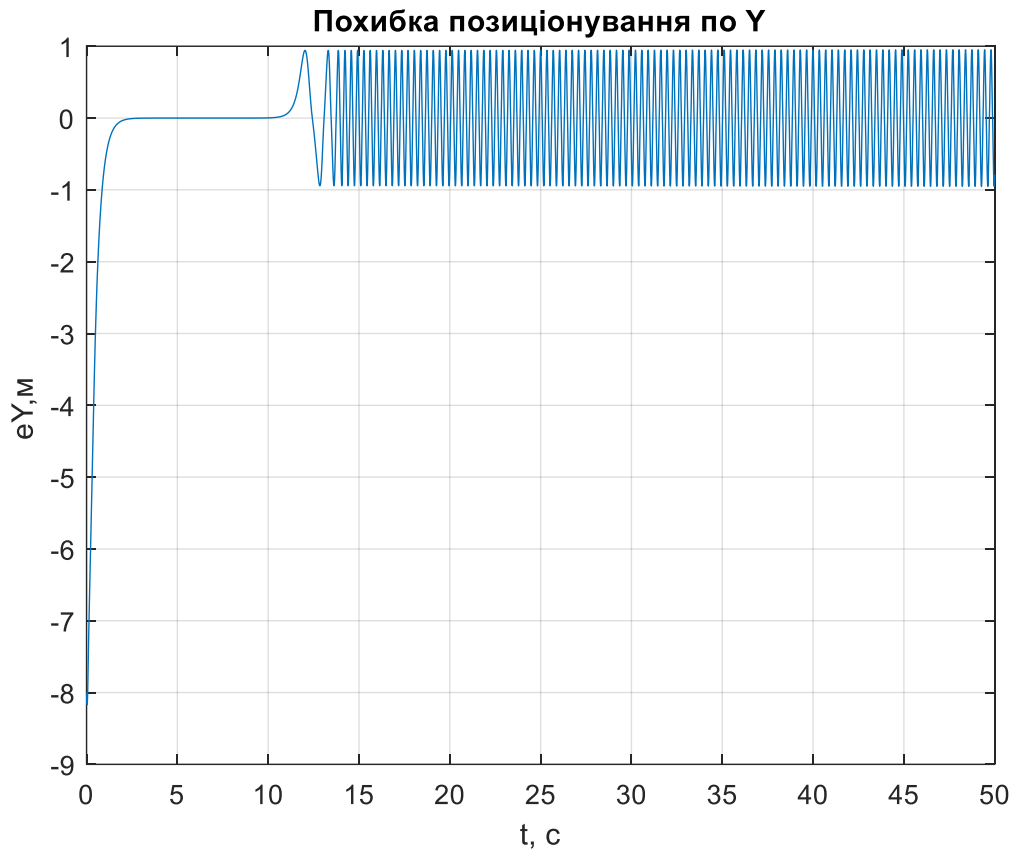


Рисунок 4.17 – Похибка між заданим та реальним значенням Y

За результатами моделювання з підвищеним значенням K_v , можна зробити висновок, що транспортний засіб не відпрацьовує стабільно заданий алгоритм. З траєкторії руху машини видно, що остання проїхала через точку призначення та продовжила рух в невідомому напрямку. Швидкість набирає дуже високі значення за короткий проміжок часу, а похибки набувають коливального характеру, за виключенням похибки між заданим та реальним значенням X , яка в свою чергу має зростаючий характер.

4.2 Алгоритм слідування заданої лінії з заданої позиції

Для виконання даного дослідження було створено математичну модель алгоритму, що має наступний вигляд (див. рис.4.29).

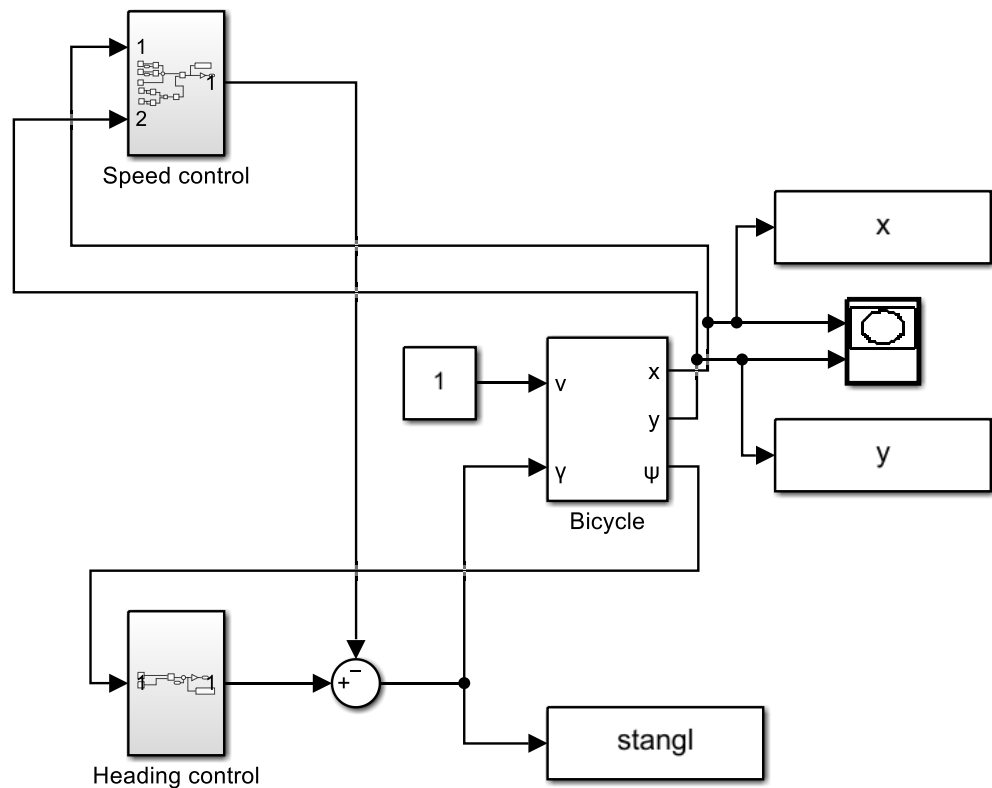


Рисунок 4.18 – Математична модель алгоритму руху слідування лінії

Для правильного відпрацювання алгоритму слідування по лінії потрібно підібрати коефіцієнти регулятора K_d і K_h таким чином, щоб сигнали з блоків «Speed control» та «Heading control» з часом врівноважувалися, для того щоб кут повороту керма прямував до нуля коли транспортний засіб наблизиться до заданої лінії та продовжить рухатись вздовж неї. Дану умову задовольняють наступні значення коефіцієнтів для регуляторів $K_d=0.5$, $K_h=1$.

Лінію по якій слідуватиме розглядуваний транспортний засіб представлено як $a \cdot x + b \cdot y + c = 0$, відповідно до даного рівняння було підібрано наступні значення $a=1$, $b=-2$, $c=4$. Початкова позиція транспортного засобу [5,8], початковий курс $-\frac{\pi}{2}$. За даних умов було проведено перший дослід.

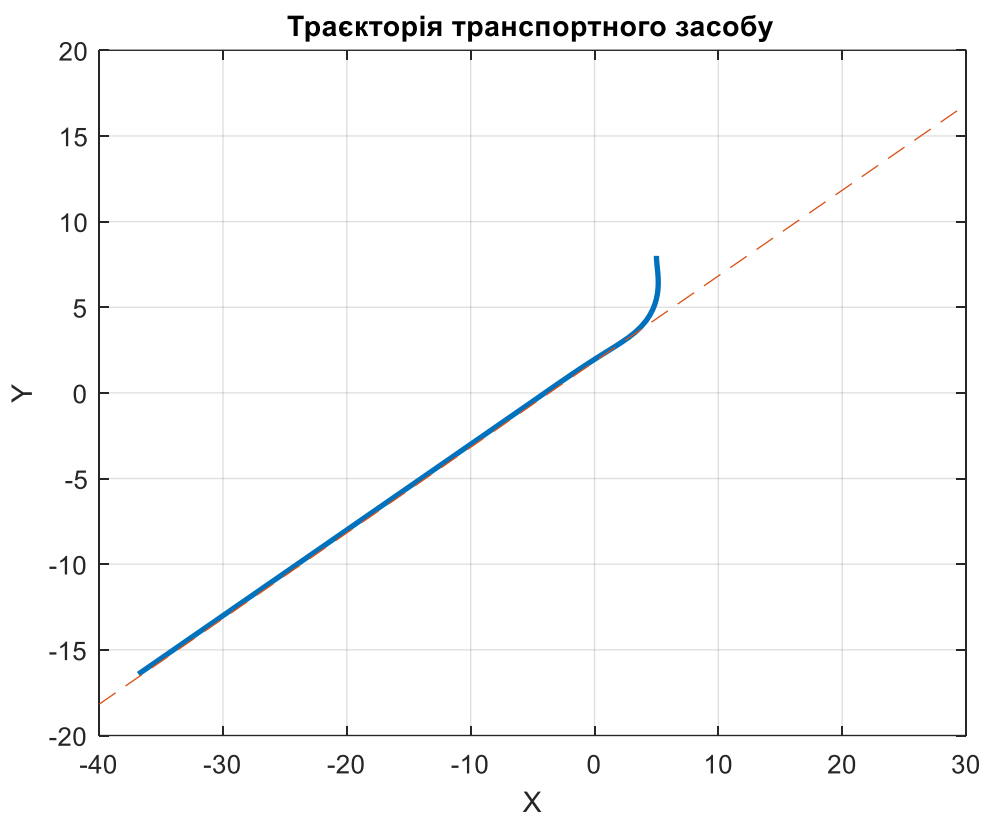


Рисунок 4.19 – Траєкторія руху транспортного засобу відносно заданої лінії

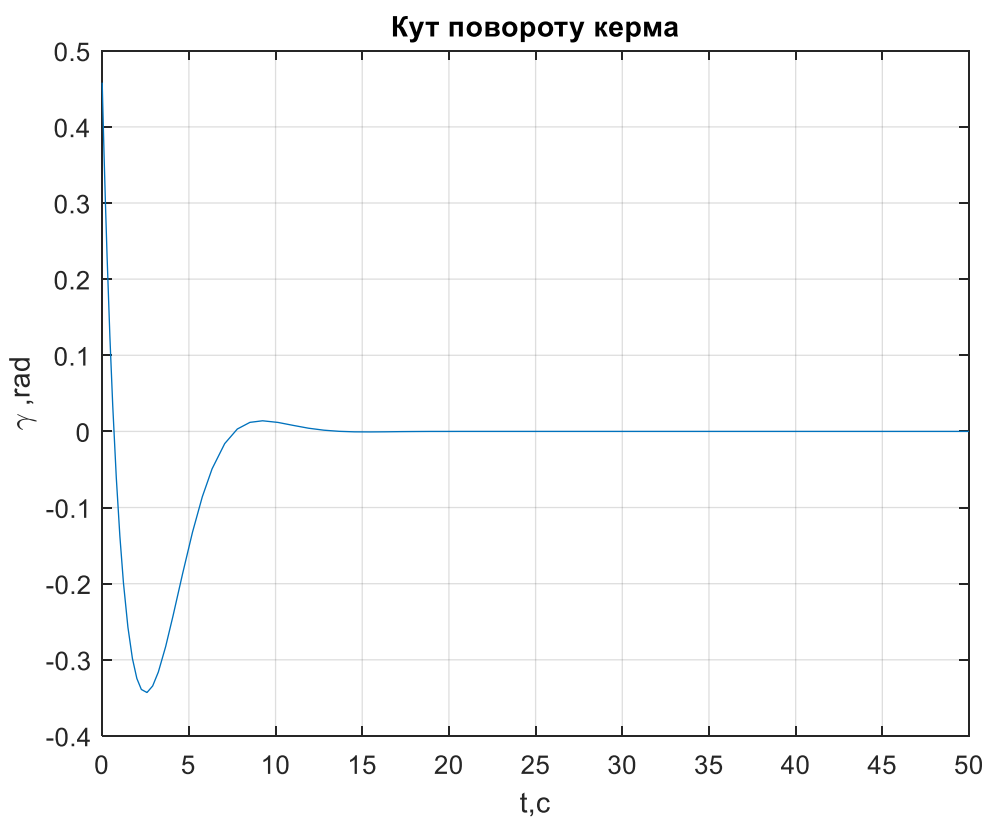


Рисунок 4.20 – Кут повороту керма

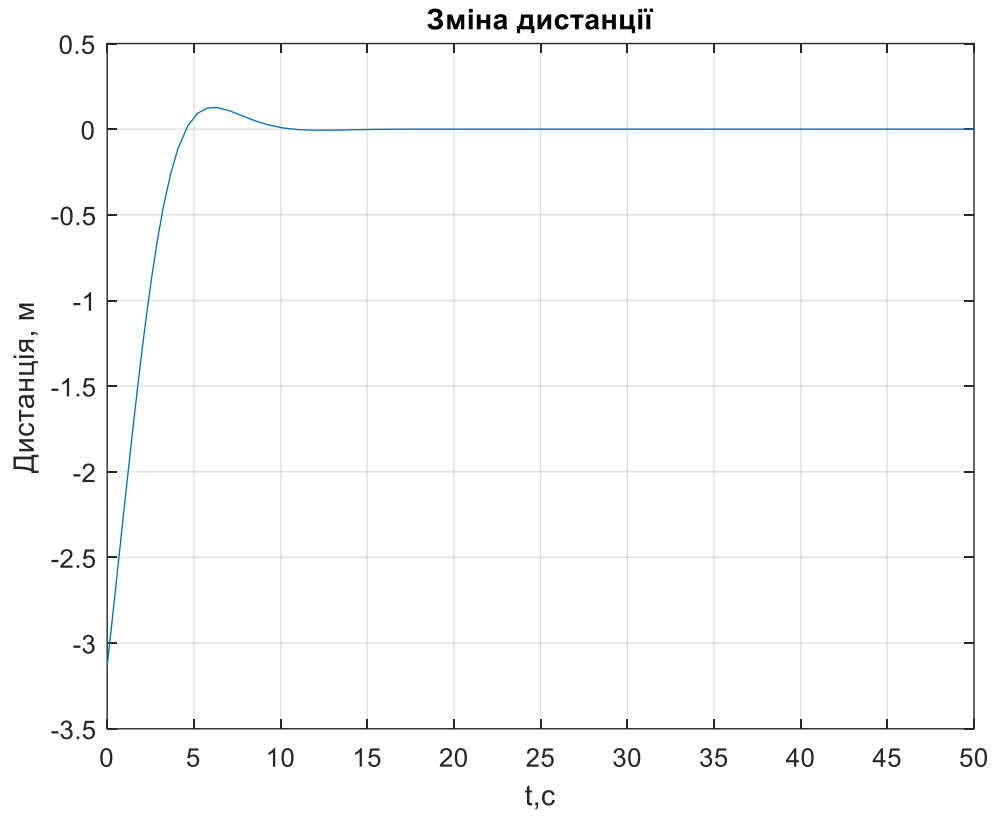


Рисунок 4.21 – Зміна дистанції між заданою лінією та транспортним засобом

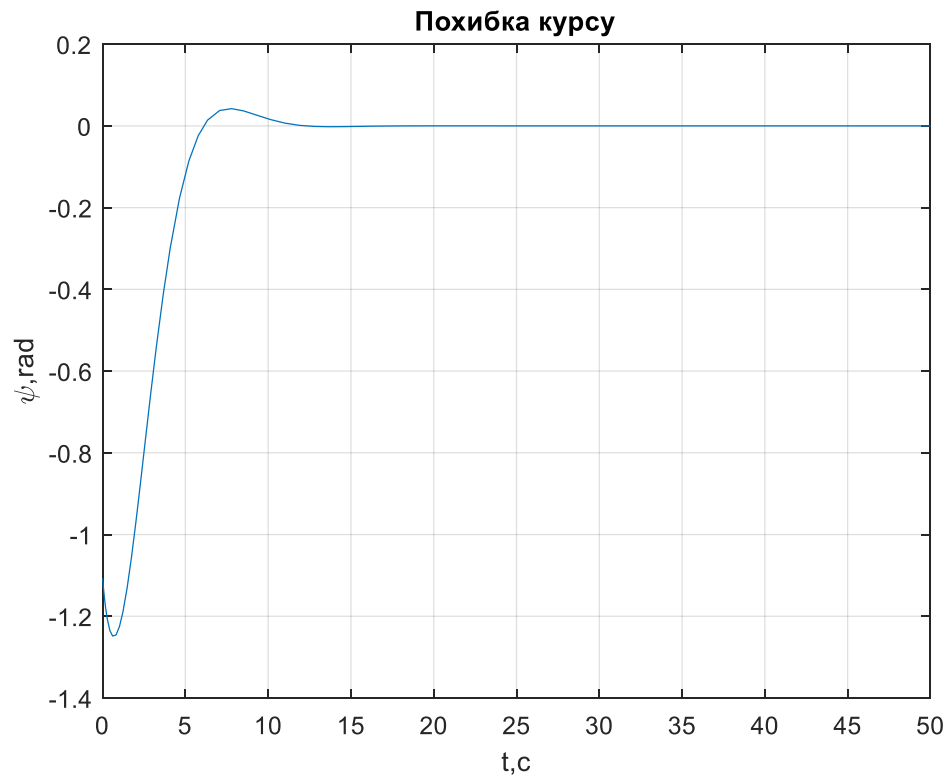


Рисунок 4.22 – Похибка курсу транспортного засобу

Як видно з отриманих графіків, при заданих параметрах, алгоритм виконується стабільно. Траєкторія руху машини наближається до заданої лінії, дистанція між машиною та заданою лінією наближується до нуля. При досягненні лінії колесо перестає повертатись, тому як подальший рух характеризується прямолінійністю. Похибка курсу руху транспортного засобу наближається до нуля.

Далі було прийнято рішення виконати дослід збільшити коефіцієнт регулятора K_d в 10 разів, тобто виставити значення 5. Після цього отримано зображені нижче результати.

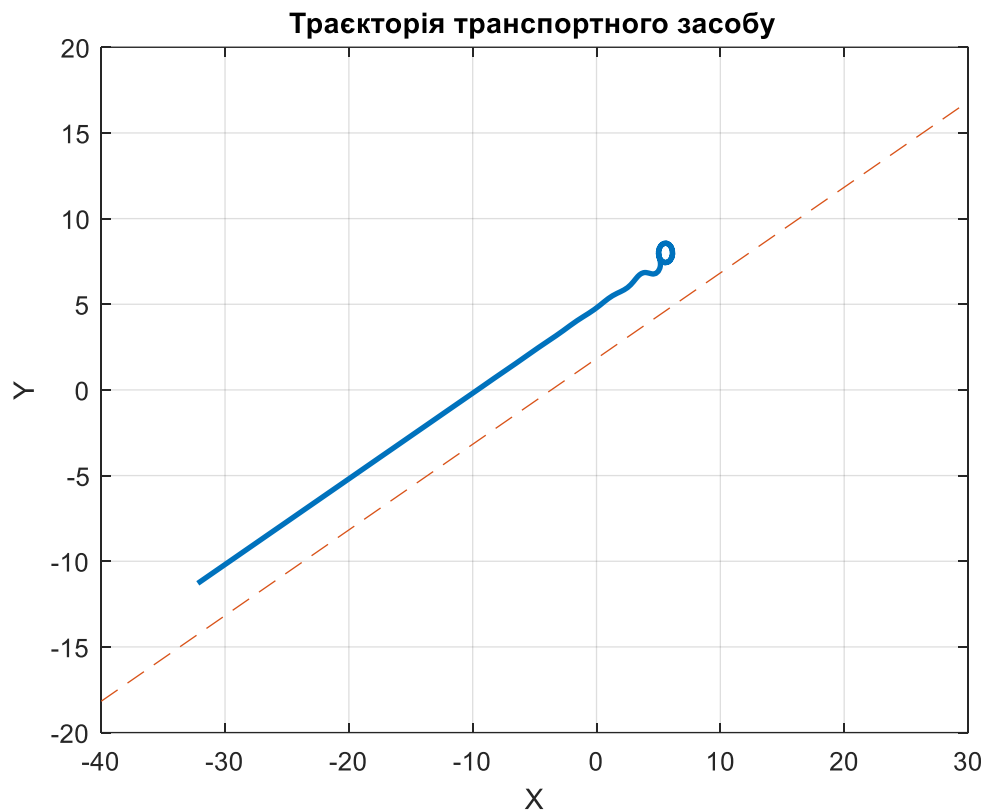


Рисунок 4.23 – Траєкторія руху транспортного засобу відносно заданої лінії



Рисунок 4.24 – Кут повороту керма

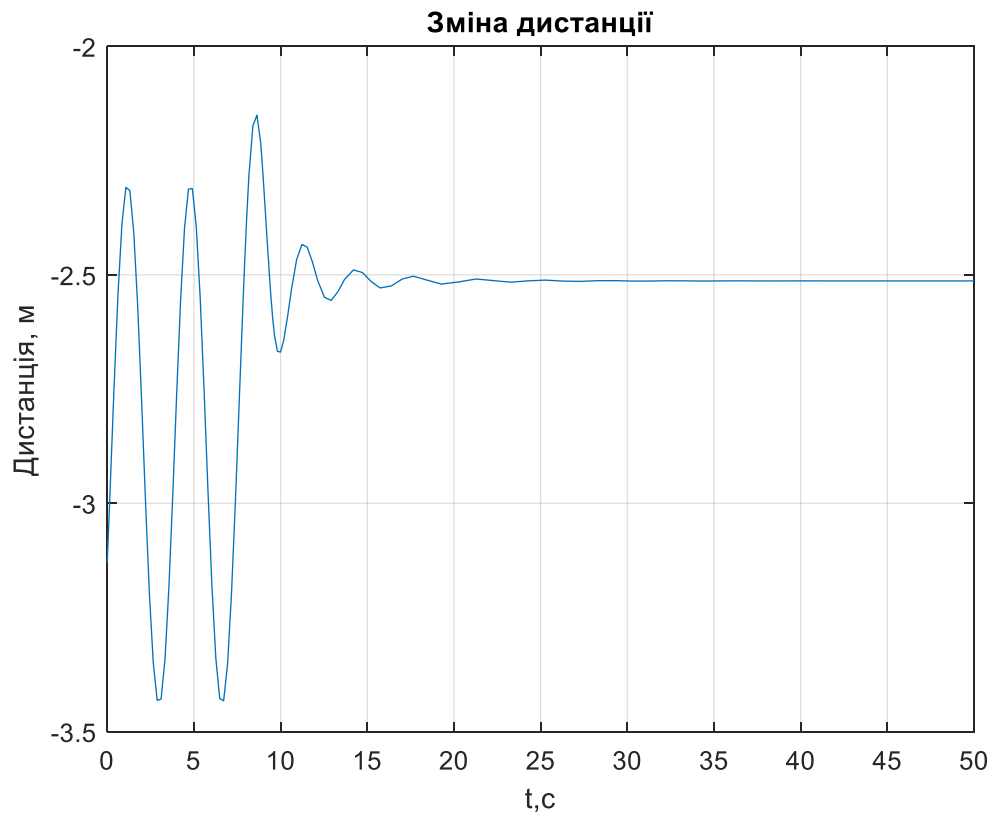


Рисунок 4.25 – Зміна дистанції між заданою лінією та транспортним засобом

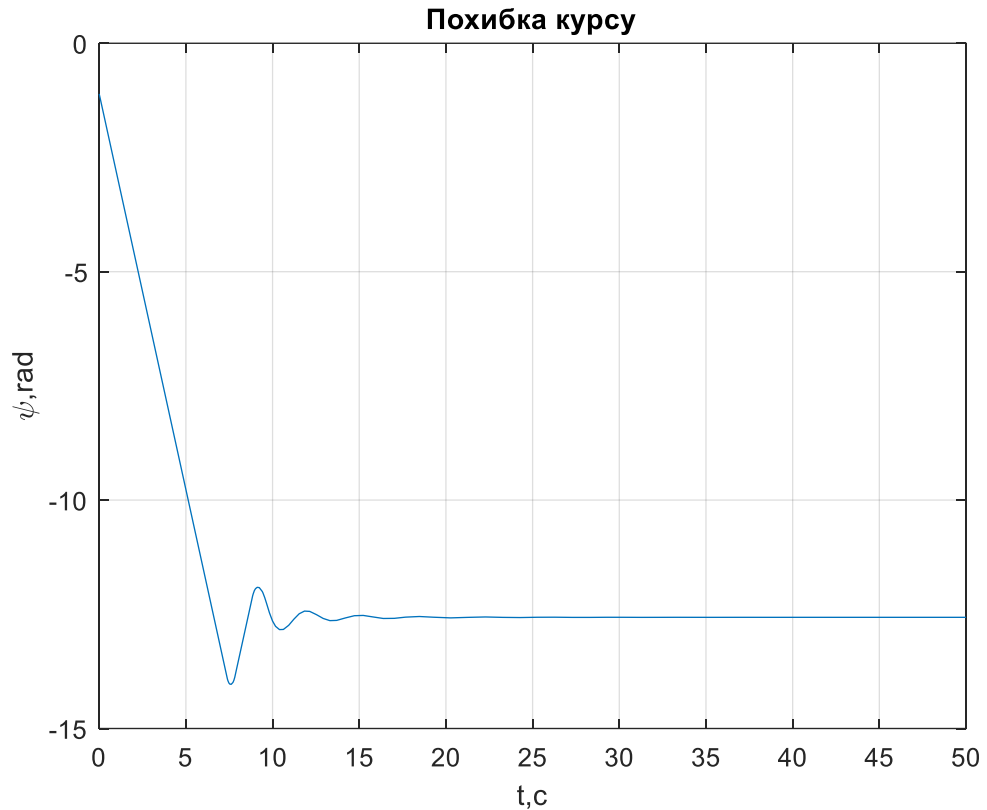


Рисунок 4.26 – Похибка курсу транспортного засобу

Провівши даний дослід можна побачити, що при значному збільшенні коефіцієнта K_d , алгоритм слідування лінії відпрацьовується нестабільно. Транспортний засіб починає рухатись кругом, після чого рухається прямо, паралельно до лінії, але на певній відстані від неї. Похибка курсу руху машини також відмінна від нуля.

Далі коефіцієнт K_d буде нормалізовано, але натомість збільшено в 10 разів коефіцієнт регулятора K_h .

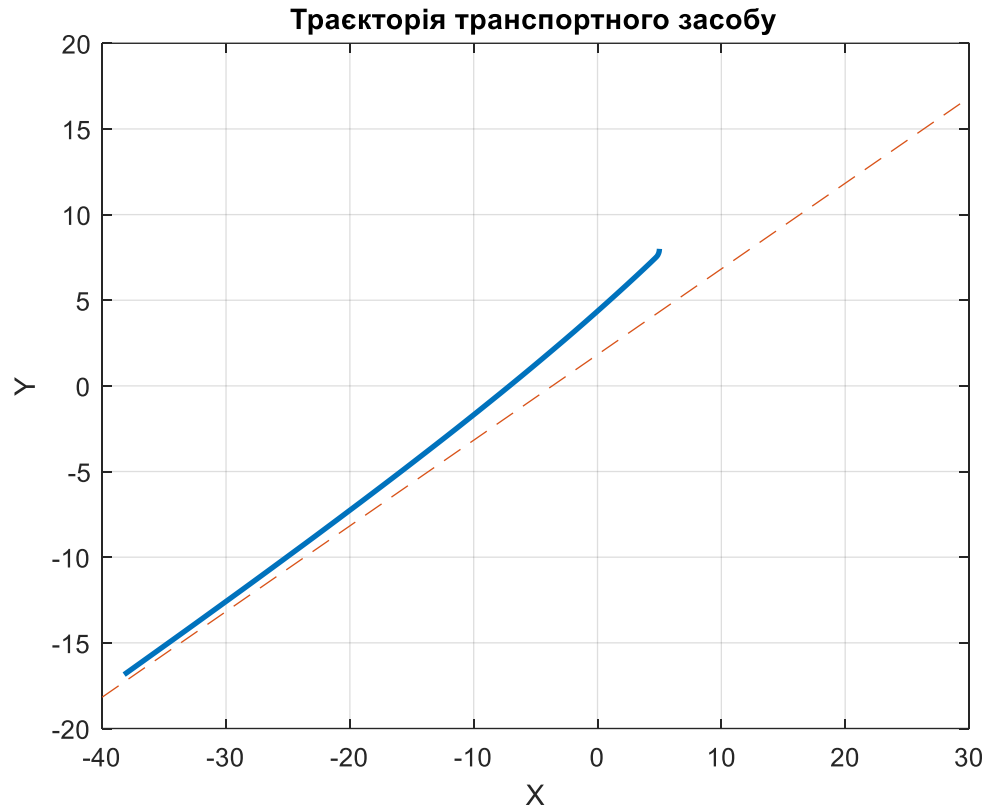


Рисунок 4.27– Траєкторія руху транспортного засобу відносно заданої лінії

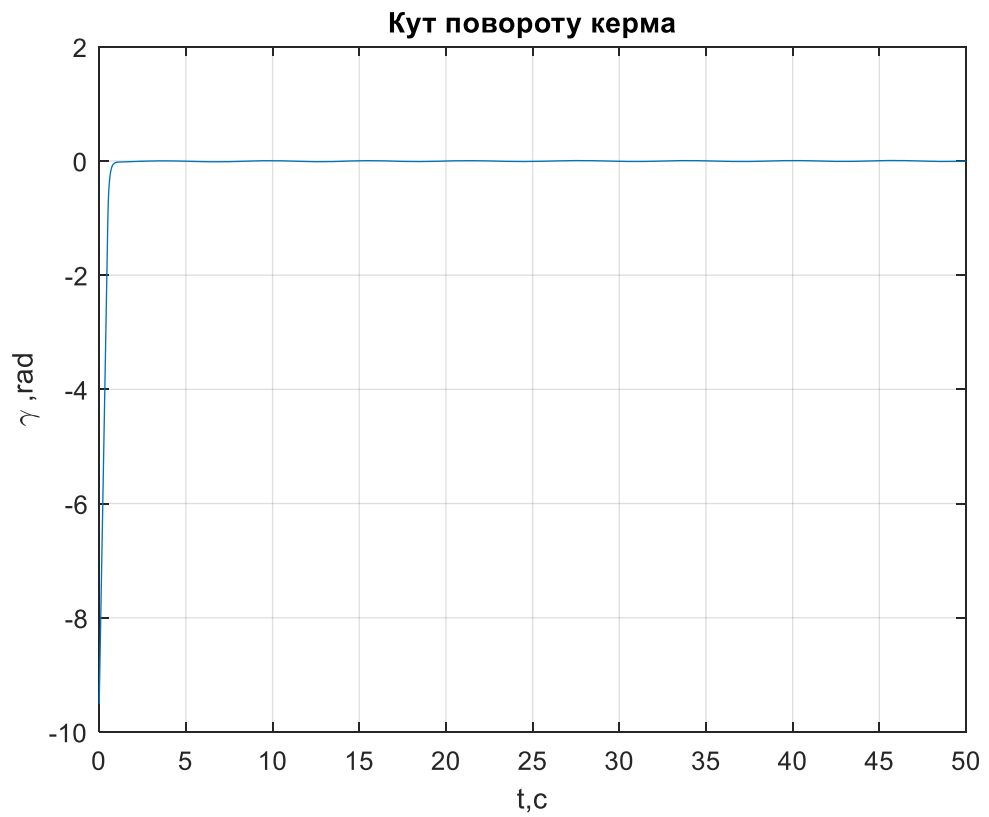


Рисунок 4.28 – Кут повороту керма

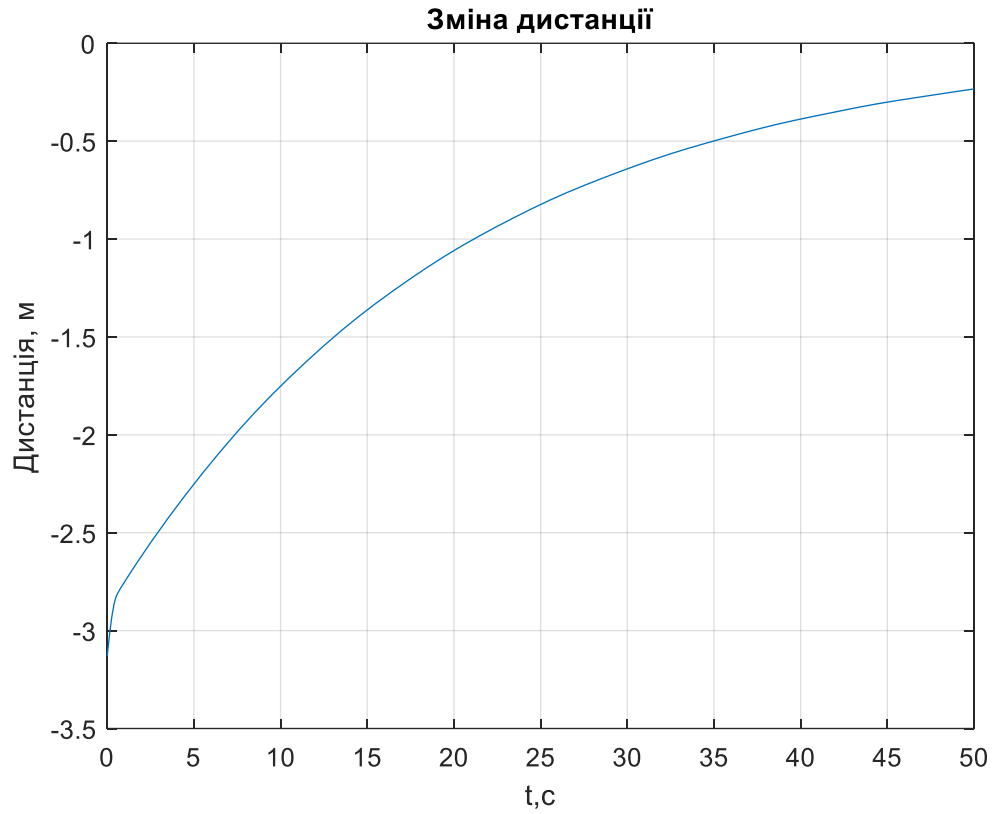


Рисунок 4.29 – Зміна дистанції між заданою лінією та транспортним засобом

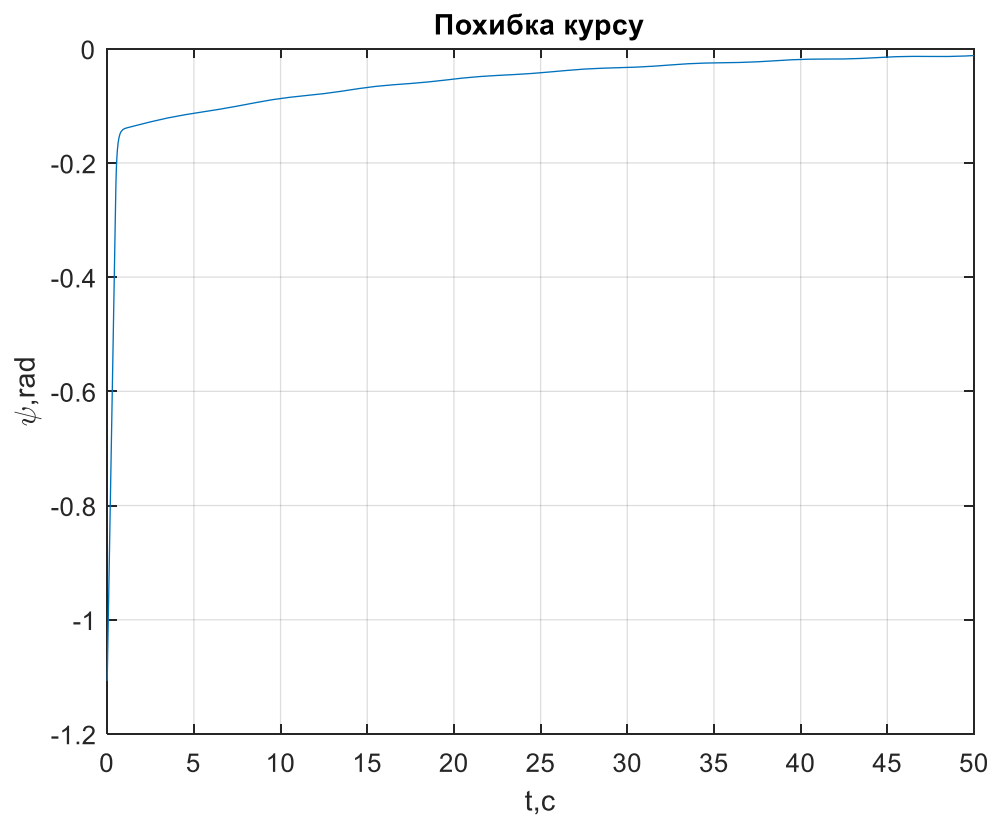


Рисунок 4.30 – Похибка курсу транспортного засобу

Як видно з отриманих результатів моделювання, алгоритм відпрацьовується стабільніше, ніж у минулому досліді, але недостатньо стабільно. Наближення до лінії занадто довге, тобто дистанція змінюється доволі повільно, похибка курсу з часом зменшується.

Висновки до розділу 4

В процесі виконання даного розділу проведено дослідження розглянутих у попередньому розділі алгоритмів керування транспортним засобом, а саме:

- алгоритм руху до заданої точки ;
- слідування заданої лінії.

Дослідження проведено на основі впливу коефіцієнтів регулятора, в залежності від алгоритму це могли бути K_v та K_h в алгоритмі руху до точки та K_d і K_h в алгоритмі слідування лінії. А саме спершу проведено досліди зі значеннями коефіцієнтів, які задовольняли стабільне відпрацювання алгоритмів керування транспортним засобом, після чого досліджувався характер поведінки алгоритму при зміні того чи іншого коефіцієнта.

Після проведення кожного з дослідів було описано характер впливу коефіцієнтів на алгоритм.

5 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ КЕРУВАННЯ ЧЕРЕЗ ROBOT OPERATING SYSTEM (ROS)

5.1 Що таке ROS?

ROS (robot operating system) [12] – це комплекс програмного забезпечення з відкритим кодом (open-source) призначеного для розробки програм для роботів. Тобто це не зовсім операційна система, хоча даний комплекс і здатен на апаратну абстракцію, низькорівневий контроль пристроїв, реалізацію часто використовуваних функціональних можливостей, передачу повідомлень між процесами та керування пакетами.

ROS базується на архітектурі графів, де процес обробки даних відбувається у вузлах, що в свою чергу зв'язуються між собою. Хоча варто наголосити, що вузли реєструються у ведучому ROS, без якого вузли не знали б про існування один одного. Вузли безпосередньо не взаємодіють між собою. Швидше, вони спілкуються через свого роду повідомлення, які публікуються до відповідних тем. Тобто робот виконує різні функції, що не перешкоджають одна одній. Навіть якщо одна з функцій робота виходить з ладу, це не зупиняє функціонування всього робота.

ROS реалізує декілька різних стилів зв'язку, таких як синхронний зв'язок у стилі RPC через служби, асинхронне потокове передавання даних за темами та зберігання даних на сервері.

5.2 Переваги використання ROS

Використання[11] комплексу програмного забезпечення ROS, загалом призначено для вирішення доволі специфічних задач при розробці різного роду роботів. Адже використання ROS несе в собі ряд переваг, таких як:

- Одноранговість. Система, побудована з використанням ROS, складається з ряду процесів, потенційно на безлічі різних хостів, з'єднаних під час виконання в одноранговій топології. Хоча фреймворки, засновані на центральному сервері, можуть реалізувати переваги багатопроцесорної та багатохостингової структури, при з'єднанні комп'ютерів у неоднорідну мережу на центральному сервері даних виникають проблеми.

— Використання декількох мов програмування. Під час написання коду багато людей мають переваги щодо одних мов програмування над іншими. Ці уподобання є результатом особистих компромісів між часом програмування, простотою налагодження, синтаксисом, ефективністю виконання та низкою інших причин, як технічних, так і культурних. На даний момент ROS підтримує чотири дуже різні мови: C++, Python, Octave та LISP.

— Базованість на інструментах. У спробі впоратися зі складністю ROS, було зроблено вибір на користь використання структури мікроядра, де велика кількість інструментів використовується для побудови та запуску різних компонентів ROS, а не для побудови монолітного середовища розробки та виконання. Ці інструменти виконують різні завдання, наприклад, здійснюють навігацію по дереву вихідного коду, отримують і встановлюють параметри конфігурації, візуалізують топологію однорангового з'єднання, вимірюють використання пропускну здатності, графічно складають дані повідомлень, автоматично генерують документацію тощо.

— Безкоштовність та відкритий код. ROS розповсюджується на умовах ліцензії BSD, що дозволяє розробляти як некомерційні, так і комерційні проекти. ROS передає дані між модулями, використовуючи міжпроцесний зв'язок, і не вимагає, щоб модулі з'єднувались в одному виконуваному файлі. Таким чином, системи, побудовані навколо ROS, можуть використовувати роздрібне ліцензування своїх різних компонентів: окремі модулі можуть включати програмне забезпечення, захищене різними ліцензіями, починаючи від GPL і закінчуючи BSD.

5.3 Основні поняття

Основними поняттями реалізації ROS[11] є вузли, повідомлення, теми та сервіси.

Вузли - це процеси, які виконують обчислення. ROS розроблений таким чином, щоб бути модульним: система, як правило, складається з багатьох вузлів. У цьому контексті термін «вузол» є взаємозамінним із «програмним модулем». Використання терміну «вузол» виникає внаслідок

візуалізації ROS-систем під час виконання: коли працює багато вузлів, зручно відображати однорангові комунікації як граф, з процесами у вигляді графічних вузлів, а однорангові посилання як дуги.

Вузли спілкуються між собою, передаючи повідомлення.

Повідомлення - це строго набрана структура даних. Підтримуються стандартні примітивні типи (ціле число, плаваюча крапка, булеве значення тощо), а також масиви примітивних типів та констант. Повідомлення можуть складатися з інших повідомлень та масивів інших повідомлень.

Вузол надсилає повідомлення, публікуючи його до заданої теми, що є просто рядком, таким як "одометрія" або "карта". А вузол, який зацікавлений у певному виді даних, підпишеться до відповідної теми. Може бути кілька одночасних «видавців» та «підписників» для однієї теми, також один вузол може публікувати та / або читати кілька тем.

Хоча тематична модель публікації та підписки є гнучкою парадигмою комунікацій, її схема маршрутизації не підходить для синхронного передавання даних, що може спростити структуру деяких вузлів. У ROS це називається службою(сервісом), що визначається назвою рядка та парою строго набраних повідомлень: одне для запиту та одне для відповіді. Це аналогічно веб-службам, які визначаються URL і мають документи запитів та відповідей чітко визначених типів. На відміну від тем, лише один вузол може рекламувати сервіс з будь-яким певним іменем: наприклад, може існувати лише одна служба, що називається «класифікувати зображення», так само, як може бути лише одна веб-служба на будь-якому даному URL.

5.4 Інструменти

ROS, окрім свого основного функціоналу, доповнений різного роду інструментами, що в свою чергу дозволяють користувачам візуалізувати результати роботи, полегшують орієнтування в структурах ROS, а також дають можливість створювати різноманітні сценарії для автоматизації складних процесів конфігурації та налаштування. Вони (інструменти) значно збільшують можливості розробників при використанні пакета ROS, через

різного роду спрощення та рішення загальних проблем, що виникають у робототехніці.

Загалом використовуються саме такі інструменти:

- `rviz` тривимірний візуалізатор, який зображує роботів, їх середовище роботи, а також дані датчиків. Має велику кількість плагінів та типів візуалізації.
- `rosviz` інструмент для командного рядка, що записує та відтворює дані повідомлення ROS. Реєструє повідомлення, що надходять, при прослуховуванні тем.
- `catkin` збірка ROS, заснована на CMake.
- `roscpp` пакет набору інструментів для розширення функціоналу оболонки `bash`, до якого входять `roscpp`, `roscd`, `roscp`. Дані інструменти дозволяють не використовувати шлях до файлу, де власне і знаходиться пакет, а використовувати тільки імена пакетів. Також сюди входять `roscd` та `roscp`.
- `roslaunch` використовується для запуску декількох вузлів та для встановлення параметрів на сервері параметрів. Записані за допомогою XML файли автоматизують процес запуску та конфігурації, спрощуючи його до однієї команди.

5.5 Вибір дистрибутиву

Список можливих дистрибутивів можна знайти на ros.org, у вкладці дистрибутивів для ROS2. На цій сторінці можна знайти як підтримувані дистрибутиви, так і ті, що вже перестали підтримуватись.

Для того щоб обрати дистрибутив варто перевірити EOL (end of life) date, тобто дату коли перестане підтримуватись даний дистрибутив.

ROS2 доступний для таких платформ як Linux, MacOS та Windows.

5.6 Інсталяція

Розглянуто процес інсталяції ROS2 Foxu Fitzroy на операційну систему Linux, а саме на її дистрибутив Ubuntu.

Для того щоб інсталиювати ROS2 Foxy Fitzroy на Ubuntu, потрібно на сторінці ros.org обрати пункт “Debian packages”, де буде представлено повний покроковий алгоритм інсталяції, що проводитиметься в терміналі.

Спочатку потрібну налаштувати кодування на UTF-8.

Наступним кроком буде додавання репозиторію ROS2. Для цього потрібно авторизувати GPG-ключ, після чого додати репозиторій до списку джерел.

Перед інсталяцією пакетів ROS2 потрібно оновити кеш налаштування репозиторію. Після цього можна переходити до безпосередньої інсталяції. Бажано встановити Desktop-версію через наявність інструментів та демонстраційних програм, що помітно полегшить роботу початківцю.

Необов'язковим, але бажаним кроком буде інсталяція інструменту `argcomplete`.

5.7 Налаштування

Для подальшої роботи з ROS2 потрібно виконати налаштування шляху до дистрибутиву через скрипт. Для цього потрібно зайти в директорію інсталюваного дистрибутиву та обрати файл `setup.bash`. Після чого в терміналі потрібно прописати “`source setup.bash`”. Для того щоб не виконувати дану процедуру перед кожним запуском ROS2, потрібно відредагувати файл `.bashrc`, використовуючи `gedit`. Тобто потрібно в терміналі прописати наступну команду: “`gedit ~/.bashrc`”. Відкриється файл, в самому кінці якого потрібно буде прописати наступне: “`source opt/ros/foxy/setup.bash`”.

5.8 Запуск програми в ROS

Для запуску програми в терміналі потрібно прописати `ros2 run` “назва програми”. Для прикладу можна розглянути процес спілкування вузла-видавця та вузла-підписника, використавши демонстраційні програми “`talker`” та “`listener`”. Для цього потрібно відкрити принаймні два вікна терміналу. В одному прописати “`ros2 run demo_nodes_cpp talker`”, в іншому - “`ros2 run demo_nodes_cpp listener`”. Програма `talker` виводить у термінал

фразу “Hello world: n”, де n – номер виведеної строки тексту, тобто такий собі лічильник. А listener виводитиме те, що він “почув” від програми talker з моменту свого запуску.

5.9 Створення власної програми в ROS2

Для того щоб створювати власні програми потрібно інсталиювати інструмент Colcon. Для чого в терміналі потрібно прописати наступні команди: “sudo apt install python3-colcon-co”; “sudo apt install python3-colcon-common-extensions”.

Для файлу colcon-argcomplete.bash потрібно виконати аналогічну операцію, що і для setup.bash при налаштуванні ROS2.

Потрібно створити робочий простір через команду mkdir, до якої потрібно прописати назву папки робочого простору. В папці робочого простору потрібно створити папку src. Для того щоб створену раніше папку зробити робочим простором потрібно прописати команду “colcon build”. Після цього в папці робочого простору утвориться ще три папки: build, install та log.

Для подальшої роботи потрібно вказати шлях до файлу local_setup.bash в папці install за допомогою команди source. Після чого можна працювати з даними, створеними в робочому просторі.

5.10 З MatLab до ROS

За допомогою середовища MatLab, як було вияснено в попередньому пункті, можна створювати різноманітні алгоритми керування транспортним засобом, та не тільки для моделювання. За допомогою ROS Toolbox можна впроваджувати створені алгоритми безпосередньо до середовища ROS.

ROS Toolbox[13] надає інтерфейс, що з'єднує MATLAB і Simulink з ROS, що дозволяє створити мережу ROS-вузлів. Набір інструментів включає функції MATLAB та блоки Simulink для імпорту, аналізу та відтворення даних ROS, записаних у файлах rosbag. Також можна підключитися до діючої ROS-мережі, щоб отримати доступ до ROS-повідомлень.

Набір інструментів дозволяє перевірити ROS-вузли за допомогою моделювання на робочому столі та підключення до зовнішніх симуляторів роботів, таких як Gazebo. ROS Toolbox підтримує генерацію коду на C ++ (з MATLAB Coder і Simulink Coder), що дозволяє автоматично генерувати ROS-вузли з моделі Simulink і розгортати їх на імітованому або фізичному обладнанні. Підтримка зовнішнього режиму Simulink дозволяє переглядати повідомлення та змінювати параметри, коли створена модель алгоритму керування працює на апаратному забезпеченні.

5.11 Проведення експерименту на реальному прототипі

Для подальшого дослідження алгоритмів проведено експеримент на реальному робочому прототипі.



Рисунок 5.1 – Прототип автономної машини, на якій досліджено алгоритми

За допомогою ROS та ROS Toolbox програми алгоритмів було записано на Raspberry pi, що виступала основним контролером.

Для цього було створено модель в робочому просторі MatLab Simulink, що має наступний вигляд

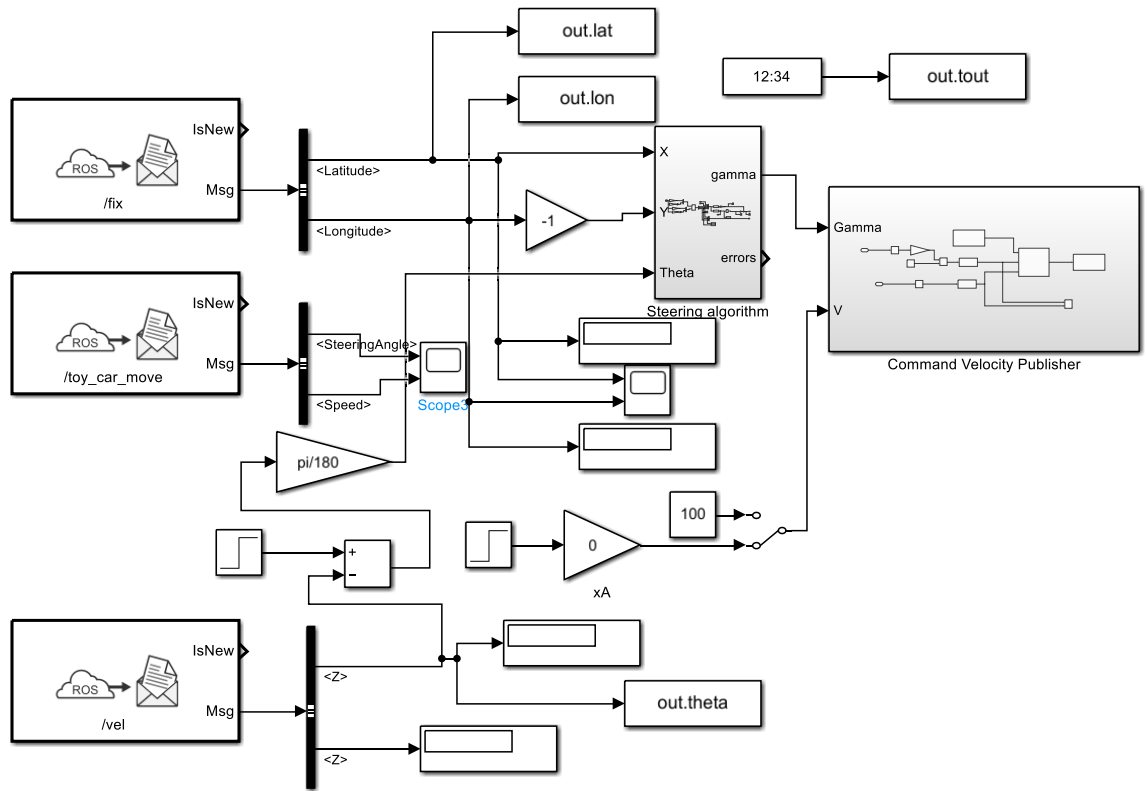


Рисунок 5.2 – Модель алгоритмів керування створена для експерименту

Як видно з рисунку моделі, вона складається з декількох основних блоків. Самі крайні блоки зліва, це блоки що надають інформації про реальні положення, параметри руху та швидкість транспортного засобу в момент відпрацювання розгляданого алгоритму керування. Далі іде блок «steering algorithm», що містить сам алгоритм керування, після якого йде «command velocity publisher», який передає інформацію про те, як повинен відпрацьовувати досліджуваний алгоритм сам транспортний засіб.

Спершу було виконано експеримент руху до заданої точки. Візуалізація даного експерименту зображено на рис.5.3.

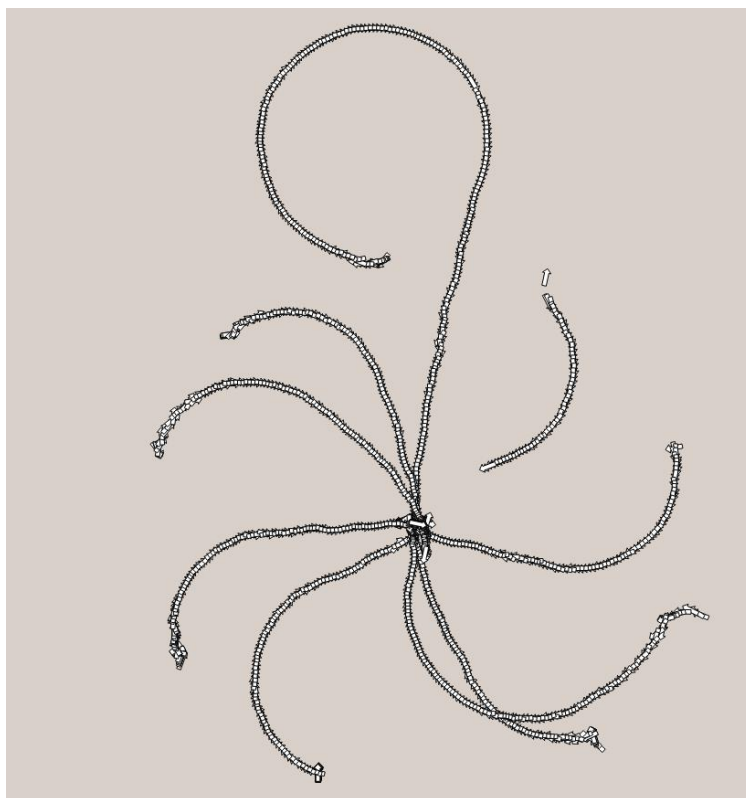


Рисунок 5.3 – Візуалізація відпрацювання алгоритму руху до заданої точки досліджуваним прототипом машинки

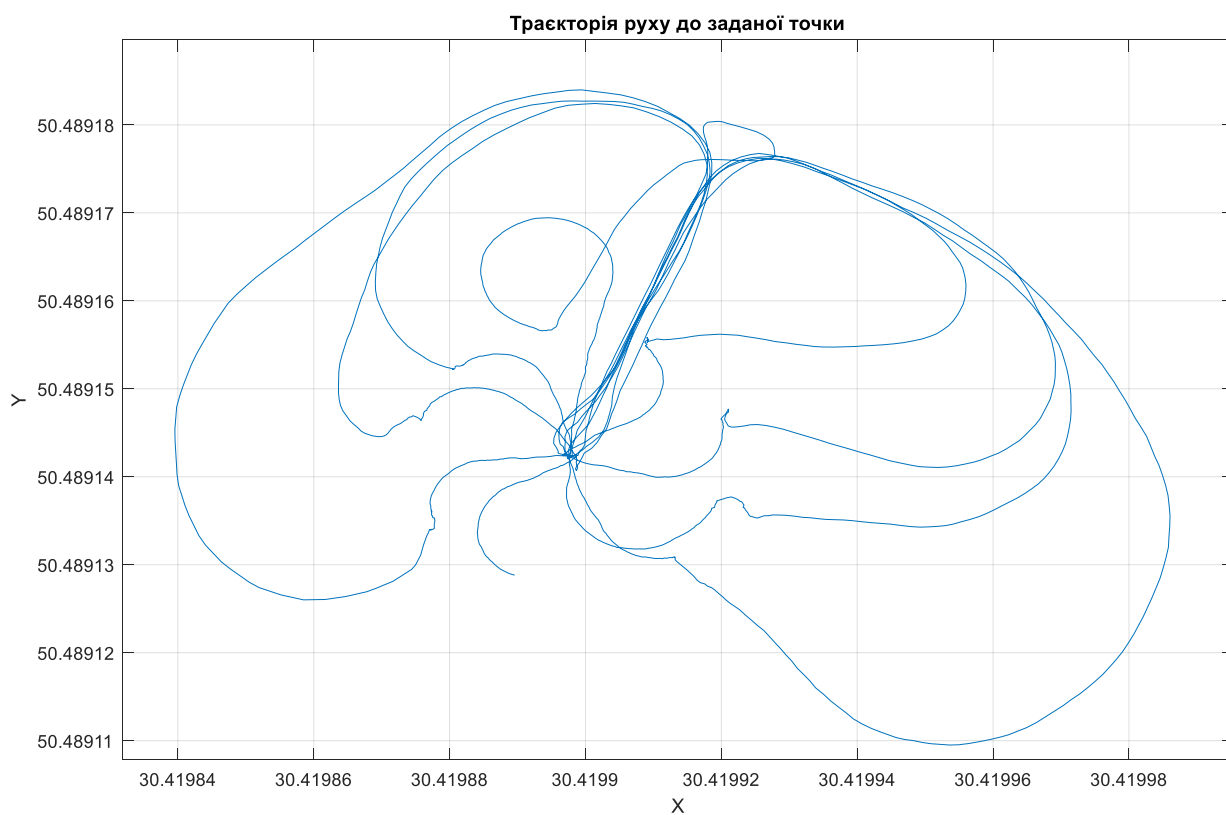


Рисунок 5.4 – Візуалізація відпрацювання алгоритму через середовище MatLab

Рис.5.4 має наступний вигляд через те, що на ньому зображено також службові лінії, тобто лінії, що утворилися при перенесенні досліджуваного транспортного засобу на задані позиції.

Як видно з візуалізації транспортний засіб успішно відпрацьовує алгоритм, адже з різних позицій транспортний засіб в кінці-кінців приїздить до заданої точки.

Після цього було проведено експериментальне дослідження за відпрацювання алгоритму руху відносно заданої лінії рис.5.5.

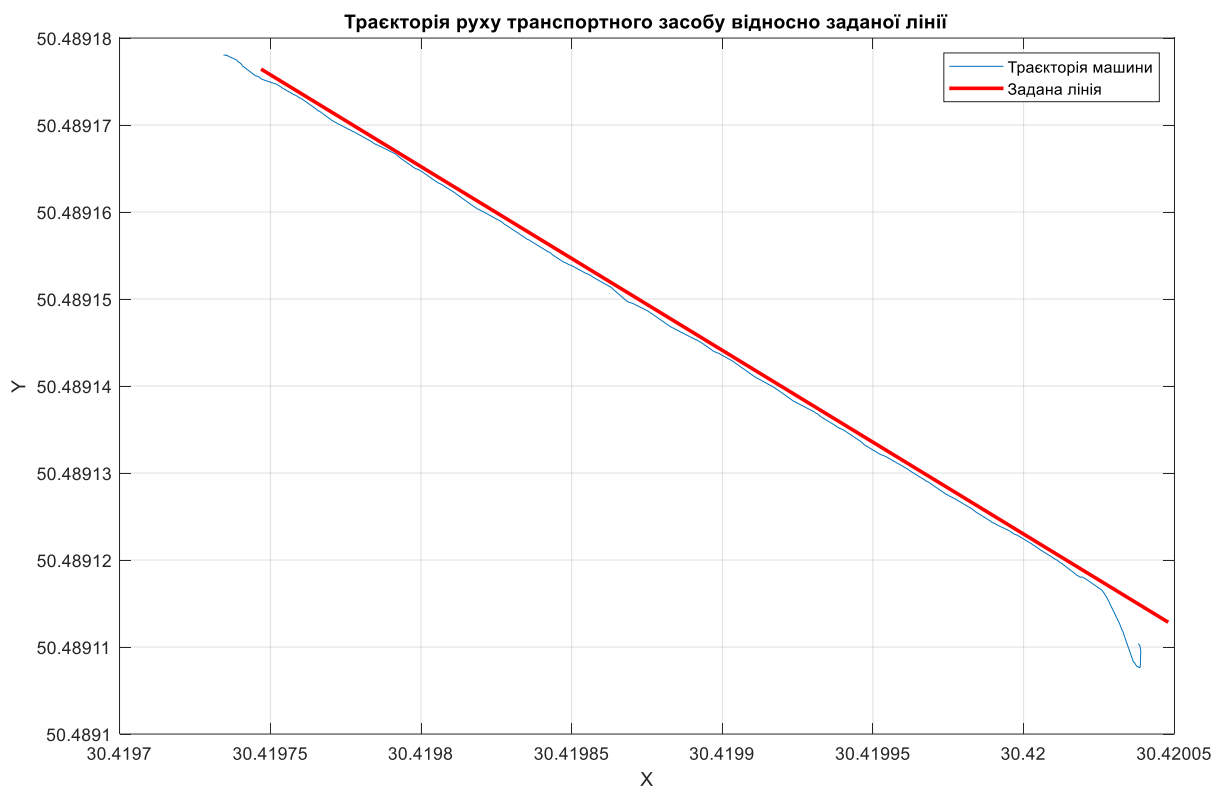


Рисунок 5.5 – Візуалізація відпрацювання алгоритму руху відносно заданої лінії прототипом транспортного засобу

Окрім візуалізації відпрацювання самого алгоритму, було також виведено графік похибки відпрацювання заданого алгоритму рис.5.6.

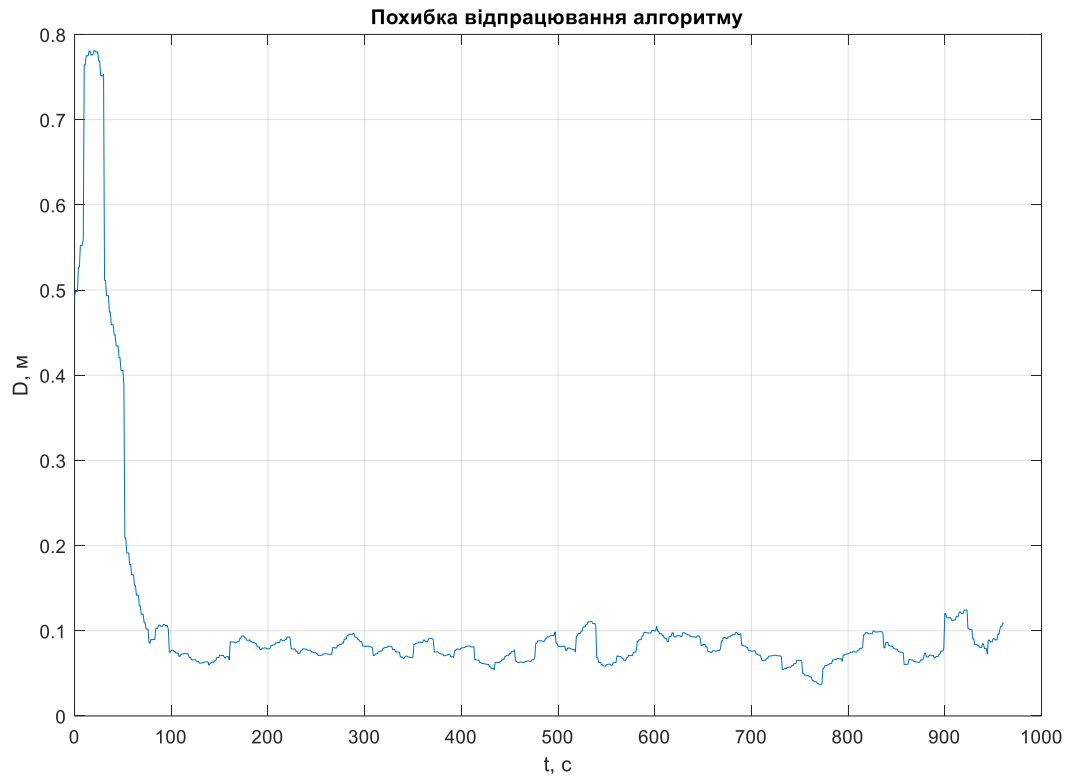


Рисунок 5.7 – Візуалізація похибки відпрацювання алгоритму

Як видно з графіку відпрацювання заданого алгоритму для досліджуваного транспортного засобу, максимальне значення похибки становить близько 8 см, причиною чого виступає точність визначення позиції на встановленому GPS-приймачі, що складає ± 2 см. Тому якщо врахувати цей нюанс, то відпрацювання алгоритму слідування лінії транспортним засобом цілком задовольняє поставлену задачу, яка, в свою чергу, становить ± 5 см.

Висновки до розділу 5

В процесі виконання даного розділу було розглянуто:

- Що таке Robot Operating System (ROS) та переваги використання даного робочого простору;
- Основні використовувані поняття в процесі роботи з даним програмним пакетом;
- Інструментарій програмного пакету;
- Особливості вибору дистрибутиву та доступність на різних операційних системах;
- Процес інсталяції та налаштування дситрибутиву Foxu Fitzroy на операційній системі Linux Ubuntu;
- Пакет інструментів ROS Toolbox для робочого середовища MatLab та його практичне використання.

Виконано експериментальне дослідження відпрацювання алгоритмів транспортним засобом, за допомогою вищеописаних засобів та інструментів програмного пакету ROS, а також ROS Toolbox в робочому середовищі MatLab.

ВИСНОВКИ

В процесі виконання бакалаврської роботи досліджено систему керування руху транспортним засобом.

1. Розглянуто поведінку транспортного засобу при виконанні повороту.
2. Розглянуто процес та причину переходу до моделі Аккермана.
3. Розглянуто кінематичну модель велосипеда для транспортного засобу.
4. Виконано кінематичний опис та кінематичне моделювання чотириколісного транспортного засобу на основі моделі Аккермана.
5. Приведено модель чотириколісного транспортного засобу до кінематичної моделі велосипеда.
6. Виконано огляд алгоритму керування транспортним засобом, що здійснює рух до заданої точки, його геометрію та кінематику.
7. Аналогічно розглянуто алгоритм слідування по заданій лінії.
8. Виконано дослідження алгоритмів керування транспортним засобом шляхом математичного моделювання в середовищі MatLab Simulink, звідки було розглянуто вплив коефіцієнтів регуляторів на якість відпрацювання перехідних процесів алгоритмів керування.
9. Описано пакет програм ROS, що призначений для практичної реалізації алгоритмів керування транспортним засобом або мобільним роботом. Розглянуто процес його інсталяції та налаштування на операційній системі Linux. Виконано огляд набору інструментів ROS Toolbox для робочого середовища MatLab Simulink.
10. Проведено експеримент з дослідження алгоритмів керування на реальному прототипі транспортного засобу за допомогою програмного пакету ROS, а також пакету інструментів ROS Toolbox в робочому середовищі MatLab.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Corke P. Robotics vision and control. School of Electrical Engineering and Computer Science, Queensland University of Technology (QUT), Brisbane, 2017, 697 p.
2. King-Hele D. Erasmus Darwin's improved design for steering carriages-and cars: Notes and Records of the Royal Society of London, Vol. 56, No. 1, 2002, pp. 41- 62.
3. Rajamani R. Vehicle dynamics and control. Department of Mechanical Engineering, University of Minnesota, Minneapolis, 2012, 495 p.
4. Belkhouche B. A method for robot navigation toward a moving goal with unknown maneuvers: Robotica 23(6), 2005, pp.709-720
5. Kapetanović N., Bibuli M., Caccia M., Mišković N. Real-time model predictive line following control for underactuated marine vehicles: IFAC-PapersOnLine, Vol. 50, Issue 1, 2017, pp. 12374-12379.
6. Muir P., Neuman C. Kinematic Modeling of Wheeled Mobile Robots: Journal of field robotics, Vol.4 , Issue 2,1987, pp. 281-340.
7. Siegwart R. Introduction to Autonomous Mobile Robots. MIT Press, 2004, 450 p.
8. Xiaokang S., Dalong T., Zhenwei W. Kinematics Modeling and Analyses of All-terrain Wheeled Mobile Robots: Chinese Journal of Mechanical Engineering, 2008, 44(6), pp. 148-154.
9. Gang M. Analysis of turning characteristic and the principle of Ackerman turning for vehicle: Mechanical Research & Application, 2007, 20(4), pp. 36–38.
10. Zhaowen D., Fuxing Z. The Analysis of Steady-State Steering Characteristic Turning Radius-Based in Vehicle: Agricultural Equipment & Vehicle Engineering, 007,1, pp.23-26.

11. Quigley M.,Gerkey B., Conley K., Faust J., Foote T.,Leibs J., Berger E., Wheeler R., Ng A. ROS: an open-source Robot Operating System: ICRA Workshop on Open Source Software, Vol.3, Issue 3.2, 2007, 6 p.
12. ROS Noetic Ninjemys [Электронный ресурс] URL: <http://wiki.ros.org/ROS/Introduction> (Last accessed: 28.05.21)
13. ROS Toolbox, Design, simulate, and deploy ROS-based applications [Электронный ресурс]. URL: <https://uk.mathworks.com/products/ros.html> (Last accessed: 29.05.21)