

КВАЛІФІКАЦІЙНА РОБОТА

магістр

Освітній рівень

Метод керування автономним рухом безпілотною транспортного засобу

Назва теми

КвРАКІТР.2023214.02.13.ПЗ

Галузь знань 17 «Електроніка, автоматизація та електронні комунікації»

Шифр, назва

Спеціальність 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Шифр, назва

Освітня програма «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Назва

Виконав:


студент II курсу, група АКІТРм-23-2


Підпис

Олександр СПОРІЙ

Ім'я, ПРІЗВИЩЕ

Керівник


Підпис, дата

Денис МАКАРИШКІН

Ім'я, ПРІЗВИЩЕ

Нормоконтролер


Підпис, дата

Людмила КОРЕЦЬКА

Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
зав. кафедри АКІТтаР


Підпис, дата

Валерій МАРТИНЮК

Ім'я, ПРІЗВИЩЕ

« 10 » грудня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Рівень вищої освіти другий (магістерський)

Галузь знань 17 – Електроніка, автоматизація та електронні комунікації

Спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітня програма Автоматизація, комп'ютерно-інтегровані технології та робототехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри АКИТтаР

Валерій МАРТИНЮК

01 вересня 2024р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Спорому Олександр Володимировичу

Прізвище, ім'я, по батькові студента

1 Тема роботи Метод керування автономним рухом безпілотного транспортного засобу

Керівник роботи Макаришкін Денис Анатолійович, к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, учене звання

Затверджено наказом ректора університету від 26.08.2024 р. №60

2 Строк подання студентом роботи на кафедру 02.12.2024р.





3 Вихідні дані до роботи Аналіз сучасного стану автоматизованих систем керування безпілотними транспортними засобами. Розробка алгоритмічного та програмного забезпечення автоматизованої системи керування безпілотним транспортним засобом на прикладі мобільного роботу Junior.

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

Вступ. Проблематика переміщення вантажу із використанням безпілотних транспортних засобів. Моделювання системи керування переміщенням вантажу із використанням безпілотних транспортних засобів. Моделювання роботи безпілотного транспортного засобу. Дослідження системи навігації для безпілотного транспортного засобу «JUNIOR». Висновки.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
презентаційні матеріали (слайди)

6 Консультанти розділів кваліфікаційної роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|--|---|
| | | завдання видав | завдання прийняв |
| Антиплагіат | Федула М.В., доцент кафедри АКИТтаР |  |  |
| Нормоконтроль | Корецька Л.О., доцент кафедри АКИТтаР |  |  |

7 Дата видачі завдання 01 вересня 2024р.

КАЛЕНДАРНИЙ ПЛАН

| Назва розділу кваліфікаційної роботи | Строк виконання | Примітка |
|--|-----------------|----------|
| 1 Вступ | 20.09.2024р. | виконано |
| 2 Проблематика переміщення вантажу із використанням безпілотних транспортних засобів | 15.10.2024р. | виконано |
| 3 Моделювання системи керування переміщенням вантажу із використанням безпілотних транспортних засобів | 30.10.2024р. | виконано |
| 4 Моделювання роботи безпілотного транспортного засобу | 15.11.2024р. | виконано |
| 5 Дослідження системи навігації для безпілотного транспортного засобу «JUNIOR» | 25.11.2024р. | виконано |
| 6 Висновки | 30.11.2024р. | виконано |
| 7 Оформлення пояснювальної записки до КРМ | 01.12.2024р. | виконано |
| 8 Оформлення презентаційних матеріалів | 01.12.2024р. | виконано |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Студент


Підпис

Олександр СПОРІЙ
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Денис МАКАРИШКІН
Ім'я, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Метод керування автономним рухом мобільного робота».

Автор роботи: Спорий Олександр Володимирович.

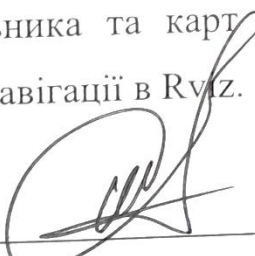
Керівник роботи: Макаришкін Денис Анатолійович

Пояснювальна записка: 77 с., 21 рис., 9 табл., 6 дод., 72 джерел.

Графічна частина: 12 презентаційних слайдів.

МОБІЛЬНИЙ РОБОТ, МОДЕЛЮВАННЯ, СИСТЕМА НАВІГАЦІЇ, 3-D ВІЗУАЛІЗАЦІЯ, АВТОМАТИЗОВАНА СИСТЕМА КЕРУВАННЯ.

Мета роботи: розробити автоматизовану систему керування мобільного робота. Розроблено модель мобільного робота в симуляторі Gazebo, інтегрованому у ROS. Створена модель мобільного робота відповідає реальним розмірам, його габаритним характеристикам, наявності датчиків. Досліджено систему навігації для мобільного робота «JUNIOR», а саме досліджено систему навігації в програмному середовищі ROS, коригування отриманих даних з лазера моделі мобільного робота, описано налаштування локального планувальника та карт вартості планувальника та представлено апробацію системи навігації в RViz.



Підпис студента

9.12.2024р.

Дата

ЗМІСТ

| | с. |
|--|----|
| ВСТУП | 4 |
| 1 ПРОБЛЕМАТИКА ПЕРЕМІЩЕННЯ ВАНТАЖУ ІЗ ВИКОРИСТАННЯМ БЕЗПЛОТНИХ ТРАНСПОРТНИХ ЗАСОБІВ..... | 8 |
| 1.1 Огляд та аналіз технічних та технологічних рішень..... | 8 |
| 1.2 Постановка завдань для керування автономним рухом мобільного робота..... | 18 |
| 1.3 Висновки до першого розділу..... | 18 |
| 2 МОДЕЛЮВАННЯ СИСТЕМИ КЕРУВАННЯ ПЕРЕМІЩЕННЯМ ВАНТАЖУ ІЗ ВИКОРИСТАННЯМ БЕЗПЛОТНИХ ТРАНСПОРТНИХ ЗАСОБІВ..... | 19 |
| 2.1 Характеристика систем керування ланцюгами переміщення виробів | 20 |
| 2.2 Ідея вирішення проблеми Last mile delivery..... | 24 |
| 2.3 Програмне забезпечення для керування автономним рухом мобільного робота..... | 30 |
| 2.4 Висновки до другого розділу..... | 33 |
| 3 МОДЕЛЮВАННЯ РОБОТИ БЕЗПЛОТНОГО ТРАНСПОРТНОГО ЗАСОБУ..... | 34 |
| 3.1 Характеристика мобільного робота AURORA «Junior»..... | 35 |
| 3.2 Моделювання основних елементів МР «Junior»..... | 37 |
| 3.3 Налаштування ПД регуляторів моделі МР «Junior»..... | 40 |
| 3.4 Організація контролю моделі мобільного роботу «Junior»..... | 46 |
| 3.5 Висновки до третього розділу..... | 50 |
| 4 ДОСЛІДЖЕННЯ СИСТЕМИ НАВІГАЦІЇ ДЛЯ БЕЗПЛОТНОГО ТРАНСПОРТНОГО ЗАСОБУ «JUNIOR»..... | 51 |
| 4.1 Дослідження системи навігації ROS..... | 51 |
| 4.2 Коригування отриманих даних з лазера моделі МР..... | 54 |
| 4.3 Налаштування локального планувальника..... | 55 |

| | |
|---|----|
| 4.4 Налаштування карт вартості планувальника..... | 65 |
| 4.5 Апробація системи навігації в Rviz..... | 70 |
| 4.6 Висновки до четвертого розділу..... | 76 |
| ВИСНОВКИ..... | 77 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 78 |
| ДОДАТКИ..... | 86 |

ВСТУП

В останні 20-25 років з'явилася нова галузь робототехніки, яка зосереджена на розробці нових автомобілів, так званих інтелектуальних транспортних засобів (ІТЗ). На сьогоднішній день автомобіль є одним із найважливіших винаходів минулого століття. В світі налічується близько мільярда автомобілів, і очікується, що дане число подвоїться в найближчі десять років [1-3]. Дана обставина спричинила активне розроблення методів автоматизації типових завдань, які виконуються людьми в процесі водіння ІТЗ.

Визначимо ІТЗ як ТЗ, оснащений спеціальними пристроями сприйняття навколишнього середовища, прийняття рішення та контролю, які дають змогу автоматизувати завдання водіння ТЗ. До завдань водіння відносять: об'їзд можливих перешкод, включно з обгоном інших ТЗ на дорозі; безпечне дотримання смуг руху згідно із встановленими правилами дорожнього руху; відстеження поведінки навколишніх об'єктів з оцінкою можливості виникнення небезпечних ситуацій на дорозі; слідування за встановленим маршрутом. Таке рішення розробляється для забезпечення безпеки та зручності пасажирів.

Інтеграція ІТЗ у життя людства поки що перебуває на початковому рівні. Технології та методи для безпілотних ТЗ все ще перебувають на стадії апробації в реальних умовах. Розробка інтелектуальних безпілотних автомобілів є популярним напрямком досліджень серед великих автомобільних корпорацій BMW [4], Tesla [5], а також великих ІТ-компаній - Google, Uber [6], Baidu (проект Apollo) [7].

Використання автономних ТЗ для вирішення інших завдань (крім транспортування пасажирів) так само викликає інтерес великих інвестиційних корпорацій.

Дослідження ведуться в наступних галузях промисловості:

- видобуток корисних копалин;
- сільське господарство;

- пошуково-рятувальні роботи;
- будівництво;
- інше [8].

Основною мотивацією для автоматизації даного типу завдання як керування ТЗ є зниження витрат на персонал, збільшення кількості часу роботи та виключення впливу людських факторів.

Важливо зазначити, що з плином часу технологія роботизації ІТЗ знайшла використання в галузі менеджменту та логістики - системі керування ланцюгами поставок (далі SCM-система) [9]. Зокрема, останнім часом актуальним стає вирішення завдання так званого останнього кілометра, тобто доставку товару зі складу безпосередньо кінцевому одержувачу [10]. У зв'язку з бажанням скоротити часові та ресурсні витрати на доставку в кінцеві точки традиційним способом (тобто людина-кур'єр), транспортні компанії розглядають альтернативні підходи, що використовують безпілотні транспортні засоби: безпілотні наземні роботи (БНР) та безпілотні літальні апарати (БПЛА) [11].

Метою роботи є розв'язання задачі останнього кілометра на основі мобільного робота (МР), що включає створення реалістичної моделі МР, і подальша апробація моделі МР для вирішення завдання доставки.

Для досягнення цієї мети необхідно розв'язати наступні **завдання**:

- комп'ютерне моделювання МР;
- розробка системи керування МР;
- проєктування контролерів приводів та системи автономної навігації та орієнтування МР.

Об'єктом дослідження є процеси керування мобільним роботом для переміщення вантажу в умовах міської забудови.

Предметом дослідження є методи та алгоритми керування мобільним роботом для переміщення вантажу в умовах міської забудови.

Методи досліджень. Використовувались основні положення загальної теорії систем, теорії моделювання процесів, системного аналізу (декомпозиції,

ієрархічності та ін.). Внаслідок проведення моделювання процесу та розроблення методів керування мобільним роботом для переміщення вантажу в умовах міської забудови використано апарат модельно-орієнтованих підходів, методи концептуального моделювання, теоретико-множинні підходи, алгебру систем, принципи побудови баз знань та формування логічного висновку, евристичні оцінки.

Наукова новизна отриманих результатів: розроблено модель системи автоматичного керування мобільним роботом для переміщення вантажу в умовах міської забудови в режимі реального часу.

Практична значимість отриманих результатів: розробленні програмно-технічні засоби для керування мобільним роботом для переміщення вантажу в умовах міської забудови.

Апробація результатів дослідження: опубліковано повнотекстову статтю у фаховому виданні Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах» Випуск №3, 2024 рік.

Кваліфікаційна робота складається із вступу, чотирьох розділів, висновків до кожного розділу, загальних висновків, списку використаних джерел, шести додатків. Загальний обсяг роботи складає 77 сторінок комп'ютерного тексту, у тому числі: 21 рисунок та 9 таблиць, список використаних джерел вміщує 72 найменування.

У вступі сформульовано мету та завдання досліджень, обґрунтована актуальність наукової задачі, відображено основні наукові результати та практичне значення.

У першому розділі магістерської кваліфікаційної роботи описано проблематику переміщення вантажу із використанням безпілотних транспортних засобів, для чого було проведено огляд та аналіз технічних та технологічних рішень з даного питання. Проведено постановку завдань для керування автономним рухом мобільного робота.

У другому розділі кваліфікаційної роботи виконано моделювання системи

керування переміщенням вантажу із використанням безпілотних транспортних засобів, а саме охарактеризовано систему керування ланцюгами переміщення виробів, запропоновано ідею вирішення проблеми Last mile delivery, проаналізовано програмне забезпечення для керування автономним рухом мобільного робота.

У третьому розділі кваліфікаційної роботи магістра виконано моделювання роботи мобільного робота для переміщення виробів, а саме охарактеризовано конструкцію мобільного робота AURORA «Junior», виконано моделювання основних елементів мобільного робота «Junior» в програмному середовищі. Проаналізовано налаштування ПД регуляторів моделі мобільного робота «Junior», а також організовано контроль моделі мобільного робота «Junior».

У четвертому розділі кваліфікаційної роботи наведено дослідження системи навігації для мобільного робота «JUNIOR», а саме дослідження системи навігації в програмному середовищі ROS, коригування отриманих даних з лазера моделі мобільного робота, описано налаштування локального планувальника та карт вартості планувальника та представлено апробацію системи навігації в Rviz.

У загальних висновках по кваліфікаційній роботі магістра представлено отримані результати роботи «Метод керування автономним рухом мобільного робота».

1 ПРОБЛЕМАТИКА ПЕРЕМІЩЕННЯ ВАНТАЖУ ІЗ ВИКОРИСТАННЯМ БЕЗПЛОТНИХ ТРАНСПОРТНИХ ЗАСОБІВ

1.1 Огляд та аналіз технічних та технологічних рішень

На сучасному етапі існує низка дослідницьких робіт, присвячених автоматизації процесів SCM систем. Одна із робіт [12] присвячена аналізу дій компанії Amazon [13], як прикладу успішного використання роботизації на виробництві. Amazon [13] - одна з найбільших компаній, які перебудували свої склади в склади нового покоління, що використовують спеціалізованих автономних складських роботів. Витрати використання класичного підходу в компанії призводили до високих витрат часу виконання та, відповідно, відправлення замовлень клієнтам.

Серед проблем, з якими стикається компанія-гігант, автори роботи виділяють наступне:

- досить низька ефективність і продуктивність систем поставок і оформлення замовлень;
- великий час очікування отримання замовлення;
- відставання від технологічних інновацій компаній-конкурентів;
- інше.

Компанія Amazon [13] впровадила в складські відділи спеціалізованих автономних МР типу Kiva [14], які значно спростили збір замовлень, прискорили термін виконання замовлень, забезпечуючи гнучкість та масштабованість роботизованої системи. Головним чином, більша швидкодія призвела до загального підвищення лояльності клієнтів до фірми Amazon [13] і, як результат, підвищення продажів. Автори дослідження [12] роблять висновок, що подібна автоматизація може бути корисною й іншим компаніям-гігантам у сфері так званої e-commerce.

Інше дослідження [11] проводить огляд наявних рішень з автономної

доставки товарів за допомогою використання роботів. Також представлено декілька існуючих типів роботів: БНР, БПЛА та їхнє застосування в конкретному кейсі. Серед розробок автономної доставки за допомогою використання БПЛА відносять кейси компаній Dominos, UPS, Alphabet (компанія, що володіє Google), Amazon, DHL (поштова компанія). Автор дослідження [11] окремо виокремлює БНР, призначених для доставки товарів пішохідною зоною - тротуаром. Прикладами таких проєктів є компанія Marble [15], Starship Technologies [16], Kiwi [17]. Дослідник [11] розглядає безліч актуальних проєктів з автономної доставки товарів, які розвиваються на даному етапі. Таким чином, тематика останнього кілометра є затребуваною і увійшла на міжнародний ринок за запитом економіки.

Прикладом реального випадку розроблення МР для розв'язання проблеми останнього кілометра є наступне дослідження [18]. Автори [18] пропонують розв'язання проблеми в міських умовах за допомогою використання БПЛА (фірма Intel, модель БПЛА - Aero Ready to Fly), який діє за наступним принципом: БПЛА знаходить первинне (приблизне) місце доставки за допомогою GPS системи, кінцеве - за допомогою мітки, нанесеної на поверхню точки доставки. В місця посадки автори дослідження [18] виділяють балкон або ганок будинку з міткою на вікні чи стіні будівлі. Початковий концепт було спочатку протестовано в змодельованому середовищі ROS/Gazebo, далі успішно апробовано так би мовити у польових умовах. Автори дослідження [18] надають вихідний код розробки для інших груп розробників в якості основи для інших проєктів.

Автори дослідження [19] представляють рішення останнього кілометра за допомогою використання так званого робомобіля. У дослідженні представлено опис компонентів всієї системи: виявлення пішоходів, локалізація, навігація. В якості ТЗ використовується машина для гольфу Yamaha G22E. У даному дослідженні [19] дослідження зосереджено на створенні автономної системи ТЗ для роботи в умовах високої міської завантаженості. Автори дослідження

приділяють увагу процесу сприйняття ТЗ використовуючи декілька різних давачів (лазерний далекомір, камера, GPS, IMU) для точного визначення присутності інших об'єктів на дорозі в режимі реального часу. Аналіз інформації з давачів сприяє плануванню оптимальних траєкторій руху ТЗ уникаючи зіткнень з іншими учасниками руху.

Існує дослідження [20], яке зосереджене не тільки на процесі розробки сервісного МР для лікарень, а й на вдосконаленні методів керування траєкторією руху МР. Автори роботи зазначають, що сучасні мобільні робототехнічні системи (МРС) концентруються тільки на точності положення МР, залишаючи відкритим питання контролю руху в так званих небезпечних зонах (наприклад, сходи чи вузькі проходи). Для підвищення рівня безпечного функціонування МР необхідно не тільки аналізувати дані з давачів самого МР, а й з давачів, встановлених у його оточенні. У дослідженні [20] описується нова система - visible light communication system (світлова система комунікації) для запобігання ситуацій падіння МР в небезпечних ділянках в приміщеннях. Система комунікації повідомляє МР про близькість небезпечних місць, таким чином МР функціонує в замкнутому просторі, виконуючи завдання з доставки ліків у палати без ризику падіння. Автори роботи називають свою світлову систему внутрішнім GPS та локалізацією положення.

Багато проєктів, що пропонують використовувати МРС для вирішення завдань автоматизації SCM-систем, включають етап моделювання та проєктування індивідуального дизайну МР. Даний етап є ключовим початковим етапом розроблення власного проєкту; визначення сфери використання МР і середовище його функціонування безпосередньо визначають зовнішній вигляд і тип обмежень мобільності (тобто переміщень) МР у просторі. У рамках огляду літератури було розглянуто декілька робіт, присвячених дизайну платформ МРС і залежність конструкції МР від передбачуваного середовища використання. Багато вивчені роботи авторів спрямовані на вирішення конкретних прикладних завдань у сфері МР, зокрема - конфігурації коліс мобільної платформи, питання

поліпшення всеспрямованих коліс, розроблення нових дизайнів МР. Унаслідок цього деякі автори не регламентують існуючі проблеми робототехніки в статті, але описують своє рішення про реалізацію мобільної платформи із заданими вимогами до платформи (вимоги до переміщення, методів побудови карт і локалізації положення).

У дослідженні [21] автори пропонують один із можливих варіантів конструкції МР - складання недорогої, простої для розуміння та невеликої за габаритними розмірами платформи для використання в навчанні та наукових дослідженнях, що передбачає дизайн та виготовлення роботизованої платформи згідно з встановленими вимогами. Важливість поставленої проблеми полягає в тому, що в багатьох популярних МР архітектура апаратного рівня і набір інтерфейсів сильно відрізняється, внаслідок чого з кожним МР студентам необхідно проходити повільний етап ознайомлення як із апаратною так і з програмною частинами, перш ніж починати будь-яку роботу з даною моделлю МР. На основі аналізу наявних рішень автори розробили власну платформу, яка містить достатню кількість інтерфейсів вводу/виводу для виконання навчальних практичних завдань. У роботі розкривається поняття МР та різноманітність видів - у дослідженні є приклади наявних платформ, включно з посиланнями на повний опис моделі.

Автори детально описують апаратний та програмний рівні розроблюваної моделі МР, супроводжуючи опис докладними схемами використовуваних компонентів з коротким технічним описом. Особливо важливим розділом є докладний опис набору давачів та інтерфейсів, представлений у табличному вигляді. Автори дослідження за допомогою діаграм і схематичних рисунків продемонстрували архітектурне рішення, відобразивши взаємодію нижнього та верхнього рівнів платформи. Дане дослідження було опубліковано у 2017 році і її підхід є новим і цікавим рішенням поставленого завдання, включно з інтеграцією МР в Robot Operating System (ROS). Тоді як у більшості випадків, інтеграція МР в ROS відсутня і стає ще одним завданням (через велику кількість

відмінностей у конструкціях МР, що використовуються).

Інше дослідження [22] містить метод поліпшення якості руху всеспрямованої платформи підбором оптимальних параметрів встановлених ПІД регуляторів, використовуючи алгоритм диференціальної еволюції. Підхід ґрунтується на використанні отриманих розв'язків для прямої та зворотної задач кінематики платформи, що розглядається, для виконання аналітичних експериментів.

Результати моделювання аналізуються для оцінювання якості поточного підбору параметрів ПІД регулятора. Незважаючи на те що в даній роботі не порушено питання про використання методу для МР довільного типу, рішення можна застосувати для більшості МРС. Слід відмітити, що автори, незважаючи на отримання оптимальних параметрів виключно з моделювання, стверджують, що подібний підхід можна використовувати до будь-яких систем і набуде широкого поширення в практиці.

Зіткнувшись із проблемою великої складності програмного забезпечення (ПЗ) для МР (внаслідок наявності великої кількості бортових давачів) автори дослідження вирішують дану проблему за допомогою інтеграції системи ROS у свою розроблену платформу [23]. Перевагою даного дослідження є розкриття організації апаратних компонентів платформи у вигляді схеми розміщення, включно з позначенням займаного простору в МР. Дослідження також містить методику тестування розробленої платформи: рух по заданій траєкторії, тест на синхронізацію повороту коліс. Представлена інформація процесу складання містить змістовні інструкції, з докладним позначенням використаних компонентів: моделі апаратних компонентів, версій пакетів ROS та ОС Linux.

Єдиним недоліком дослідження [23] є недостатня детальність в описі інтеграції апаратної шини в послідовний порт ОС в ROS.

Автори дослідження [24] наголошують на особливому пріоритеті задач навігації та локалізації положення в невідомому навколишньому середовищі для МР, що виконують поставлені завдання в приміщеннях (наприклад, офісний

приміщення, житлове приміщення або лікарня). Автори вказують, що для ефективної навігації досить використовувати RGB-D камеру та метод Visual SLAM (або візуальний метод одночасної локалізації та картографування). Дані з камери RGB-D, крім двовимірного растра кадру, доповнені depth map (або картою глибини). Дана технологія використовується, наприклад, в камері Microsoft Kinect. Автори роботи наводять класифікацію типу руху МР залежно від виду використовуваних коліс, наводять кінематичну модель розробленої всеспрямованої платформи (яка використовує колеса типу Mecanum), частково описують апаратний рівень, наводять взаємодію рівнів системи керування (інтеграція з ROS, мікроконтролери, RGB-D камера). Автори роботи демонструють результат виконання SLAM за допомогою використання RGB-D камери, наводять результати експериментів щодо точності отриманих результатів. Даний метод побудови карти та локалізації положення загальновідомий, проте використовується рідше за методи, що базуються на використанні лазерних далекомірів.

Існує дослідження [25] присвячене розробці всеспрямованого МР на основі використання коліс Mecanum [26] для навчальних цілей. Автори дослідження визначають доцільність використання МР у різних повсякденних завданнях (транспортування об'єктів, спостереження, переміщення об'єктів у робочому кабінеті), що вимагає від МР здатності до швидкого маневрування під час переміщення та вміння пошуку ефективною траєкторію шляху до наступної точки плану виконання завдання. Автори дослідження вирішують дану проблему використанням всеспрямованих коліс типу Mecanum [26]. Наводять загальні характеристики та особливості використання наявних типів коліс (класичних, всеспрямованих тощо), наводять порівняльну таблицю із переваг та недоліків для наступних типів коліс: universal wheel (або універсальні всеспрямовані колеса), powered steered wheel (або ведучі кермові колеса), mecanum [26] (або так зване, шведське колесо) та caster wheel (або роликові колеса).

Виконаний порівняльний аналіз є корисною інформацією для будь-якого

інженера, який зацікавлений у проектуванні МРС для функціонування в заданих умовах. Крім якісних характеристик для типів коліс, автори дослідження детально описують розроблений ними МР: розв'язання задач кінематики, механічні властивості коліс, склад і маркування деталей (електроніка, типи бортових давачів). Основна користь даної статті полягає в докладному описі типів коліс у табличному вигляді, що демонструє переваги та недоліки для кожного типу коліс.

Крім того існує дослідження також присвячене типам коліс [28], яка в свою чергу є оглядовою і не містить власних розробок авторів, на відміну від інших статей. Автори дослідження відзначають особливу популярність всеспрямованих МР серед усіх мобільних і наводять усі типові структури використання всеспрямованих коліс. Проте, в описі не вказано приклади задачі, для вирішення яких вибір даного типу колеса буде оптимальним. Проте огляд містить всю мінімально необхідну інформацію про кожен із типів всеспрямованих коліс, подано схеми різних конфігурацій коліс у МР та містить доповнення порівняльної таблиці переваг та недоліків чотирьох типів коліс. Дана робота також містить цінні описи та схеми організації керування МР такого типу. Інша важлива перевага даного дослідження полягає в наявності бібліографії з безліччю посилань на конкретні практичні використання кожного типу всеспрямованого колеса, що служить початковою точкою огляду літератури для наявних МР, потенційних проблем притаманних певному виду МР і способах вирішення.

Інше розглянуте дослідження [28] описує процеси аналізу вимог, проектування та складання всеспрямованої МРС. Це більш практичне дослідження, що містить набір рекомендованих практик для проектування МРС з метою успішної участі в різних змаганнях, наприклад, Robo змаганнях, типу RoboCup (міжнародні змагання серед МР) або змагань тематики спеціалізованої пошуково-рятувальної робототехніки.

Тип МРС, тобто всеспрямований МРС, був обрано на основі аналізу всіх

змагальних роботів класу «мобільна платформа», як зазначено раніше, всеспрямована платформа краща за будь-яку іншу неголономну. Автори роботи дають рекомендації до вирішення завдання кінематики МР і розміщення апаратних частин. Особливо корисним є приклади конфігурації коліс і готове креслення панелі МР для встановлення коліс. Також представлено використані в процесі розробки проєкту аналітичні експерименти для перевірки маневреності платформи. Серед недоліків можна відмітити відсутність формальних пояснень як для конкретної конфігурації коліс і на яких підставах було обрано конкретну форму панелі для встановлення коліс. Тому можна було б зробити невелику частинку статті про методику вибору конфігурації коліс, причини непопулярності деяких конфігурацій.

Незважаючи на велику популярність всеспрямованих коліс, вони володіють низкою недоліків, які складно компенсувати: наприклад, дані колеса не можуть функціонувати в середовищі з брудною або нерівною поверхнею, дослідження [27] відзначає додатковий недолік, такий як неефективність при переміщенні важких вантажів. Для підвищення ефективності та збереження корисних властивостей, автори роботи представляють свою модифікацію всеспрямованого колеса. Особливості даного колеса описується в дослідженнях [29, 30], в яких розкривається як принцип роботи, так і методи розміщення та керування. Принципово новим є здатність даного типу колеса на вільне переміщення по нерівній поверхні з більшим навантаженням.

Автори дослідження [31] запропонували два способи покращення шведського всеспрямованого колеса. Автори даного дослідження чітко вказують розв'язувану проблему: під час переміщення похилою або нерівною поверхнею може виникнути ситуація, в якій частина коліс торкатимуться поверхні лише ободом колеса, а не роликami (тобто пасивними елементами шведського колеса), тим самим перешкоджаючи подальшому руху МРС. Автори дослідження зауважують, що винахідник, так званого, шведського колеса [26] Ілон запропонував дещо змінений механічний дизайн для вирішення даної проблеми.

Однак покращення має також і негативний ефект - знижується ефективність колеса, внаслідок використання передачі сили через ролики колеса, що обертаються [31]. З огляду на обидва зазначені недоліки варіантів колеса, автори роботи пропонують простий та складний варіанти вирішення. Складний, варіант включає в себе зміну механіки колеса включенням у конструкцію спеціальних роликів, що обертаються. Перевагою даного дослідження є його прикладне спрямування і наявність прикладів використання в МР, однак автори дослідження не вказали на експериментальне підтвердження роботи поліпшеної моделі шведського колеса; робота описує тільки модель колеса, але без результатів експериментів неможливо підтвердити усунення недоліків класичного шведського колеса. Відсутній опис кінематики колеса, що в свою чергу є одним із ключових недоліків статті.

Ідея про покращення конструкції шведського колеса [26] була також запропонована в дослідженні [32]. Автори дослідження аналогічно визначають головне обмеження в використанні шведського колеса, як можливість його застосування тільки під час переміщення по рівним поверхням. Автори дослідження пропонують два можливих вирішення проблеми, проте обидва включають зміни механіки коліс. Перевагою даного дослідження є наявність якісних та наочних схем рішень, що пропонуються, проте автори дослідження не претендують на досконалість своїх рішень і описують декілька недоліків. Дане дослідження може розглядатися тільки як гіпотетичне вирішення проблеми - доти, доки автори дослідження не виконають експериментального підтвердження.

Прикладні дослідження, що описують розроблення всеспрямованої МРС розглядаються в роботах [33, 34]. Об'єднує дослідження використання знову ж таки шведського колеса.

Автори дослідження [33] проектували апаратну частину свого МР для керування одним мікроконтролером, що забезпечує повноцінний всеспрямований рух із мінімумом використовуваних компонентів. В апаратній

частині автори роботи не використовують жодних давачів крім енкодерів, інтегрованих у електропривід коліс. У рамках своєї майбутнього дослідження вони планують розвивати МР з метою створення дешевого модульного автономного МР для використання на об'єктах виробництва [33].

Автори дослідження [34] також пропонують способи вирішити одну з головних сучасних проблем робототехніки - високу вартість обладнання. Дороговизна МР значно обмежує використання в завданнях, у яких ризик пошкодження системи істотно високий. У даному дослідженні наводиться один зі способів зниження вартості - використання деталей МР, надрукованих за допомогою використання 3D-принтера.

Автори дослідження пропонують моделі для друку деталей всеспрямованої МРС та посібник зі складання. Результатом є повністю надрукований МР. Актуальність даного дослідження зумовлена доступністю та дешевизною друку на 3D-принтері на сучасному етапі, що в умовах навчальних закладів буде одним із економічно вигідних і найоптимальніших рішень. Автори дослідження демонструють, як даний спосіб дає змогу швидко розширювати сферу використання даного МР: приклад створення і підключення базового маніпулятора для роботи над такими завданнями, як закручування/відкручування кріплень або відкривання/закривання дверей.

Автори дослідження [35] розв'язують на базі своєї розробленої МРС завдання організації системи віддаленого керування. Система віддаленого керування розроблялася з метою спрощення керування рухом МР з додатковим графічним інтерфейсом моніторингу та діагностики параметрів роботи [35]. Актуальність організації спеціалізованих захищених систем віддаленого керування зумовлена ще й необхідністю захисту від втручання злоумисників. Розробка автономних платформ здебільшого включає в себе організацію віддаленого доступу для налагодження або управління вручну. Автори дослідження розглянули декілька способів керування МР і реалізували свою систему, яка використовує ручний джойстик. Сильною стороною даного

дослідження є формальний опис організації системи віддаленого керування за допомогою використання блок-схем.

1.2 Постановка завдань для керування автономним рухом мобільного робота

Для досягнення поставленого завдання кваліфікаційної роботи магістра необхідно розв'язати наступні завдання:

- комп'ютерне моделювання МР;
- розробка системи керування МР;
- проектування контролерів приводів та системи автономної навігації та орієнтування МР.

1.3 Висновки до першого розділу

Було описано проблематику переміщення вантажу із використанням безпілотних (інтелектуальних) транспортних засобів, для чого було проведено огляд та аналіз технічних та технологічних рішень з даного питання. Крім того виконано постановку завдань для керування автономним рухом мобільного робота.

2 МОДЕЛЮВАННЯ СИСТЕМИ КЕРУВАННЯ ПЕРЕМІЩЕННЯМ ВАНТАЖУ ІЗ ВИКОРИСТАННЯМ БЕЗПЛОТНИХ ТРАНСПОРТНИХ ЗАСОБІВ

Supply chain management (SCM) або керування ланцюгами поставок - це спосіб організації контролю за процесом виробництва і переміщення товарів або послуг, починаючи від постачання сировини та компонентів продукту до доставки продукту споживачеві. Для контролю поставленого завдання організовується граф організацій поставок, що беруть участь у виробництві або транспортуванні вже готової продукції, включно з постачальниками сировини, фабриками, складами чи точками продажу товару кінцевим споживачам.

Методи SCM значною мірою базуються на методах та стратегіях промислового проєктування, системного проєктування, керування операціями, логістики, закупівель, інформаційних технологій та маркетингу, це виконується для вироблення загального комплексного підходу.

Фундаментальні питання, досліджувані в галузі SCM, ґрунтуються на методах теорії керування та стійкості для пошуку рішень для керування виникаючими ризиками [36].

Виокремлюють такі ключові процеси в ланцюжку поставок товарів:

- управління взаємовідносинами з клієнтами;
- управління обслуговуванням клієнтів;
- керування попитом;
- виконання замовлення клієнту;
- керування виробничим потоком;
- управління відносинами з постачальниками деталей;
- розробка продукту та комерціалізація;
- керування процесами повернення.

Однією з базових задач SCM-систем є мінімізація загальних витрат, з урахуванням наявних конфліктів, що існують між учасниками ланцюжка

поставок товарів. Прикладом таких конфліктів є взаємозв'язок між відділом продажів, який бажає володіти більш високим рівнем кількості запасів товарів для задоволення можливих потреб клієнтів, і складом, для якого більш нижчий рівень запасів товарів необхідний для зниження витрат енергетичних ресурсів на зберігання продукції.

2.1 Характеристика систем керування ланцюгами переміщення виробів

Розрізняють 6 компонентів традиційного керування ланцюгами переміщення виробів:

- планування - планування та керування всіма ресурсами, необхідними для задоволення попиту клієнтів на продукт або послугу компанії;
- пошук «джерел» або sourcing - вибір постачальників для надання товарів і послуг, необхідних для створення продукту.

Встановлення процесів для моніторингу та керування відносинами з постачальниками. Ключові процеси даного компонента включають замовлення, отримання, керування запасами та авторизацію платежів постачальників;

- making або виробництво - організація заходів, необхідних для приймання сировини, виробництва готового продукту, перевірки якості товару, пакування для відправлення клієнту;

- логістика або доставка - координація замовлень клієнтів, планування доставки товарів, відправлення вантажів, виставлення рахунків клієнтам та отримання платежів;

- повернення товарів - створення мережі або процесу для повернення дефектних, надлишкових чи небажаних товарів;

- enabling або включення - створення процесів підтримки для моніторингу інформації по всьому ланцюгу поставок товарів і забезпечення відповідності всім встановленим нормативним вимогам. Дані процеси включають: людські ресурси, фінанси, керування об'єктами, сферу ІТ, керування дизайном продукту,

продажами та забезпечення якості товарів.

Автоматизація завдань SCM-систем полягає на сьогоднішній день в тому, що світ наближається до переходу на нову, так звану, парадигму Індустрія 4.0. Індустрія 4.0 характеризується повсюдним впровадженням кіберфізичних систем у життя людини, у технологічні процеси виробництва різних галузей. До кіберфізичних систем відносять робототехніку, хмарні обчислення, великі дані, симулятори, доповнену реальність, інтернет речей (або internet of things, IoT), інформаційну безпеку [37].

Індустрія 4.0 відноситься до сучасної тенденції автоматизації та обміну даними у виробничих технологіях. Деякі завдання SCM-систем можливо автоматизувати наступним чином:

- автоматизований процес виробництва на заводах;
- автономні вантажні машини, що транспортують товари на склади з оновленням місця розташування продукту в реальному часі;
- автоматизовані склади, що використовують МР для обробки всіх операцій: від збору до переміщення продуктів по території складу;
- відстеження статусу продукту замовником;
- автономні дрони в якості доставщики товарів у кінцеву точку призначення [38].

Нові технології керування ланцюгами поставок товарів включають робототехніку, великі дані, інтелектуальний склад, IoT та штучний інтелект. Дані технології нині впливають на кожен компонент ланцюга поставок товарів, а саме: задоволення потреб споживачів, пошук постачальників, закупівлі, розподіл, переміщення та зберігання на спеціальних складах [38]. Прогрес сприяє зростанню ефективності, економічності технологічних процесів, скороченню часу виконання замовлення та економії коштів у ланцюзі поставок товарів [39].

В свою чергу автоматизація завдань поставок товарів може призвести до втрати робочих місць на низькому (або виробничому) рівні. Таким чином, з

часом персоналу необхідно підвищувати власну кваліфікацію і вміти використовувати новітні технології.

Проблема, так званого, останнього кілометра - термін last mile delivery описує завдання доставки готового товару з єдиної транспортної точки (або transportation hub) у кінцеву точку призначення тобто до споживача. Last mile delivery є частиною процесу планування поставок і перевезень товарів зі складу в кінцевий пункт призначення, будучи частиною процесу SCM-систем - сфери логістики та доставки.

Затребуваність доставки товару в кінцевий пункт тільки зростає - все частіше здійснюються покупки готових товарів онлайн, оскільки сфера інтернет-продажів досягає максимуму у своїй різноманітності надання послуг (одяг, продукти, товари для дому, техніка, лікарські препарати тощо); клієнти очікують на доставку в найкоротші терміни. Таким чином, усередині даного процесу виділяють декілька головних проблем транспортування готових товарів:

- дороге переміщення готового товару до кінцевої точки призначення при використанні послуг кур'єра на автомобілі чи іншому виді транспорту (великі витрати на ресурси доставки);

- більшість споживачів перебувають поза домом, коли відбувається доставка товару; внаслідок такого випадку товар може зіпсуватися через навколишні умови або бути вкраденим;

- організація оптимальної та ефективної доставки.

Однак найбільше впливу дана проблема має не на кінцевих споживачів товарів, а на бізнес, що займається доставкою або продажем товарів. Серед великих фірм, які зіткнулися із завданням Last mile delivery, виокремлюють компанії-гіганти Amazon та Walmart, Alibaba, UPS, FedEx, та ін. Наразі є декілька відомих способів вирішення даних проблем.

- постійна взаємодія з споживачем: своєчасна і швидка доставка замовленого товару, постійне інформування споживача про статус доставки товару («у стадії відправлення», «у процесі доставки», «у дорозі», тощо).

- зростання кількості центрів-складів у місті. це призведе до зменшення відстані між складом товаром та споживачами.

- надання можливості споживачу відстежувати в реальному часі статус замовлень.

- оптимізація процесу доставки товару. Головним чином включає в себе завдання удосконалення планування маршруту доставки замовленого товару для того, щоб скоротити загальний час доставки, споживання палива для ТЗ. При цьому алгоритм планування має враховувати: близькість місця розташування, доступність кур'єра в даний момент, вагову місткість ТЗ доставки та інші фактори.

- використання зручних точок із встановленими сейфами. Один із базових способів обійти проблему дорогих доставок одиничних замовлень - встановлення спеціальних шаф-сейфів у доступних місцях, наприклад торгових центрах чи банках. Отримуючи спеціальний ключ або сканер штрих-коду, споживач зможе отримати посилку безпечним і надійним способом.

- використання автономних ТЗ. На сучасному етапі це один зі способів розширення так званої, on-demand economy або економіки «за запитом», економічна активність технологічних компаній, цифрових торговельних майданчиків, що задовольняють запити споживачів негайним наданням послуг чи товарів). Хоча наразі існують невирішені питання щодо використання автономних ТЗ у міських просторах, варіант використання БПЛА є одним із потенційних способів вирішення проблем Last mile delivery.

Проблема Last mile delivery є актуальною темою для розробки якісних та нових рішень, які б оптимізували процес доставки готових товарів та зменшили витрати компаній на ресурси доставки.

Один із варіантом розв'язання проблеми є впровадження в процес доставки, наприклад, МР типу «Юніор» і програмування на здійснення автономної доставки від точки складу до кінцевої точки призначення.

2.2 Ідея вирішення проблеми Last mile delivery

Одне з рішень проблеми Last mile delivery є можливість використання БПЛА. Дане рішення має основну перевагу в тому, що БПЛА не залежать від величини дорожнього трафіку, особливостей наземних маршрутів доставки; це гнучке рішення, що не потребує змін у дорожньому середовищі, не становить критичної небезпеки для оточуючих (за дотримання техніки безпеки з боку людини). З іншого боку, БПЛА супроводжується обмеженнями в використанні. По-перше, БПЛА жорстко обмежений можливою вантажопідйомністю, що безпосередньо впливає на характер товарів, які доставляються, в свою чергу сильно скорочує асортимент видів товарів для доставки. По-друге, БПЛА в середньому може перебувати в режимі польоту близько півгодини, що приблизно відповідає одній доставці в кінцеву точку. По-третє, виходячи з обмеження за часом польоту БПЛА складається серйозна проблема розбіжності ідеології Last mile delivery із запропонованим рішенням: доставка повинна бути ефективною й охоплювати декілька кінцевих точок за низької витраті ресурсів; БПЛА ж у такому випадку при повному заряді може охопити одну - дві суміжні кінцевих точки. У такому випадку можна припустити, що на складі буде кілька десятків БПЛА, розподілених за зонами доставки, однак головне обмеження залишається незмінним - склад повинен перебувати в безпосередній близькості до кінцевих точок доставки, оскільки БПЛА необхідно не тільки доставити товар, а й повернутися на склад своїм ходом. По-четверте, залишається відкритим питання надійного кріплення товару до БПЛА, висоти польоту БПЛА в умовах міської забудови, організації безпеки польотів декількох БПЛА в одному середовищі.

У разі використання наземного МР ситуація може бути якісно покращена за всіма параметрами. МР також має обмеженням у вантажопідйомності, але вантажопідйомність у БПЛА значно нижча. У МР може бути поміщено одразу декілька товарів для доставки, що дає змогу МР виконувати доставку відразу в

декілька кінцевих точок, тим самим відповідаючи ідеології Last mile delivery.

Інша значна перевага - БНР забезпечений живленням від повнорозмірного свинцево-кислотного акумулятора, що має час роботи приблизно в 10 годин (наприклад, у БНР типу «Junior» встановлено свинцево-кислотний акумулятор Delta DTM 1233 L з ємністю 33Ампер*годин із 10 годинним розрядом). За такий період часу МР може виконати доставку в десятки кінцевих точок, замінюючи використання декількох БПЛА із поодинокими польотами і постійною підзарядкою. БНР «Junior» є малогабаритним МР, що дозволяє легко об'їжджати перешкоди на шляху, проїжджати різні вузькі простори, виконувати паркування всередині невеликої площі. Таким чином, використання наземного МР має економічну та логістичну вигоду порівняно з розробкою доставок за допомогою використання БПЛА [1-3].

Проект передбачає використання МР типу «Junior» в якості безпілотного ТЗ для доставки товарів у декілька кінцевих точок призначення.

На складі в МР будуть завантажені товари для доставки, в системі керування МР вказані точки доставки. Далі система керування обробляє точки доставок і планує оптимальний маршрут для МР. Статус доставки товару оновлюватиметься в застосунку в споживача, який очікує на доставку замовлення. Передбачається, що система керування МР оновлюватиме статус замовлення («в дорозі», «очікує доставки», «очікує клієнта», «доставлено клієнту» тощо), розраховувати приблизний час доставки, а також повідомлятиме про прибуття в точку доставки в додатку споживача. Після завершення завдання МР буде повертатись на центральний склад.

Середовище роботи БНР «Junior». На поточний момент залишається відкритим питання вибору середовища, в якому буде функціонувати безпілотний ТЗ. Використання безпілотного ТЗ у середовищі міської забудови досі є небезпечним і правове регулювання даного питання відсутнє. Але в якості прототипу пропонується використання БНР «Junior» на проїжджій частині із дотриманням норм та правил ПДР. Причини даного рішення наступні:

- БНР «Junior», що функціонує поза проїжджою частиною міста (наприклад, на тротуарах), є потенційною загрозою для будь-яких живих об'єктів, які перебувають поруч із БНР «Junior»; дана загроза набагато вища, ніж БНР «Junior», що функціонує на проїжджій частині міста;

- БНР «Junior», що виконує доставку в конкретну точку не проїжджою частиною, має ризик збільшити в кілька разів час доставки через об'їзд численних перешкод на шляху (наприклад, пішоходів), а також через рух неоптимізованим, надлишковим шляхом;

- БНР «Junior», що виконує доставку в конкретну точку не проїжджою частиною, може втратити керування чи вийти з ладу чи просто застрягти в пошкоджених ділянках поверхні, по якій пересувається БНР (наприклад, по тротуару, що має різкі перепади або пошкоджену поверхню).

Також важливо зазначити, що БНР «JUNIOR» на проїжджій частині - це найкраще вирішення для забезпечення безпеки оточуючих живих об'єктів (тварин чи людей), оскільки проїжджа частина ізольована від більшої частини таких об'єктів. Ізолювання є не єдиним аргументом на користь безпеки БНР – наприклад, «JUNIOR» може розвинути лише невелику швидкість руху – до 15 км/год, в майбутньому планується збільшити дану швидкість до 20 км/год, БНР не буде реальною загрозою для життя людини. Не можна стверджувати, що за такої невеликої швидкості повністю виключається небезпека використання даного БНР, однак за належного розроблення та тестування руху МР можливо мінімізувати даний фактор небезпеки для людей.

Під час розвитку ідеї використання БНР «JUNIOR» були висунуті наступні вимоги до вирішення проблеми Last mile delivery.

- Система планування маршруту повинна створити маршрут руху БНР «JUNIOR» максимально оптимальним за часом та витраченими ресурсами МР (заряду акумулятора).

- Інформаційна система (ІС) зі створення замовлень доставки повинна інформувати споживачів про зміну статусу замовлення через спеціальний

додаток. Статуси: «в дорозі», «очікує доставки», «очікує клієнта», «доставлено клієнту» тощо. Додатково, ІС має інформувати споживача про приблизний час доставки замовлення.

- БНР «JUNIOR» повинен мати достатньо внутрішнього простору для розміщення товарів (приблизно 10-15 кг навантаження).

- БНР «JUNIOR» повинен бути оснащений відповідним акумулятором, що дає можливість автономної роботи на рівні 10 - 15 годин без підзарядки.

- БНР «JUNIOR» повинен розвивати швидкість руху до 20 км/год для виключення фактора перешкоди МР на проїжджій частині міста.

- БНР «JUNIOR» повинен мати автономний механізм відкриття верхньої кришки для отримання доступу до товарів усередині МР.

Обмеження, що накладаються на виконання проєкту. За наявної комплектації БНР «JUNIOR» МР не може вмістити в себе велику кількість товарів. Внаслідок цього необхідно провести аналіз розташування внутрішніх апаратних компонентів МР, припустити нове розташування і потім обчислити кількість вільного місця, що утворилося. Знання величин вільного простору дає можливість вирішити завдання про рюкзак для даного МР.

Наступним обмеженням є необхідність присутності споживача у кінцевій точці під час доставки замовленого товару. БНР «Junior» може підїхати до кінцевої точки, але вивантажити потрібний товар зможе тільки споживач. Альтернативним рішенням, яке б повністю звільнило споживача від участі в даному процесі - використання БПЛА, який міг би здійснювати короткі вильоти з відсіку БНР «JUNIOR» із обраним товаром і доставку до дверей споживача (будинку, під'їзду чи воріт).

Останнім обмеженням є чутливість МР до поверхні дороги. Для МР необхідна середня рівна поверхня, щоб уникнути можливих пошкоджень і забезпечення безперешкодного пересування по поверхні.

Наразі існує декілька стартапів, зосереджених на вирішенні завдання Last mile delivery. Більшість даних проєктів розвиваються у країнах Західної Європи,

США та Китаї.

Наприклад, проєкт Marble [15] представляє доставку товарів за допомогою використання малорозмірного колісного МР, який пересувається тротуаром (рис. 2.1, а). Проєкт наразі розвивається в місті Сан-Франциско, США, серед інвесторів є китайський гігант Tencent. Створена в 2015 році, компанія Marble [15] працює над створенням повністю автономних МР для доставки, які можуть орієнтуватися в динамічному середовищі міської забудови.



Рисунок 2.1 – Мобільні роботи для автономної доставки виробів:
а – проєкт Marble [15]; б - проєкт Nuro [40]; в – проєкт Starship [16]; г -
проєкт Robby [41]

МР Marble [15] використовує камери, лазерний далекомір та 3D карти міста з високою роздільною здатністю для власної навігації. У той час як компанія вперше провела випробування своїх МР у співпраці зі службою доставки їжі Eat24, роботи Marble [15] мають змінні вантажні відсіки для перевезення різних видів товарів: від продуктів харчування до ліків.

Наступний проєкт Nuro [40] (рис. 2.1, б) належить до категорії великогабаритних доставочних МР у Кремнієвій долині, США. Засновники проєкту - колишні інженери-автогонщики компанії Google, МР Nuro [40] також оснащений необхідними для навігації давачами: радаром, лазерним далекоміром, камерою. Таким чином, вартість використання обладнання буде значно нижчою, ніж вартість використання звичайного автомобіля з автоматичним керуванням.

Проєкт Starship [16] засновано 2014 року (рис. 2.1, в). Лондонська компанія Starship Technologies [16] (частина інженерних підрозділів в Естонії) є однією з відомих компаній-постачальників МР для автоматизованої доставки вантажів. Компанія Starship [16], заснована творцями Skype, створила своїх малорозмірних МР для автономної доставки. Споживачі замовляють товари через додаток, який дає змогу відстежувати стан доставки МР. МР Starship оснащений шістьма колесами, мають дальність дії дві милі (приблизно 3,22 км), доставляють вантаж протягом півгодини.

Проєкт Robby 2 [41] - інший проєкт створений у Кремнієвій долині, США (рис. 2.1, г). Пробіг МР Robby 2 [41] складає понад 20 миль (приблизно 32,2 км) без підзарядки, робоча область МР становить вісім міст Каліфорнії, США. Робот Robby 2 [41] укомплектований потужними інфрачервоними камерами, які сприяють безпечній навігації в темний час доби. МР Robby 2 [41] спроектований таким чином, щоб він міг виконувати свої завдання незалежно від погодних умов навколишнього середовища - його електроніка перебуває під надійним захистом від вологи.

МР Savioke, ще один стартап створений у Кремнієвій долині, який використовує МР (рис.2.2) для доставки кави та різних предметів всередині приміщень: виробничих приміщеннях, у готелях, медичних установах.



Рисунок 2.2 – Загальний вигляд МР Nuro (ліворуч) та МР Savioke (праворуч) для автономної доставки товарів

2.3 Програмне забезпечення для керування автономним рухом мобільного робота

Robot Operating System (ROS) – досить популярний відкритий фреймворк для розроблення ПЗ для МР [42]. Найменшою одиницею середовища є так званий package, який повинен вирішувати певне завдання. Згідно з ідеологією, ROS package можуть бути поліпшені будь-яким розробником, а нові package - додані до загальної бази знань з описом інструкції з використання та складових модулів ROS. Кожен package запускає власний процес з індивідуальним найменуванням.

Процес у ROS називається node або вузол. Основна особливість ROS полягає в архітектурі у вигляді графів, де вузлом є node, створювана package, а передача даних між node здійснюється через topic. Під передачею даних ROS має на увазі передачу messages від одного процесу іншому за допомогою використання topic. Кожен процес може публікувати свої topic, відправляючи на них messages з даними, а також підписуватися на інші topic, так би мовити «слухаючи» повідомлення від інших процесів. Своєю чергою, кожна topic

підтримує тільки один тип переданих messages.

ROS в свою чергу надає наступні стандартні дані типів повідомлень [42]:

- Int8, int16, int32, int64, uint8, uint16, uint32, uint64 - цілочисельний тип даних із різною розрядністю;
- Time - тип даних час, виражений через 32-розрядні числа без знаку;
- Bool - логічний тип даних;
- String - рядковий тип даних;
- Float32, float64 - число з плаваючою комою з різною розрядністю;
- Duration - тип даних проміжку, виражений через 32-розрядні числа зі знаком.

Додатково, ROS [42] містить свої власні спеціалізовані повідомлення, які необхідні для програмування МР. Наприклад, пакет turtlesim_node, що міститься в системі ROS [42], реалізує симуляцію МР у вигляді черепахи (черепаха - символ ROS системи). Даний пакет підписаний на тему turtleX/cmd_vel, яка публікує повідомлення типу geometry_msgs/Twist (одні з власних повідомлень ROS), що описують лінійну та кутову швидкість черепахи TurtleX. Повідомлення geometry_msgs/Twist складається з наступних компонентів [42]:

- geometry_msgs/Vector3 angular: float64 x, float64 y, float64z (опис кутової швидкості);
- geometry_msgs/Vector3 linear: float64 x, float64 y, float64z (опис лінійної швидкості).

Інша можливість ROS [42] - сумісність із різними типами симуляторів. Це дає змогу розробникам налаштовувати свої ROS [42] пакети після проведення експериментів у режимі симуляції.

Gazebo [43] - відкритий 3D-симулятор, інтегрований з ROS. Gazebo [43] надає точну симуляцію різних типів існуючих МР (маніпулятори, БНР, БПЛА), а також дозволяє створювати власні моделі МР. Симулятор враховує задану фізику світу та моделі, що надає можливість проводити експерименти (наприклад, верифікацію нових алгоритмів) з МР в режимі симуляції. Також

симулятор надає можливість створювати бажане середовище для МР, додавати різні перешкоди та інші об'єкти. На додачу, Gazebo [43] надає різноманітні плагіни, які симулюють різні типи давачів. Формат опису моделі в Gazebo [43] представлений у двох способах - SDF (Simulation Description Format) і URDF (Unified Robot Description Format). Симулятор підтримує STL та DAE формати полігональних сіток для об'єктів.

Rviz [44] - засіб для виконання 3D-візуалізації даних (повідомлень), що надходять на теми (шини для передачі певного типу даних) ROS. Візуалізуючи дані, розробники можуть оцінити, як МР «бачить» навколишнє середовище і як він сприймає себе в ньому - власне положення, орієнтацію в просторі, а також місце розташування на карті. Rviz [44] допомагає у налагодженні програмних компонентів ROS [42] і є незамінним інструментом під час роботи з МР як у реальному оточенні, так і в симуляції. Графічний інтерфейс ПЗ представлений на рисунку 2.3.

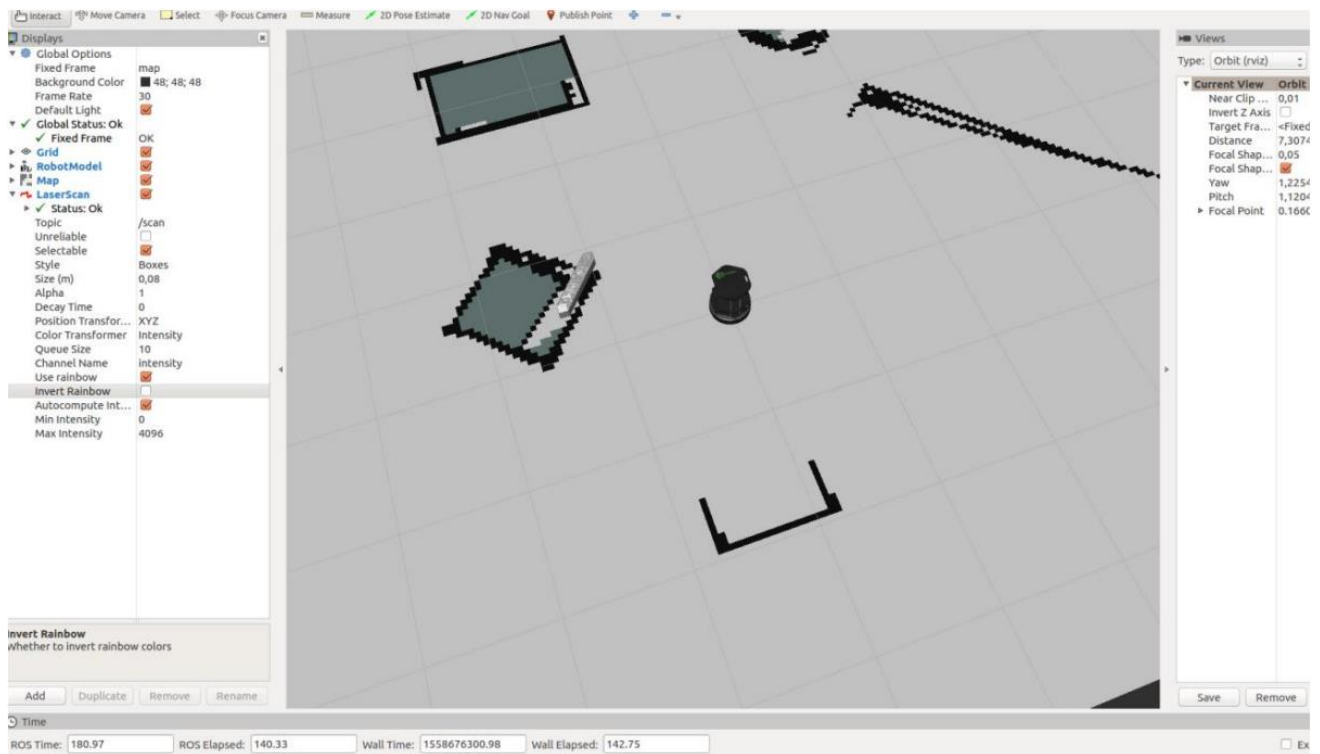


Рисунок 2.3 – Загальний вигляд графічного інтерфейсу Rviz [44]

2.4 Висновки до другого розділу

В даному розділі виконано моделювання системи керування переміщенням вантажу із використанням безпілотних транспортних засобів, а саме охарактеризовано систему керування ланцюгами переміщення виробів, запропоновано ідею вирішення проблеми Last mile delivery, проаналізовано програмне забезпечення для керування автономним рухом мобільного робота.

3 МОДЕЛЮВАННЯ РОБОТИ БЕЗПЛОТНОГО ТРАНСПОРТНОГО ЗАСОБУ

В останні десятиліття широке використання симуляцій стало невіддільною частиною досліджень у галузі робототехніки [45-48]. Використання симуляторів допомагає здешевити та прискорити розробку робото-технічних систем (РТС), є безпечнішим інструментом первинної верифікації ідей та алгоритмів, а також надає можливість виконувати тестування нових концепції та алгоритмів, незалежно від наявності необхідного обладнання у наукової групи. Моделювання може використовуватись для всіх типів РТС та завдань на початкових етапах розвитку проекту.

Симулятори стали прогресивними інструментами, які надають можливість виконати відтворення складних середовищ та використовувати визначувану користувачем фізику, що, в свою чергу, дає можливість створювати моделі РТС, які поведуться максимально близько до реальної поведінки МР. Gazebo є одним із найпопулярніших 3D-симуляторів МР, який успішно використовується для симуляції БПЛА [49] та БНР [50-53], та для проведення різноманітних експериментів, включно з тестуванням базових рухів МР, плануванням траєкторії шляху та колаборативною взаємодією з іншими МР [54], експериментами з маніпуляторами [55] та моделюванням сценаріїв пошуково-рятувальних операцій [56]. Інші спеціалізовані симулятори (наприклад, UWSim [57], V-REP [58]) дають змогу моделювати водне середовище для автономних підводних МР.

У рамках поставленого технічного завдання зі створення системи навігації, розроблення алгоритмів переміщення та автономної навігації для МР «Junior» було побудовано модель у симуляторі Gazebo і проведено тестування алгоритмів керування.

Моделювання робомобіля було виконано в середовищі ROS Gazebo. Підхід до моделювання був наступний: врахування фізичних характеристик МР,

врахування давачів МР та характеристик, використання САД-моделей МР для моделювання.

При створенні моделі використовувалися наступні середовища для програмування: ROS, Gazebo [43].

Використані наступні інструменти: САПР Solid Works. Формат опису моделі: URDF.

В якості прикладного САПР було використано ПЗ Solid Works. У задачі моделювання САПР використали для експорту необхідних САПР-моделі в програмне середовище Gazebo [43] у форматі STL. Також у САПР були задіяні інструменти вимірювання певних деталей МР для побудови більш точної моделі у симуляції. Для опису моделі у Gazebo [43] був використано формат Unified Robot Description Format (URDF). URDF є файлом формату XML, який описує МР за допомогою певних перемінних. Існують основні перемінні, які необхідні для опису візуальної структури МР (так би мовити «скелета»): link - з'єднання (зчленування) та joint або суглоб. Зчленування з'єднуються між собою за допомогою суглобів, являючи собою деревоподібну структуру моделі МР. Інші перемінні призначені для опису внутрішніх характеристик зчленувань та суглобів, додавання плагінів давачів та їх налаштування, вказівки контролерів суглобів та характеристик.

3.1 Характеристика мобільного робота AURORA «JUNIOR»

МР «JUNIOR» є розробкою фірми «AURORA ROBOTICS». Загальний вигляд МР представлено на рисунку 3.1. За своєю зовнішньою будовою «JUNIOR» є класичною машиною із Ackermann drive або системою приводів та рульовою рейкою для здійснення керування рухом машини [60]. Даний МР було розроблено з метою навчання школярів та студентів принципів автономного руху БНР. Такий навчально-налагоджувальний комплекс є зручним та оптимальним рішенням для роботи школярів та студентів: невеликі габарити та

загальна вага машини (відносно повнорозмірних машин, які використовуються для розробки автономного функціонування компанії Uber, Google), відкрита програмна система на базі ОС Linux Ubuntu 16.04 LTS та ROS, відкрита апаратна система.

Відкрита апаратна система передбачає вільний фізичний доступ до бортових датчиків машини, влаштування електричного ланцюга машини, можливості зміни апаратного складу машини, вдосконалення та заміни будь-яких компонентів (як датчиків, і базових елементів – акумулятора, коліс, мотор-колес тощо) [59].



Рисунок 3.1 – Загальний вигляд МР «Junior»

З машиною передбачається використання спеціального полігону, побудованого на кшталт реальної проїжджої частини міста з відповідними елементами: світлофорами, дорожніми знаками, тунелем, дорожньою розміткою та пішохідним переходом. Завдяки всьому комплекту даного навчального

комплексу студенти можуть розробляти алгоритми автономного руху машини у штучному середовищі дорожнього оточення [59].

Фізичні характеристики є лінійними розмірами МР «Junior». Ці дані необхідні для створення коректної моделі у симуляторі Gazebo [43]. Точні габаритно-вагові характеристики були запитані у компанії-виробника «AURORA ROBOTICS» та наведені в Додатку В.

3.2 Моделювання основних елементів МР «Junior»

Основними елементами МР «JUNIOR», що представляють модель, були обрані наступні: зовнішній каркас МР - `base_link`, переднє праве колесо - `right_steer_wheel`, переднє ліве колесо - `left_steer_wheel`, права частина рульової рейки, частина майбутнього механізму системи підвіски робота згідно Ackermann drive - `right_steer_link`, ліва частина рульової рейки, частина майбутнього механізму системи підвіски робота згідно з Ackermann - drive `right_drive_left_steer_link`, заднє ліве колесо - `left_drive_wheel`.

Для того, щоб модель МР візуально відповідала реальному роботу, були використані частини МР для даних елементів та деяких давачів, розглянутих нижче (САПР-модель є у вільному доступі) (рис. 3.2).

Усі зчленування моделі були задані у пропорціях, що відповідають пропорціям та розмірам реального МР. Щоб адаптувати інерційні характеристики МР до моделі, було розраховано спрощений інерційний тензор на зчленування. У процесі було замінено складну структуру каркаса МР (`base_link`) суцільним кубоїдним об'єктом (`solid cuboid`); структури рульових коліс МР (`right_steer_wheel`, `left_steer_wheel`), задніх коліс замінені на суцільний циліндр (`solid cylinder`) для обчислення наближених тензорів інерції зчленувань моделі. Колізії зчленувань були зазначені з урахуванням структури всіх зчленувань і в тому ж розмірі, що й зчленування. Файли моделювання МР наведені у Додатку Г.

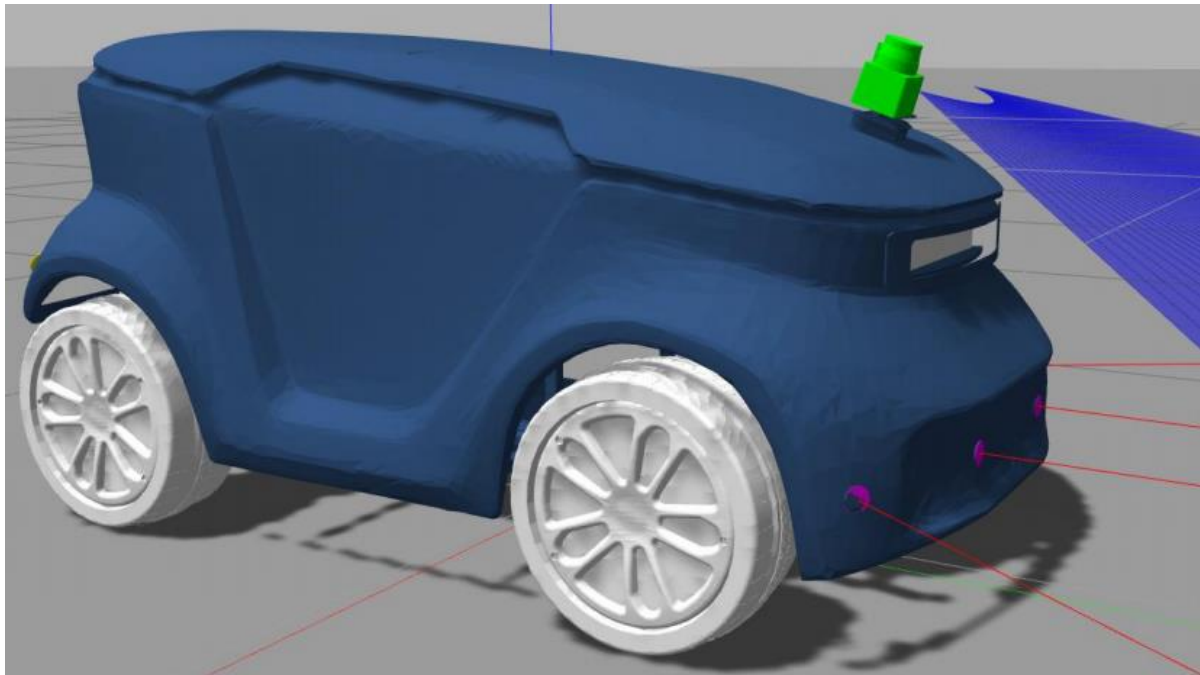


Рисунок 3.2 – Загальний вигляд моделі МР «Junior»

Моделювання суглобів, моторів та трансмісія МР.

URDF формат дозволяє описувати наступні типи суглобів: призматичний (prismatic) – ковзний суглоб, що зміщується вздовж вказаної осі та володіє обмеженим діапазоном руху, заданими верхньою та нижньою границями, обертальний без лімітів (continuous) – обертальний суглоб (або шарнірний), який обертається щодо вказаної осі та не має верхніх та нижніх границь, плаваючий (floating) – суглоб, що володіє рухом у всіх шести ступенях свободи, обертальний з лімітами (revolute) – обертальний суглоб (шарнірний), який обертається щодо заданої осі та має обмежений діапазон руху, заданий верхньою та нижньою границею і нерухомий (fixed) – фіксований суглоб, який насправді не є суглобом. Усі ступеня свободи даного типу суглоба заблоковані, він вимагає вказівки осі руху, границь. Даний тип суглобу використовується для жорсткого зчеплення зчленування МР між собою, а також приєднання давачів МР до його зчленування.

Обертальний тип суглоба без обмежень був використаний до суглобів передніх та задніх коліс (left_steer_wheel_joint, left_drive_wheel_joint,

right_steer_wheel_joint, right_drive_wheel_joint). Обертальний тип суглоба із обмеженням був використано до кермових механізмів моделі (left_steer_joint and right_steer_joint), для подальшої розробки системи підвіски та керування моделлю згідно з принципом Акерманна [60].

Так як на реальному МР використовуються два мотори для виконання руху задніх коліс та один мотор для організації рульового механізму, були налаштовані наступні ROS-контролери для суглобів: SteerRight_controller і SteerLeft_controller - контролери положення (або ROS JointPositionController), EffortDriveRight_Control - контролери зусиль.

Контролери положення відносяться до суглобів кермового механізму, контролери зусиль відносяться до суглобам задніх коліс (оскільки на реальному МР моторами оснащені два задніх колеса). Рисунок 3.3 демонструє фінальну схему роботи контролерів серед ROS.

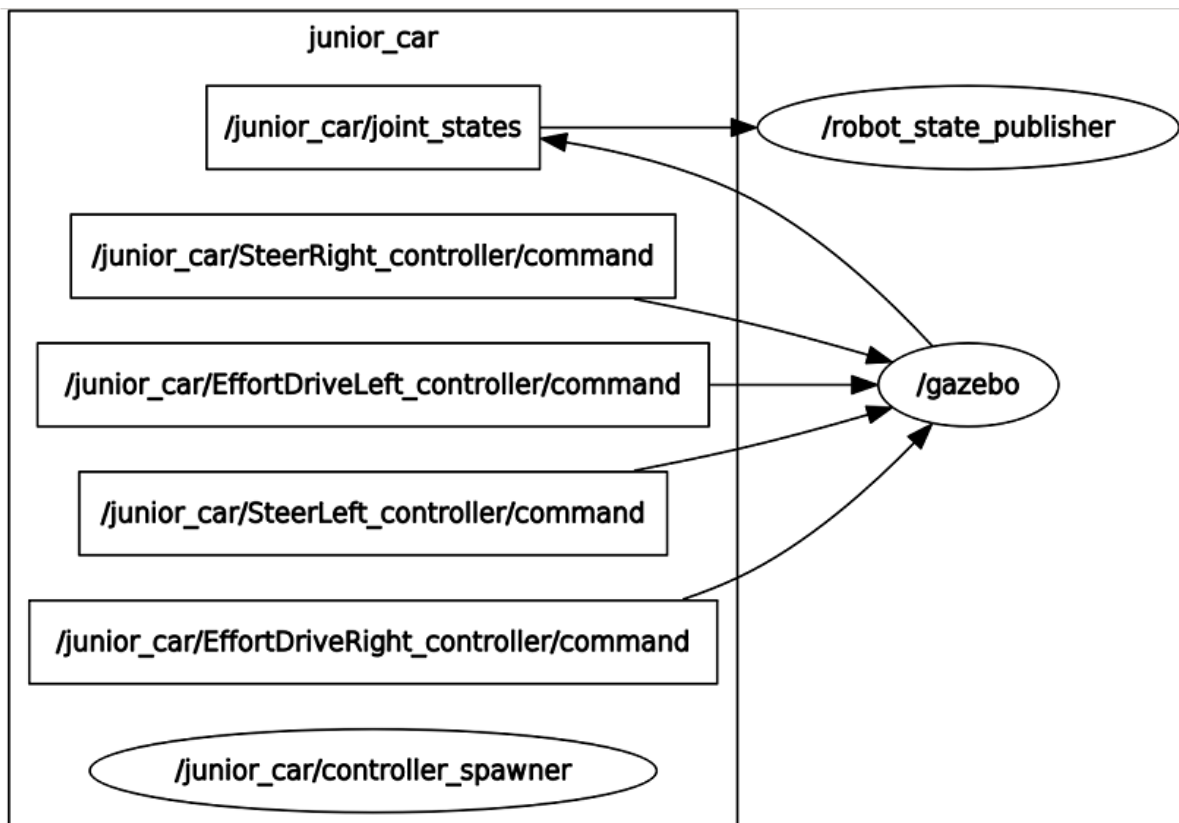


Рисунок 3.3 – Загальний вигляд візуалізації в rqt_graph тем повідомлень моделі МР «Junior» (теми повідомлень давачів не відображені)

Моделювання давачів наведено додатку В, табл. В.3 та представляє список давачів МР «Junior» та їх імплементація в програмних середовищах ROS та Gazebo.

Моделі давачів були поміщені в тому ж положенні та орієнтації, як і на реальному МР, для отримання точних даних із давачів. Всі плагіни давачів були налаштовані згідно з технічним описом існуючих давачів. Для візуалізації давачів використовувалося ПЗ Rviz (входить у фреймворк ROS).

3.3 Налаштування ПД регуляторів моделі МР «Junior»

Етап налаштування ПД регуляторів є продовженням роботи з моделювання МР «Junior». При описі контролерів моделі використовувався конфігураційний файл, який вказував ROS і Gazebo тип контролера, що використовується (в даному випадку - position controller, effort controller) і параметри ПД регулятора (інтегральний i , пропорційний p та диференціальний d коефіцієнти контролера).

У загальних випадках моделі МР можуть використовуватися коефіцієнти за замовчуванням такі як: $i = 0.01$, $p = 100.0$, $d = 10.0$; у разі створення конкретної моделі МР необхідне індивідуальне налаштування параметрів контролерів.

Загально відомо, що ПД-контролер (або регулятор) - пристрій, який підтримує певні параметри ОК на заданому (бажаному) рівні. Регулятор слідкує за станом ОК і створює впливи керування для того, щоб забезпечити стабільність параметра регулювання. ПД-регулятор розшифровується як пропорційно-інтегрально-диференціальним регулятором, відповідає за створення вихідного сигналу, що є алгебраїчною сумою трьох складових із різними передаточними характеристиками. У формуванні вихідного сигналу ПД регулятора приймають участь наступні коефіцієнти:

- пропорційний - значення пропорційне помилці неузгодженості (різниці заданого і реального значень регульованого параметра);

- диференціальний - похідна помилки;
- інтегруючий - інтеграл помилки.

Всю роботу ПІД-регулятора можливо описати наступними кроками (рис. 3.4). Перший крок - вимірювання реального значення регульованого параметру. Другий крок - обрахунок помилки неузгодженості, величина помилки надходить на вхід до складових регулятора. Третій крок - у результаті суми складових утворюється вплив керування, який подається на елемент регулювання. Регулювання параметрів системи може відбуватися за допомогою однієї (Р або І регулятор), двох (PI або PD регулятор) або усіх трьох складових (PID регулятор).

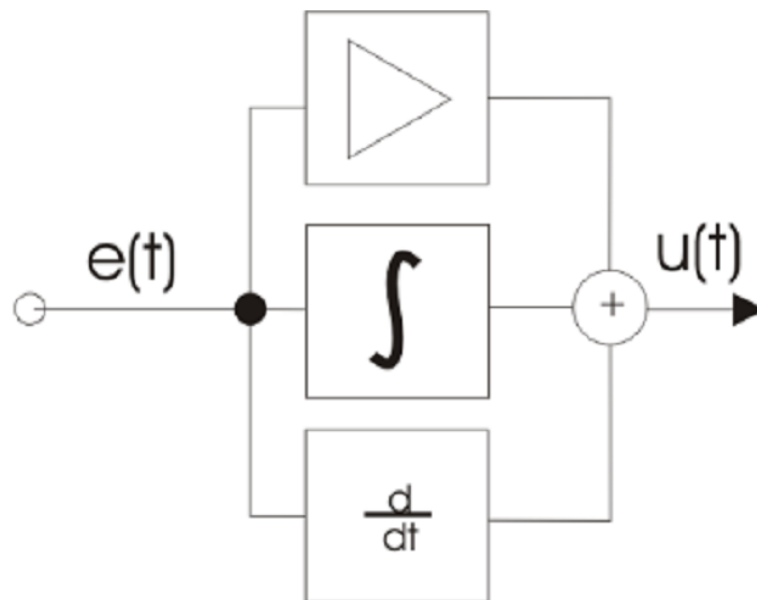


Рисунок 3.4 – Загальний вигляд структурної схеми ПІД-регулятора

Р (або пропорційна) складова. Вихідний сигнал регулятора компенсує відхилення регульованого параметра, тобто вихідний сигнал зростає, якщо помилка неузгодження зростає; якщо помилка рівна нулю, тоді вихідний сигнал регулятора також буде рівним нулю. Іншими словами, пропорційна складова орієнтується на реальне значення, тобто реагує на поточну зміну в системі керування [61-63].

І (або інтегрувальна) складова - у вихідному сигналі помилка

неузгодження буде множитись на коефіцієнт складової та додаватись до попереднього значення сигналу з інтегрувальної ланки. Таким чином, вихідний сигнал весь час накопичується і з часом збільшує власний вплив на всю систему керування. Іншими словами, інтегрувальна складова орієнтується на «минуле», тобто поступово впливає на систему керування, орієнтуючись на минулі стани [61-63].

D (або диференціувальна) складова – відповідає за передбачення відхилення регульованого параметра в системі керування і в майбутньому протидіє даному відхиленню.

Налаштування ПІД-регуляторів є досить складним завданням. У багатьох практичних випадках коефіцієнти ПІД-регулятора визначається методом підбору. Якість налаштування ПІД-регулятора можна визначити за графіком зміни регульованого параметра в часі.

Наведемо загальну концепцію налаштування ПІД-регулятора. Всі складові регулятора налаштовуються окремо. Спочатку відключаються (прирівнюються до нуля) диференційний та інтегрувальний коефіцієнти. Тоді виконується налаштування пропорційного коефіцієнту. Потім виконується налаштування коефіцієнту, що інтегрує, який має мінімізувати коливання параметра регулювання. Останнім (за потреби) виконується налаштування диференціувального коефіцієнту. У більшості випадків ПІД-регулятор налаштовується ітераційно, тобто доти, доки не буде досягнуто бажаного результату дії ПІД-регулятора на регульований параметр системи керування.

Відомо метод Зіглера-Нікольса, який є евристичним методом налаштування ПІД-регулятора [61-64]. Послідовність налаштування ПІД-регулятора згідно з методом, наступна:

- коефіцієнт диференціальної та інтегральної (I, D) складових прирівнюється до нуля;
- коефіцієнт пропорційної складової K_p зростає з нуля доти, доки не буде досягнуто значення граничного коефіцієнта посилення (або ultimate gain) K_u , за

якого на виході контуру керування виникають послідовні та стабільні коливання;

- розраховується значення періоду T_u отриманих послідовних і стабільних коливань;

- K_u та T_u використовують для встановлення коефіцієнтів P , I , D в залежності від типу використовуваного ПД-регулятора, результат наведено в Додатках.

Для налаштування реальних ПД параметрів моделі було використано метод Зіглера-Нікольса описаний вище. Перед налаштуванням необхідно було запустити наступні модулі ROS: `rqt_gui`, `rqt_recognifure`, а також запустити симуляцію моделі в Gazebo [43] і роботу контролерів. Інтерфейс `rqt_gui` та підключених плагінів зображено на рис. 3.5. Інтерфейс `rqt_recognifure` представлено на рис. 3.6. Відомо, що пакет `rqt_gui` надає зручний спосіб (GUI інтерфейс) для надсилання повідомлень (команд) на контролери моделі, а також за допомогою підключеного плагіна `Mat_plot` відстежувати стан контролера під час надсилання повідомлень. Модуль `rqt_recognifure` надає інтерфейс користувача для налаштування ПД параметрів усіх контролерів моделі.

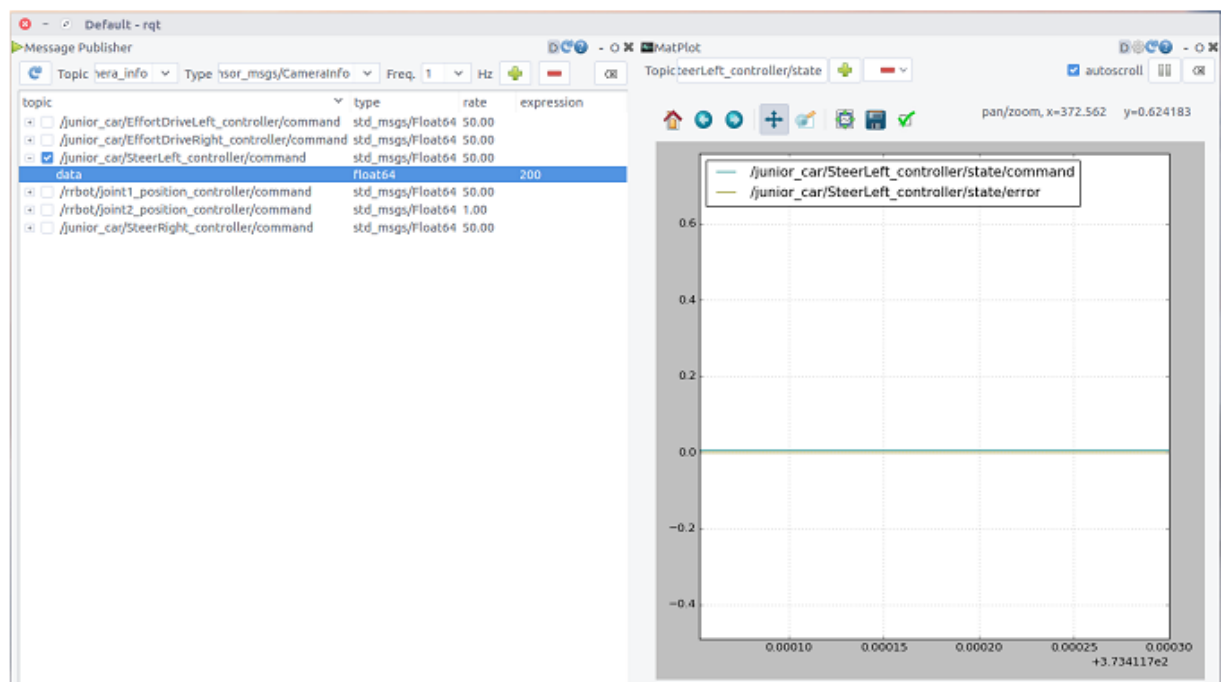


Рисунок 3.5 – Загальний вигляд модуля `rqt_gui` в середовищі ROS

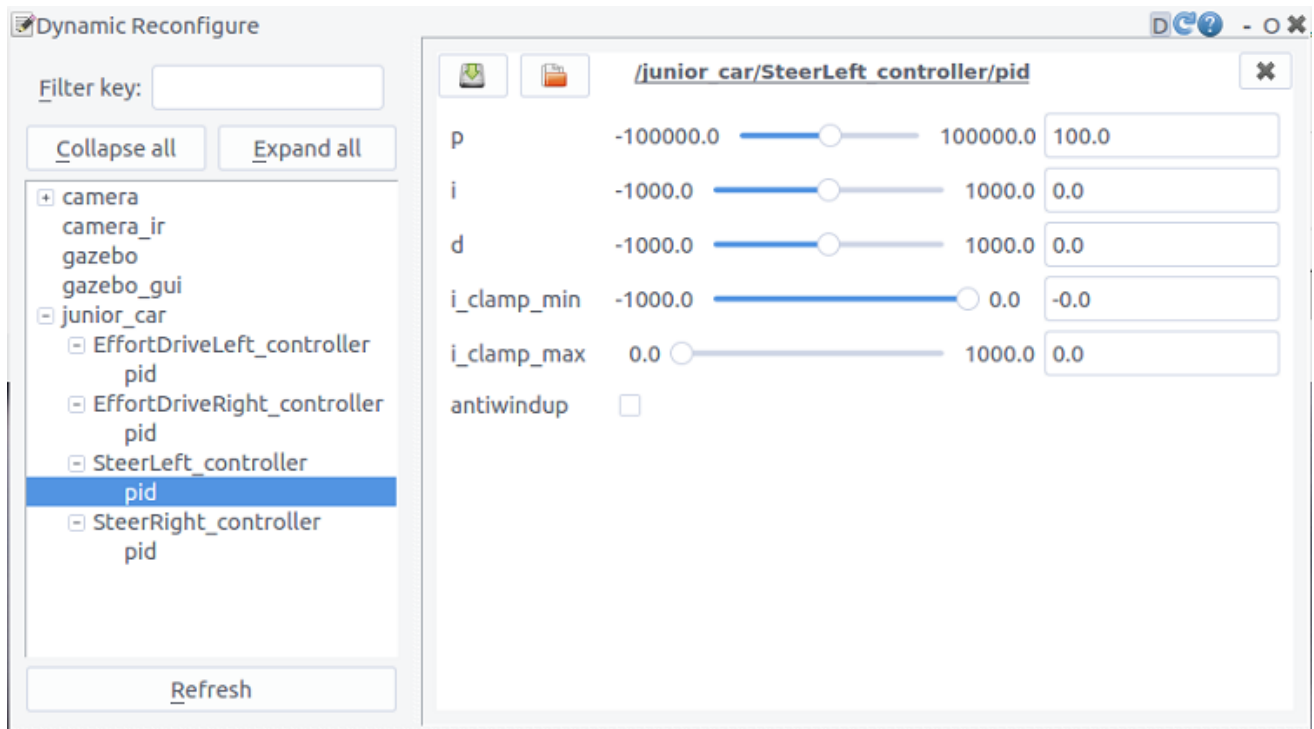


Рисунок 3.6 – Загальний вигляд модуля `rqt_reconfigure` в середовищі ROS

Згідно з відомим методом Зіглера-Нікольса, складові I та D були прирівняні до нуля, а складова P регулятора поступово зростала. Після досягнення граничного коефіцієнта посилення K_u , представлено на графіку у вигляді стабільних коливань (рис. 3.7). Після обчислення періоду коливань, отримані дані використовували для розрахунку коефіцієнтів K_p , T_i , T_d , згідно з методом Зіглера-Нікольса (див. Додаток В, табл. В.4). У підсумку було отримано наступні значення коефіцієнтів для контролерів положення рульового механізму (`SteerRight_controller` та `SteerLeft_controller`):

- Пропорційна складова $K_p=180.0$;
- $K_u = 300.0$;
- $T_u = 3$ с;
- інтегральна складова $T_i=1.5$;
- диференціальна складова $D_i=0.375$.

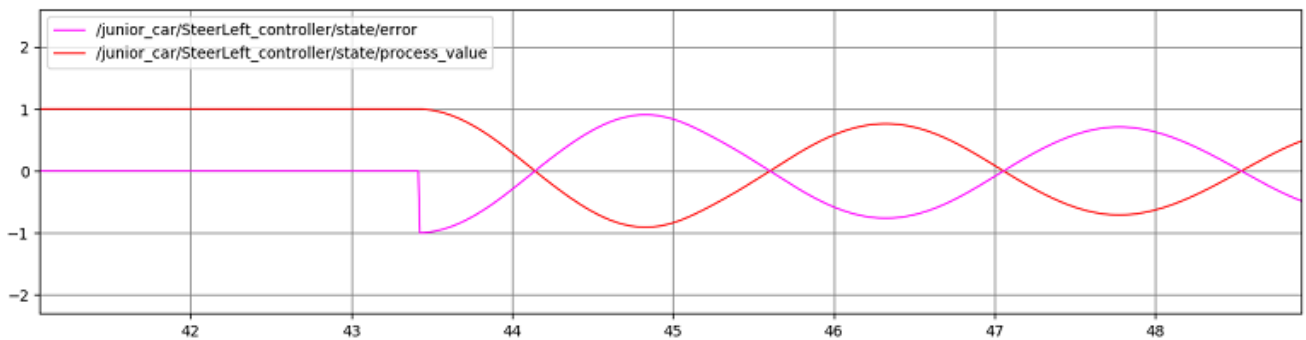


Рисунок 3.7 - Процес налаштування ПІД-регулятора:

рожевим кольором візуалізовано поточну помилку, червоним - безпосереднє значення величини суглоба (його позиція) в поточний момент часу [61-63]

Наступним етапом роботи знадобилося додаткове налаштування ПІД-регуляторів для отримання більш точних значень коефіцієнтів ПІД-регулятора. Для цього аналогічно було використано `rqt_gui`, `rqt_recognifure`, модель, запущена в середовищі Gazebo. Під час додаткового налаштування за допомогою `rqt_recognifure` по черзі було налаштовано всі коефіцієнти ПІД-регулятора. Точність налаштування визначалася наявністю стабільної поведінки системи керування і часом загасання коливань. Зовні це перевіряли поведінкою моделі МР в середовищі Gazebo: за заданого на контролер значення суглоб мав прийти в зазначене йому положення за допустимий встановлений час (час загасання коливань). Оскільки під час надсилання повідомлень на рульову рейку реального МР, його передні колеса повертаються за 1 секунду, цей час було прийнято як допустимий час загасання коливань для моделі МР. Остаточні значення ПІД-регулятора для передніх коліс, що задовольняють встановленим умовам, наступні:

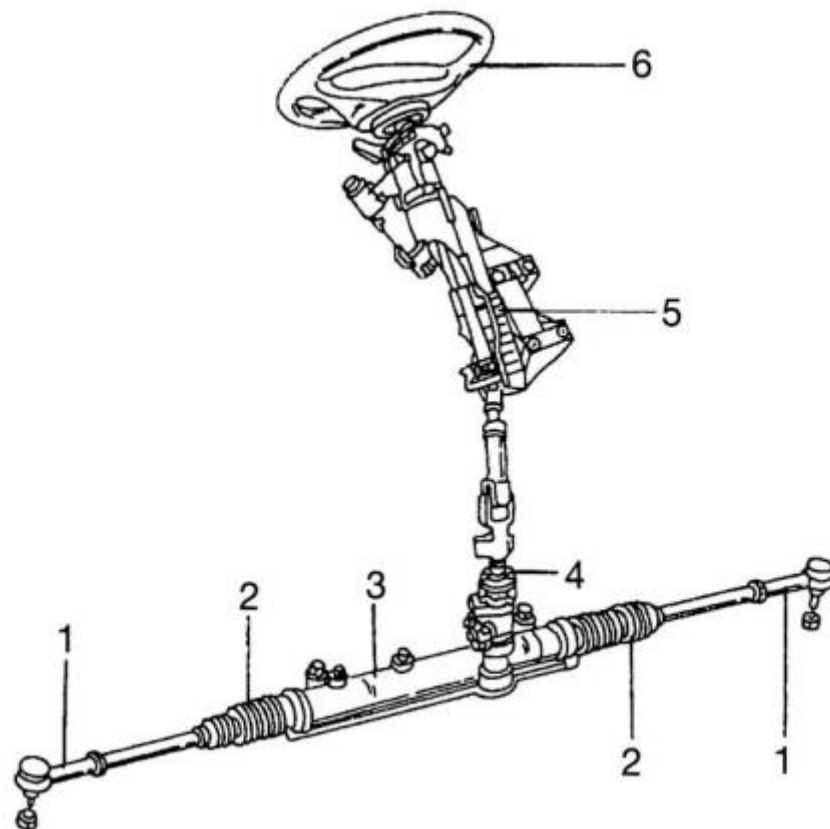
- диференціальна складова $D_i=14.375$;
- інтегральна складова $T_i=50.5$;
- пропорційна складова $K_p=380.0$.

3.4 Організація контролю моделі мобільного роботу «Junior»

Виконаємо опис принципу рульового керування та геометрії Аккерманна [60] з реалізацією в середовищі ROS для віртуальної моделі МР «Junior».

3.4.1 Принцип рульового керування та геометрії Аккерманна [60]

МР «Junior» має систему рульового керування, яка є аналогічною системі повнорозмірних машин із заднім приводом. У МР «Junior» приводними колесами є задні колеса, а передні - відповідають за напрямок руху МР. У разі повнорозмірних автомобілів, рульове керування здійснюється за наступною загальною схемою (рис. 3.8).



1 - кермові поперечні тяги; 2 – манжети; 3 - рейка керма; 4 - еластична муфта;
5 - рульова колонка; 6 - рульове колесо

Рисунок 3.8 – Загальний вигляд системи рульового керування автомобілем:

За допомогою рульового колеса (рис. 3.8, б) виконується задання бажаного напрямку руху ТЗ. У свою чергу, напрямок повороту рульового колеса передається через рульовий вал на рейку керма (рис. 3.8, з). Рух рейки керма сприяє руху рульової поперечної тяги (рис. 3.8, 1), на наконечниках якої розташовано поворотні кулаки. В поворотні кулаки вставляються маточини коліс, гальмівні диски та колеса ТЗ. Поворотні кулаки, в залежності від руху рульової рейки виконують поворот, тим самим спрямовуючи напрямок руху передніх коліс. У МР «Junior» ролі рульового колеса та рульової рейки об'єднані в один рульовий мотор. В залежності від напрямку обертання рульового мотора змінюється коригування напрямку рульових коліс МР. Поворот кермових коліс відбувається відповідно до геометрії Аккермана [60], якої дотримуються в багатьох сучасних ТЗ.

Геометрія Аккермана [60] досягається механічним способом, за допомогою правильної побудови рульової трапеції. За допомогою рульової трапеції передні колеса під час виконання повороту повертаються на різну величину кута; як правило, зовнішнє колесо повертається на менший кут, ніж внутрішнє колесо. Це відбувається для того, щоб зменшити знос внутрішньої шини колеса та уникнення втрати зчеплення з дорогою під час виконання повороту.

3.4.2 Розроблення контролю рульового керування моделі МР

Для реалізації завдання було використано класичний (або загальний) випадок розрахунку геометрії Аккермана [60]. Таким чином, для здійснення обчислень величини кута повороту передніх коліс (α_1 та α_0), необхідно було знати лінійні геометричні параметри МР: відстань між передніми колесами ($T = 0,54\text{м}$) і бічну відстань між переднім та заднім колесом ($L = 0,70\text{м}$). Дані параметри були використані як константи при розрахунку величин кутів повороту коліс.

Згідно із принципом геометрії Аккермана [60], загальний кут α повороту

всього МР може бути визначений наступним співвідношенням:

$$\operatorname{tg} \alpha = \frac{L}{R}, \quad (3.1)$$

де R - величина ортогональної відстані від поздовжньої осі МР до центру повороту (рис. 3.9).

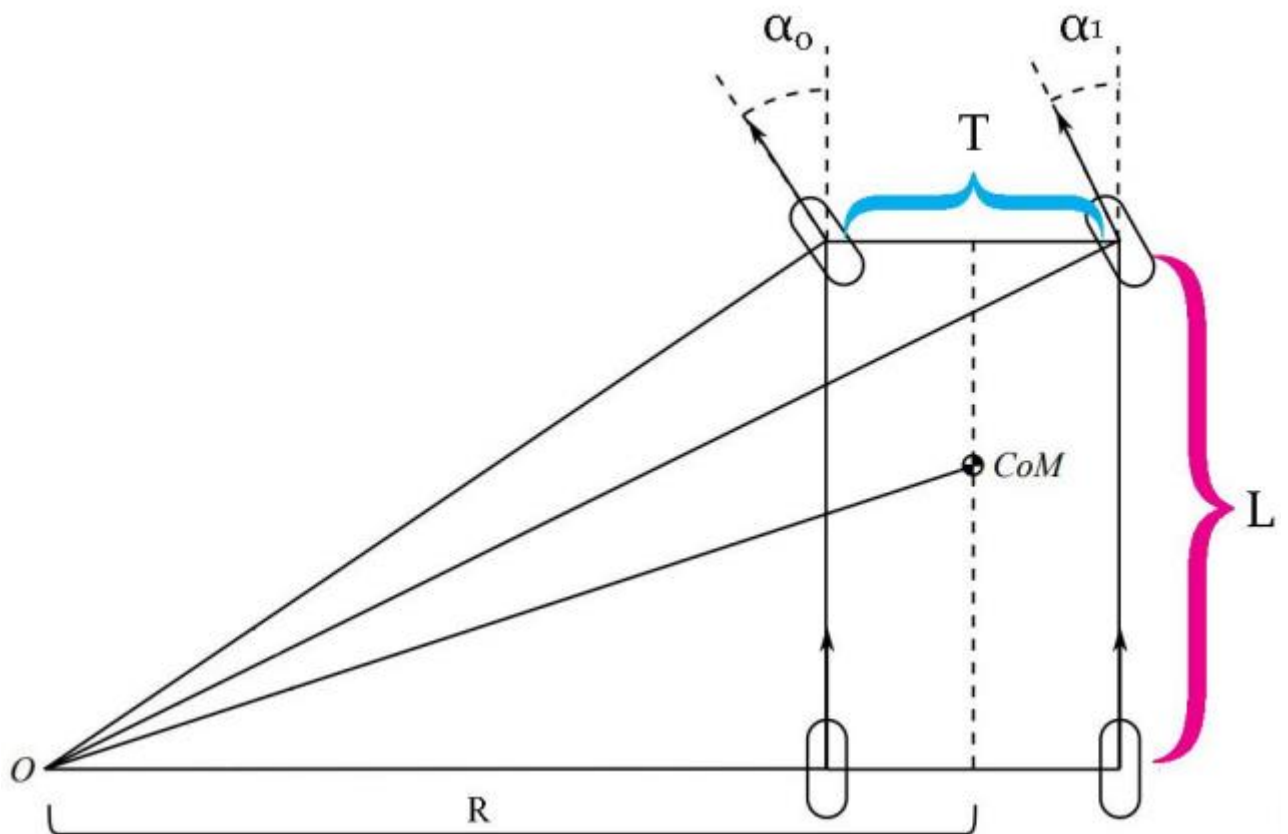


Рисунок 3.9 – Загальний вигляд геометрії Акерманна для ТЗ [60]:

L - бічна відстань між серединами переднього та заднього коліс ТЗ;

R - відстань від поздовжньої осі ТЗ до центру повороту; T - відстань між серединою передньої пари коліс

Величини кутів повороту передніх коліс α_1 та α_0 визначаються з наступних співвідношень:

$$\alpha_1 = \operatorname{tg}^{-1} \left(\frac{L}{R - \frac{T}{2}} \right); \quad (3.2)$$

$$\alpha_0 = \operatorname{tg}^{-1} \left(\frac{L}{R + \frac{T}{2}} \right). \quad (3.3)$$

Своєю чергою, величина кута повороту α МР визначається заздалегідь як вхідне значення для обчислень кутів Аккермана [60]; таким чином, радіус R повороту МР зі співвідношення (3.1) може бути розрахований за наступною формулою:

$$R = \frac{L}{\operatorname{tg} \alpha}. \quad (3.4)$$

Використовуючи дані співвідношення, було змінено контроль передніх рульових коліс моделі в середовищі. Змінений принцип роботи керуванням кермових коліс моделі став наступним: на вхід ноді надходить бажаний кут повороту всієї моделі (α , формула 3.1). Величина кута задається в радіанах, в залежності від вказаного знака («+» або «мінус») визначався напрямок повороту вправо або вліво, відповідно. В залежності від бажаного повороту всієї моделі виконувалось обчислення індивідуальних величин кутів повороту кермових коліс згідно з геометрії Аккермана [60] формули (3.2-3.3); зовнішнє кермове колесо повертається на меншу величину кута, ніж внутрішнє рульове колесо під час виконання повороту моделі.

Під час розрахунку враховувалися лінійні розміри МР «Junior». Вихідний код розробленої *node* представлено в Додатку Г.

3.5 Висновки до третього розділу

В розділі виконано моделювання роботи мобільного робота для переміщення виробів, а саме охарактеризовано мобільного робота AURORA «Junior», виконано моделювання основних елементів МР «Junior».

Крім того проаналізовано налаштування ПД регуляторів моделі МР «Junior», а також організовано контроль моделі мобільного роботу «Junior».

4 ДОСЛІДЖЕННЯ СИСТЕМИ НАВІГАЦІЇ ДЛЯ БЕЗПЛОТНОГО ТРАНСПОРТНОГО ЗАСОБУ «JUNIOR»

Система навігації МР складається із загальних ключових компонентів. Своєю чергою, кожен компонент вимагає виконання індивідуальної конфігурації, яка залежить від типу МР. Для створення системи навігації моделі МР «Junior», знадобилося наступне:

- реалізація всіх обов'язкових компонентів системи навігації;
- зміна ноди контролю кермових коліс згідно з геометрією Акерманна [60];
- створення зв'язку між `cmd_vel` системи навігації та поточних нод керування моделлю МР;
- коригування даних з лазера, встановленого на модель МР;
- налаштування параметрів `move_base`.

4.1 Дослідження системи навігації ROS

Система навігації (СН) - це комплекс взаємопов'язаних модулів, які надають можливість МР будувати карту навколишнього середовища, орієнтуватися в побудованій карті та прораховувати шлях до встановленої на ній `goal point` (або цільової точки). СН отримує інформацію з одометрії, давачів та видає команди швидкості для надсилання їх на мобільну базу МР.

Для правильної роботи СН ROS МР повинен бути налаштований наступним чином (вимоги до апаратного забезпечення МР):

- МР повинен бути диференційно-колісним або голономним. Передбачається, що мобільна база МР керується шляхом надсилання бажаних команд швидкості у наступному вигляді - швидкість - x , швидкість - y , швидкість - θ ;
- лазер повинен бути встановлений на плоскій рівній поверхні, паралельній землі. Лазер використовується для побудови карти та локалізації положення МР;

- СН ROS розроблена для МР квадратної форми, тому продуктивність буде найкращою для МР, близьких до квадратної або круглої форми. СН працює з МР довільної форми та розмірів, але можуть виникнути проблеми для МР прямокутної форми під час виконання навігації у вузьких просторах (наприклад, у дверних проймах).

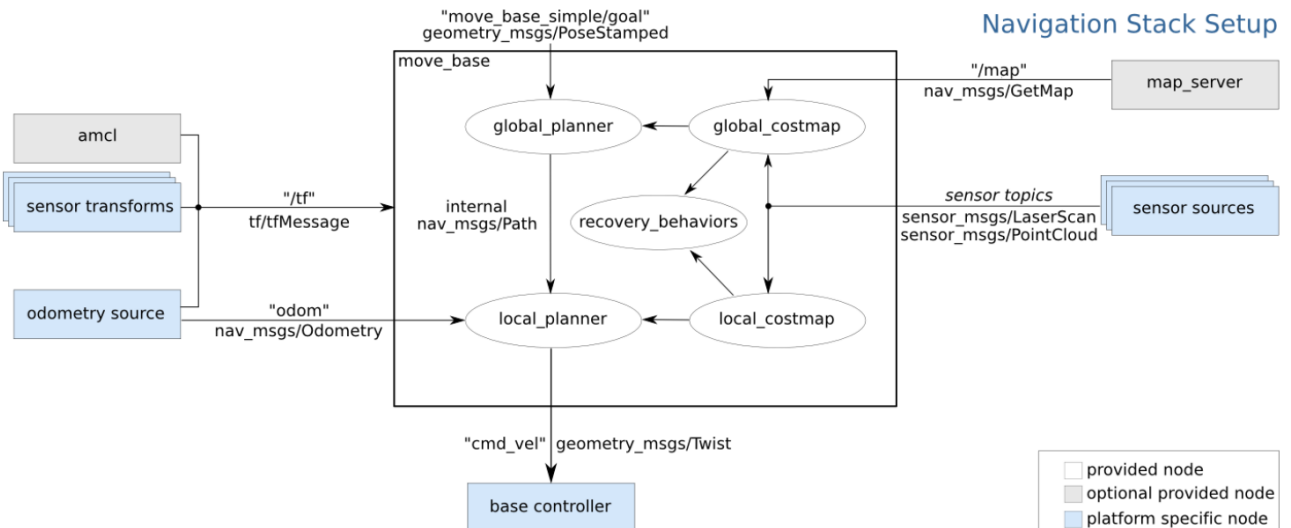


Рисунок 4.1 – Загальний вигляд стеку навігації в ROS [65]

Крім апаратних вимог, СН складається із цілого комплексу взаємопов'язаних програмних пакетів ROS. Рисунок 4.1 відображає стандартизовану схему СН для будь-якого типу МР. Кожен елемент (крім `amcl` та `map_server`) є обов'язковим і незамінним у стеку навігації - за відсутності будь-якої його частини СН не буде функціонувати. Компоненти в діаграмі, оформлені в білому кольорі, є реалізованими та готовими до використання, у сірому кольорі - реалізовані та опціональні у використанні, у блакитному кольорі компоненти мають бути розроблені для кожної платформи МР.

Модуль `move_base` є основною частиною СН МР, який отримує необхідні дані з інших компонентів системи та використовує їх для аналізу та ухвалення рішень у процесі навігації МР. `Move_base` координує отримані дані і посилає відповідні команди на контролер бази (через `cmd_vel`), приводячи в рух МР.

Запуск і конфігурація `move_base` для моделі МР «Junior» описані в Додатку Д.

Даний модуль містить у собі наступні підмодулі:

- `Local_planner` - локальний планувальник;
- `Global_planner` - глобальний планувальник;
- `Global_costmap` - глобальна карта. Карта, яка зберігає інформацію про перешкоди, які існують у світі МР. Глобальна карта використовується для створення довгострокових планів по всьому навколишньому простору МР;

- `Local_costmap` - локальна карта. Карта, яка зберігає інформацію про перешкоди у світі МР. Ця карта використовується для локального планування та обходу існуючих перешкод;

- `Recovery_behaviors` - поведінка відновлення. Даний підмодуль поєднує у собі кілька заходів, що проводяться в разі втрати плану (шляху) у `move_base`. Зазвичай, даний підмодуль приводиться в дію, коли МР сприймає себе застряглим на карті.

Усі модулі `move_base` проходять налаштування у спеціальних конфігураційних файлах формату «*.yaml», а `move_base` запускається за допомогою використання `launch`-файлу.

Необхідні компоненти системи навігації.

Конфігурація перетворень. СН вимагає, щоб МР публікував інформацію про відносини між координатними системами всередині МР, використовуючи `tf` `ros`-пакет.

`Transformation` (або `Tf`) - це пакет, який дозволяє користувачеві стежити кілька координатних систем з плином часу. `Tf` відображає відносини між координатними системами у вигляді деревоподібної структури (буферизованої за часом), що дає змогу користувачеві обчислювати координати вектора в будь-якій координатній системі для будь-якого моменту часу.

Інформація про одометрію. СН вимагає публікації інформації про одометрію МР з використанням `tf` та спеціальних повідомлень `nav_msgs/Odometry`.

Дані з датчиків. СН використовує інформацію з бортових датчиків МР для уникнення зіткнень із перешкодами; СН передбачає, що бортові датчики публікують або `sensor_msgs/LaserScan` повідомлення (повідомлення з лазерного далекоміра), або `sensor_msgs/PointCloud` (повідомлення з камери глибини) в ROS.

Base controller (або контролер бази) - СН припускає, що він може надсилати команди швидкості за допомогою спеціальних повідомлень `geometry_msgs/Twist`; зі свого боку, ці повідомлення мають перебувати в межах теми `cmd_vel`. Це означає, що має існувати нода, яка зчитує тему `cmd_vel`, приймає команди типу `vx`, `vy`, `vtheta` (`cmd_vel.linear.x` - лінійна швидкість в осі `x`, `cmd_vel.linear.y` - лінійна швидкість в осі `y`, `cmd_vel.angular.z` - кутова швидкість в осі `z`) і перетворює на команди на мотори мобільної бази МР.

Далі більш детально будемо описувати етапи створення та налаштування компонентів для створення СН.

Наступним етапом роботи стала спільна організація наявних нод керування рухом моделі МР з темою `cmd_vel`, з якою взаємодіє модуль навігації `move_base`. На даному етапі було взято готову реалізацію `cmd_vel` з офіційного сайту ROS [65] для МР, що працюють із геометрією Акерманна [60]. Для того, щоб зв'язати два окремі модулі, потрібно було створити єдину тему спеціальних повідомлень `/junior_car/ackermann_cmd`, яка об'єднувала дані `steer_remap` і `drive_remap` в єдине повідомлення типу `ackermann_msgs/AckermannDriveStamped`.

4.2 Коригування отриманих даних з лазера моделі МР

Для коректної роботи СН однією з важливих вимог апаратного забезпечення МР є правильно встановлений лазер (лазер, розташований на поверхні строго паралельній площині землі). У випадку МР «Junior», лазерний далекомір на ньому розташовується під кутом приблизно в 15° . Оскільки модель

повторює фізичні характеристики реального МР, створений лазер у Gazebo також має дану величину куту нахилу.

Проте це суперечить вимогам справної роботи СН. Для усунення даної суперечності було ухвалено рішення створити спеціальну `node rectified_scan`, яка б зчитувала дані, що надходять з лазера моделі, та коригувала їх у дані, які відповідають правильно встановленому лазеру (компенсуючи наявний кут нахилу).

`Node rectified_scan` зчитує тему `/scan`, у якій публікуються повідомлення, що надсилаються лазером моделі МР. При отриманні нового повідомлення від лазера, `node` створює нове власне повідомлення типу `sensor_msgs/LaserScan`, коригуючи наступні поля отриманого повідомлення: `std_msgs/Header header`, `float32[] intensities`, `float32[] ranges`. Своєю чергою, у строці `Header` замінюється поле `frame_id`, вказуючи на нову вісь віртуального лазера, розміщеного згідно із встановленими вимогами СН; у полях `float32[] intensities`, `float32[] ranges` відбувається коригування вхідних значень від отриманих повідомлень теми `/scan`.

Нове повідомлення з даними публікується в іншій темі, яка далі використовує метод одночасної локалізації положення та картографування `hector_mapping`. Вихідний код пакета коригування даних із лазера наведено в Додатку Е.

4.3 Налаштування локального планувальника

Налаштування локального планувальника зберігається у файлі `teb_local_planner.yaml` (див. Додатки). Налаштування являє собою перерахування параметрів локального планувальника та величини у разі конкретного МР. Кожен із параметрів володіє величинами за замовчуванням, які можуть підходити для всіх МР, однак рекомендується уточнювати величини усіх параметри, спираючись на властивості конкретного МР.

У разі неголономних МР, що мають рульове керування (передні колеса МР можуть повертатися і задавати напрямок руху МР), у ROS існує невелика кількість планувальників (локальних чи глобальних), які б враховували специфічну конфігурацію МР. У загальному випадку в стеку навігації МР є голономним (або всеспрямованим). Прикладом доступного (відкритого) локального планувальника для МР з геометрією Аккермана [60] є середовище `Teb Local Planner` [66]. Пакет `teb_local_planner` реалізує плагін для `base_local_planner`, або для 2D системи навігації. В основі планувальника лежить метод `Timed Elastic Band` [67, 68], він локально оптимізує траєкторію руху МР з урахуванням часу виконання траєкторії, уникнення перешкод, дотримуючись заданої максимальної величини швидкості та прискорення. Даний підхід ґрунтується на іншому наявному методі, `Elastic Band`, що має свою реалізацію у вигляді локального планувальника в середовищі ROS (`eband_local_planner`) [42].

`Teb_local_planner` [67, 68] публікує спеціальні повідомлення у низку тем: `local_plan`, `global_plan`, `teb_markers`, `teb_poses`, `teb_feedback`.

- `local_plan` (тип спеціального повідомлення `nav_msgs/Path`) - локальний план або траєкторія руху, яку оптимізує `teb_local_planner` і слідує їй. Використовується здебільшого для візуалізації;

- `global_plan` (тип спеціального повідомлення `nav_msgs/Path`) - глобальний план, якому `local_plan` намагається слідувати. Використовується в основному для візуалізації;

- `teb_markers` (тип спеціального повідомлення `visualization_msgs/Marker`) - `teb_local_planner` надає додаткову інформацію про сцену планування через маркери з різним простором імен. Простори імен `PolyObstacles` чи `PointObstacles` візуалізують усі точкові та полігональні перешкоди, які наразі враховуються під час оптимізації. Простір імен `TebContainer` візуалізує всі знайдені й оптимізовані траєкторії руху, що спираються на альтернативні топології (тільки у випадку вмикання паралельного планування);

- `teb_poses` (тип спеціального повідомлення `geometry_msgs/PoseArray`) -

список дискретних поз (SE2 простору) поточного `local_plan`. Використовується в основному для візуалізації;

- `teb_feedback` (тип спеціального повідомлення `teb_local_planner/FeedbackMsg`) - повідомлення 33 містить заплановану траєкторію руху, включно з профілем швидкості, часову інформацію, список існуючих перешкод. Використовується для налаштування, для роботи даної теми необхідно ввімкнути параметр «`publish_feedback`».

Своєю чергою, `local_plan` підписаний на наступну низку тем: `odom` - (тип спеціального повідомлення `nav_msgs/Odometry`) - інформація про одометрію, яка дає локальному планувальнику поточну величину швидкості МР; `via_points` - (тип спеціального повідомлення `nav_msgs/Path`) - надає проміжні точки користувача; `obstacles` - (тип спеціального повідомлення `costmap_converter/ObstacleArrayMsg`) - надання перешкод користувача у вигляді точкових, лінійних або багатокутних перешкод (на додаток до перешкод карти витрат або замість них).

Параметри `teb_local_planner` дають змогу користувачеві виконати налаштування поведінки планувальника і, як наслідок, всього МР. Параметри згруповані в кілька категорій: конфігурація МР, точність кінцевої позиції (або `goal tolerance`), конфігурація траєкторії руху, перешкод, оптимізація, планування в особливих топологіях та інші параметри.

У файлі налаштування `*yaml` міститься список даних параметрів із певним величинами (заданим користувачем). У разі, якщо параметр не задано у даному файлі, планувальник використовує значення за замовчуванням для будь-якого параметра.

Параметри конфігурації МР наведено в Табл. 4.1 та Додатках. У першому стовпчику перераховано всі параметри, у другому наведено коротке пояснення, у третьому – величини за замовчуванням, у четвертому - визначене значення для моделі МР «Junior».

Таблиця 4.1 – Технічні параметри конфігурації МР [42]

| Назва | Пояснення | Значення за замовчуванням | Задане значення |
|------------------------------|--|---------------------------|-----------------|
| <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> |
| acc_lim_x | max поступальне прискорення МР в м/с ² | 0,5 | 10 |
| max_vel_x | max поступальна швидкість МР в м/с | 0,4 | 10 |
| acc_lim_theta | max кутове прискорення МР в рад/с ² | 0,5 | 10 |
| max_vel_x_backwadrs | max абсолютна поступальна швидкість МР під час руху назад у м/с | 0,2 | 4 |
| min_turning_radius | min радіус повороту car-like МР | 0,0 | 1,0 |
| max_vel_theta | max кутова швидкість МР в рад/с | 0,3 | 10 |
| max_vel_y | max швидкість руху МР (повинно бути 0 для неголономних МР) | 0,0 | 0,0 |
| Wheelbase | дистанція між передньою та задньою осями коліс МР | 1,0 | 1,0 |
| acc_lim_y | max прискорення МР | 0,5 | 10 |
| footprint_model/type | тип моделі проєкції МР на площину, що використовується для оптимізації. регламентовані типи: point, line, circular, two_circles, polygon | point | polygon |
| footprint_model/line_start | даний параметр актуальний тільки для типу моделі проєкції «line». Містить початкові координати відрізка. | [-0.3,0.0] | - |
| footprint_model/line_end | даний параметр актуальний тільки для типу моделі проєкції «line». Він містить кінцеві координати відрізка. | [0.3,0.0] | - |
| footprint_model/front_radius | даний параметр актуальний тільки для типу моделі проєкції «two_circles». Містить радіус переднього кола. | 0,2 | - |
| footprint_model/front_offset | даний параметр актуальний тільки для типу моделі проєкції «two_circles». Він описує, наскільки центр переднього кола зміщений уздовж осі X МР. Передбачається, що вісь обертання МР знаходиться в точці [0,0]. | 0,2 | - |

Продовження таблиці 4.1

| 1 | 2 | 3 | 4 |
|-----------------------------|---|---------------------------|--|
| footprint_model/rear_radius | даний параметр актуальний тільки для типу моделі проєкції «two_circles». Містить радіус заднього кола. | 0,2 | - |
| footprint_model/rear_offset | даний параметр актуальний тільки для типу моделі проєкції «two_circles». Він описує, наскільки центр заднього кола зміщений уздовж негативної осі X MP. Передбачається, що вісь обертання MP знаходиться в точці [0,0]. | 0,2 | - |
| footprint_model/vertices | даний параметр актуальний тільки для типу моделі проєкції «polygon». Він містить список вершин багатокутника (тільки 2D координати, крім того багатокутник завжди замкнутий). | [[0.25,-0.05],[...], ...] | [[[-0.556, -0.325], [-0.556,0.325], [0.556,0.325], [0.556, -0.325]]] |

Точність кінцевої позиції визначає величини числових значень допущення відхилення від необхідної кінцевої позиції MP. Допущення виражаються через два основні значення: допустима помилка орієнтації MP (у радіанах) та допустима евклідова відстань від цільової точки (у метрах). Параметри даної категорії представлено в табличному вигляді.

Таблиця 4.2 – Технічні параметри точності кінцевої позиції MP

| Назва | Пояснення | Значення за замовчуванням | Задане значення |
|--------------------|--|---------------------------|-----------------|
| xu_goal_tolerance | допустима кінцева евклідова відстань до встановленої цілі в метрах | 0,2 | 0,2 |
| free_goal_vel | обмеження швидкості цілі, щоб MP міг досягти встановленої цілі із максимальною швидкістю | false | false |
| yaw_goal_tolerance | допустима кінцева помилка орієнтації положення в радіанах | 0,2 | 0,3 |

Налаштування траєкторії руху МР, дані параметри категорії представлено в табличному вигляді.

Таблиця 4.3 – Технічні параметри конфігурації траєкторії руху МР [67, 68]

| Назва | Примітка | Значення за замовчуванням | Задане значення |
|-----------------------------------|--|---------------------------|-----------------|
| <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> |
| global_plan_overwrite_orientation | перезапис орієнтації локальних підцілей, що надаються глобальним планувальником | true | true |
| dt_ref | необхідний часовий дозвіл траєкторії (траєкторія руху буде змінюватися між ітераціями, якщо $dt_ref \pm dt_hysteresis$ порушується) | 0,3 | 0,6 |
| min_samples | мін кількість зразків (має бути більше ніж 2) | 3 | 3 |
| dt_hysteresis | гістерезис для автоматичної зміни розміру траєкторії в залежності від поточного значення тимчасової роздільної здатності | 0,1 | 0,1 |
| global_plan_viaoint_sep | якщо величина параметра позитивна, проміжні точки видаляються із глобального плану | Мінус 0,1 Вимкнено | Мінус 0,1 |
| max_global_plan_lookahead_dist | зазначення максимальної довжини (сукупної евклідової відстані) підмножини глобального плану, що враховується для оптимізації | 3 | 3 |
| feasibility_check_no_poses | вказівка, до якої позиції в прогнозованому плані слід перевіряти здійсненність кожного інтервалу вибірки | 4 | 0 |
| force_reinit_new_goal_dist | повторна ініціалізація траєкторії руху, якщо попередня ціль оновлена з поділом більше зазначеного значення в метрах | 1 | 1 |
| publish_feedback | публікація зворотної реакції планувальника, що містить повну траєкторію руху і список активних перешкод (має бути ввімкнений тільки для оцінки або налагодження) | false | false |

Продовження таблиці 4.3

| 1 | 2 | 3 | 4 |
|----------------------------------|---|-------|-------|
| allow_init_with_backwards_motion | якщо значення істинне, базові траєкторії руху можуть бути ініціалізовані з рухами назад, якщо ціль знаходиться за початковою точкою у локальній карті (рекомендується, тільки якщо МР оснащений задніми давачами) | false | false |
| shrink_horizon_backup | дозволяє планувальнику тимчасово зменшити горизонт (приблизно 50%) у разі автоматично виявлених проблем чи перешкод (наприклад, нездійсненність рішення) | true | true |
| shrink_horizon_min_duration | зазначення мінімальної тривалості для зменшеного горизонту в разі виявлення нездійсненої траєкторії руху | 10 | 10 |
| exact_arc_length | якщо значення Істина, планувальник використовує точну довжину дуги в розрахунках величини швидкості, величини прискорення та величини швидкості повороту, в іншому випадку використовується евклідове наближення | false | false |

Налаштування перешкод описує поведінку планувальника щодо перешкод розташованих на карті. Крім стандартного параметра мінімальної відстані до перешкоди, `teb_local_planner` враховує спеціальні параметри, які також представлено в табличному вигляді.

Параметри оптимізації є розширенням стандартного підходу Elastic Band, описані в відомих роботах [69, 70]. Параметри оптимізації представлено в табличному вигляді.

Таблиця 4.4 – Технічні параметри налаштування перешкод [66, 67]

| Назва | Примітка | Значення за замовчуванням | Задане значення |
|-------------------------------------|--|---------------------------|-----------------|
| <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> |
| include_costmap_obstacles | значення обліку перешкоди в локальній карті. Кожна комірка, позначена як перешкода, розглядається як точка-перешкода | true | true |
| min_obstacle_dist | мін бажана відстань до перешкод | 0,5 | 0,7 |
| costmap_obstacles_behind_robot_dist | обмеження зайнятих локальних перешкод карти, прийняті до уваги для планування позаду МР (зазначення відстані в метрах) | 1,0 | 1,5 |
| inflation_dist | зона буфера навколо існуючих перешкод із ненульовою штрафною вартістю (повинна бути більшою, ніж min_obstacle_dist для набуття параметром чинності) | 0,6 | 0,6 |
| obstacle_poses_affected | кожна позиція існуючої перешкоди «прикріплюється» до найближчої позиції на траєкторії руху, щоб зберегти дистанцію. Додаткові сусіди також можуть бути прийняті до уваги | 30 | 30 |
| include_dynamic_obstacles | якщо для даного параметра встановлено значення true, рух перешкод із ненульовою швидкістю прогнозується і враховується під час оптимізації за допомогою моделі з постійною величиною швидкості | false | false |
| obstacle_association_cutoff_factor | даний параметр використовується тільки в тому випадку, якщо для параметра legacy_obstacle_association встановлено значення false | 5 | 5 |

Продовження таблиці 4.4

| 1 | 2 | 3 | 4 |
|---|--|-------|-------|
| legacy_obstacle_association | стратегія з'єднання позиції траєкторій руху із існуючими перешкодами для оптимізації було змінено. Можливий вибір старої або попередньої стратегії, встановивши для даного параметра значення true | false | false |
| obstacle_association_force_inclusion_factor | стратегія асоціації не наступних існуючих перешкод використовується під час оптимізації для з'єднання до дискредитованої траєкторії тільки значущих існуючих перешкод | 1,5 | 1,5 |

Таблиця 4.5 – Технічні параметри оптимізації [69, 70]

| Назва | Примітка | Значення за замовчуванням | Задане значення |
|----------------------|---|---------------------------|-----------------|
| 1 | 2 | 3 | 4 |
| no_outer_iterations | кожна зовнішня ітерація автоматично змінює розмір траєкторії руху відповідно до необхідної тимчасової роздільної здатності dt_ref і викликає внутрішній оптимізатор | 4 | 4 |
| no_inner_iterations | кількість вирішальних фактичних ітерацій, викликаних у кожній зовнішній ітерації | 5 | 5 |
| penalty_epsilon | додати запас до штрафних функцій для наближень із жорсткими обмеженнями | 0,1 | 0,1 |
| weight_max_vel_theta | вага оптимізації для задоволення max допустимої кутової швидкості | 1,0 | 20,0 |
| weight_max_vel_x | вага оптимізації для задоволення max допустимої поступальної швидкості | 2,0 | 20,0 |

Продовження таблиці 4.5

| 1 | 2 | 3 | 4 |
|----------------------------------|--|--------|--------|
| weight_kinematics_nh | вага оптимізації для задоволення неголономної кінематики МР | 1000,0 | 1000,0 |
| weight_kinematics_turning_radius | вага оптимізації для забезпечення мінімальної величини радіуса повороту (тільки для car-like МР) | 1,0 | 5,0 |
| weight_kinematics_forward_drive | вага оптимізації для того, щоб змусити МР вибирати тільки прямі напрямки (позитивна поступальна швидкість). Низька вага (1,0), як і раніше, надає можливість рухатися МР назад | 1,0 | 53,0 |
| weight_optimaltime | вага оптимізації для скорочення траєкторії руху з переходом чи часом виконання | 1,0 | 1,0 |

Конфігурація інших параметрів містить у собі опис теми одометрії МР і найменування глобальної системи координат. Параметри наведено в табличному вигляді.

Таблиця 4.6 – Інші технічні параметри планувальника

| Назва | Примітка | Значення за замовчуванням | Задане значення |
|------------|---|---------------------------|-----------------|
| odom_topic | назва теми одометрії | odom | odom |
| map_frame | система координат глобального планування (у випадку статичної карти даний параметр зазвичай має бути змінений на значення «/map») | odom | /map |

Налаштування глобального планувальника описано у файлі із назвою

avrora_unior_nav.yaml. В якості глобального планувальника було обрано SBPLLatticePlanner. Пакет sbpl_lattice_planner є спеціальною оболонкою SBPL Lattice оточення, що відповідає інтерфейсу nav_core::BaseGlobalPlanner, зазначеному в nav_core [71].

Завдяки цьому планувальник lattice_planner може бути використаний як глобальний планувальник у move_base. Глобальний планувальник генерує шлях від поточної позиції МР до вказаної цілі. Шлях генерується шляхом об'єднання серій «примітивів руху» (або motion primitives), які являють собою короткі, кінематичні можливі рухи МР. Таким чином, планування виконується у трьох вимірах x , y та θ , що призводить до плавних траєкторій руху, які враховують орієнтацію положення МР. Це є ключовим фактором для МР не круглої форми або МР, що мають неголономні обмеження.

У випадку з МР «Junior», МР має неголономні обмеження і не сумісний із більшістю планувальників у середовищі ROS.

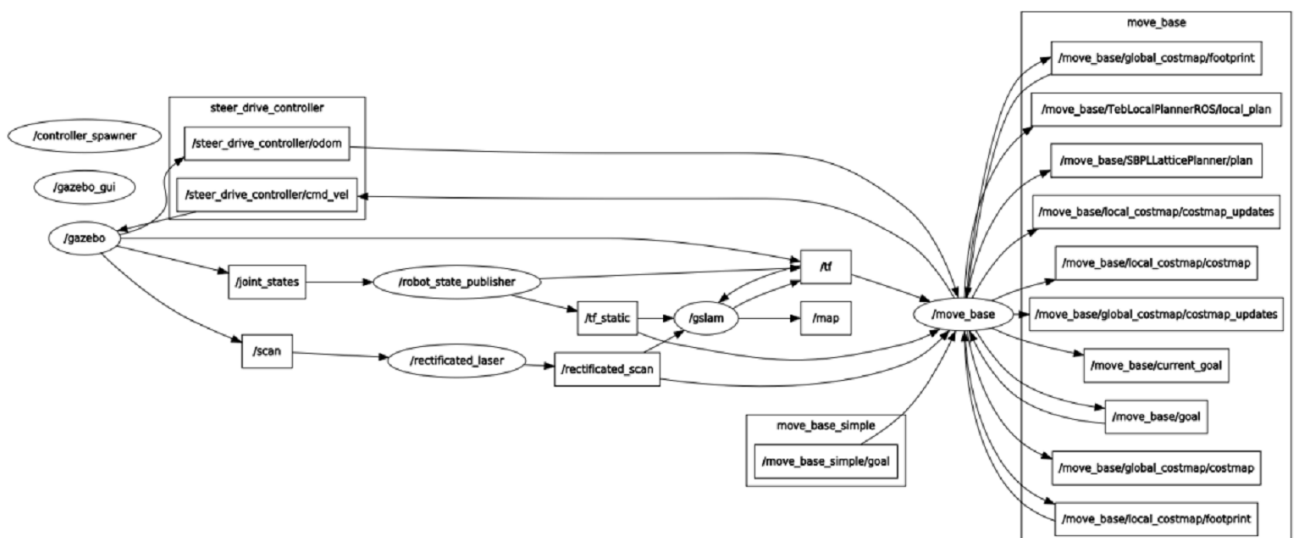


Рисунок 4.2 – Загальний вигляд роботи стека навігації в rqt_graph

4.4 Налаштування карт вартості планувальника

Система навігації використовує дві, так звані, карти вартості (або інакше costmap) для збереження інформації про існуючі перешкоди в навколишньому

середовищі MR. Одна `costmap` використовується для глобального планування, для створення довгострокових планів по всьому навколишньому середовищу; інша `costmap` використовується для локального планування та уникнення існуючих перешкод. Кожна `costmap` володіє низкою параметрів, які необхідно задати для конкретного MR і системи навігації. Прийнято розділяти параметри на три основні типи: загальні параметри конфігурації, параметри глобальної та локальної конфігурації [65-71].

`Costmap_2d` [72] є вбудованим пакетом у середовищі ROS, який надає спеціальну структуру (яка містить інформацію про те, куди може переміщатися MR) у вигляді сітки зайнятості (або `occupancy grid`). `Costmap` використовує дані давачів та інформацію зі статичної карти для збереження та оновлення інформації про перешкоди у навколишньому середовищі за допомогою спеціального об'єкту `costmap_2d::Costmap2DRos`.

`Costmap` автоматично підписується на теми давачів через ROS і оновлюється відповідним чином. Кожен давач використовується для маркування (додавання інформації про існуючі перешкоди в `costmap`), видалення (видалення інформації про неіснуючі перешкоди з `costmap`) або для обох цілей.

Операція маркування визначена для внесення змін у вартості комірки у вигляді індексу масиву `costmap`. Операція очищення виконує перерахунок `costmap`, використовуючи трасування набору променів, побудованих з точки, де знаходиться давач, через кожен точку, де були раніше зафіксовані існуючі перешкоди. Якщо для опису існуючих перешкод використовуються тривимірні структури, інформація про існуючу перешкоду для кожного з вертикальних стовпчиків проєктується вниз для отримання двовимірної `costmap`. Хоча кожна комірка в `costmap` володіє певним значенням в інтервалі від 0 до 255 різних значень вартостей, базова структура (або ж сітка зайнятості), що використовується, має лише три вартості. Кожна комірка в даній структурі може бути зайнятою, вільною або ж невідомою. Кожному статусу присвоюється спеціальне значення під час виконання проєкції на `costmap`.

Costmap виконує цикли оновлення карти з частотою, вказаною в параметрі `update_frequency`. У кожному циклі надходять необхідні дані із давачів, операції маркування і видалення виконуються в базовій структурі (тобто в сітці зайнятості), яка потім проєктується на `costmap` з відповідними значеннями.

Інфляція - процес поширення значень вартості із зайнятих комірок, що зменшуються зі збільшенням відстані від існуючої перешкоди (рис. 4.3). Інфляція необхідна для збільшення відстаней від МР до існуючих перешкод, тим самим забезпечуючи безпеку МР під час навігації по `costmap`.

У даному процесі виділяють п'ять конкретних вартостей для значень `costmap` [72]:

- `inscribed cost` або «вписана» вартість - вартість комірки, якщо вона знаходиться від реальної перешкоди на відстані меншій, ніж радіус кола, вписаного в слід (footprint) МР;

- `lethal cost` або «летальна» вартість - в даній комірці існує реальна перешкода. Тому, якби центр МР перебував у даній комірці, МР би не уникнув зіткнення;

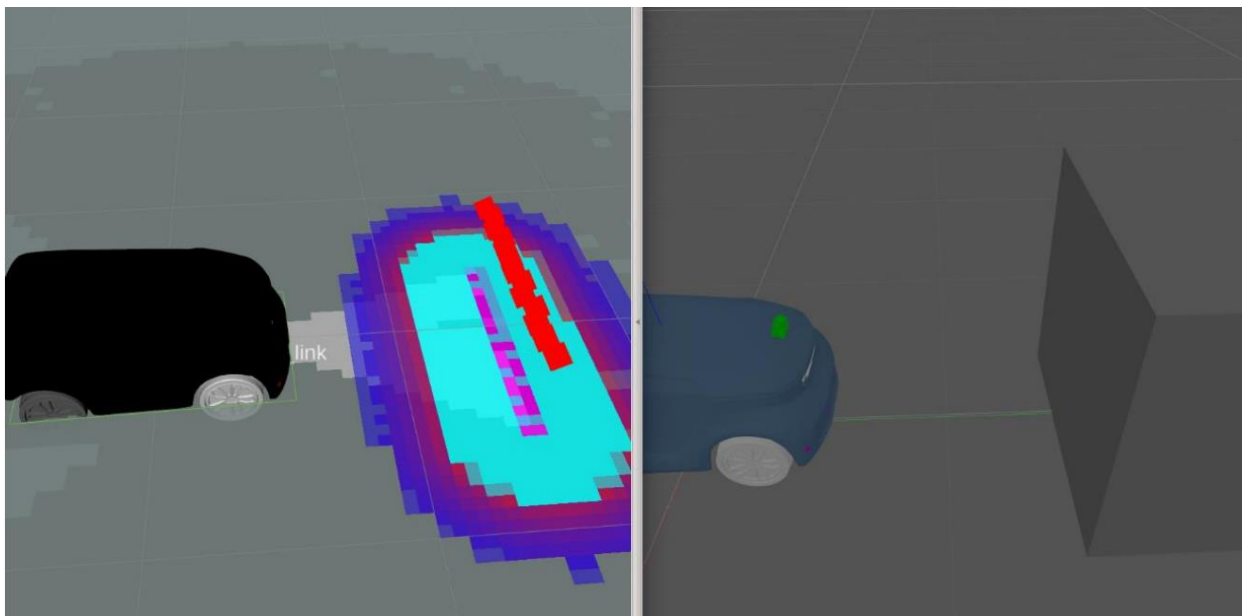


Рисунок 4.3 – Загальний вигляд відображення локальної `costmap` на прикладі реальної перешкоди «куб» [72]

- freespace cost або «Вільна» вартість - передбачає, що вартість даної комірки буде рівною нулю; тобто немає нічого, що перешкоджає МР потрапити в дану комірку;

- possibly circumscribed cost або «Можлива вписана» вартість - вартість, визначена за аналогією з inscribed cost, із заміною вписаного кола на описане. Така вартість говорить про те, що якщо центр МР знаходиться в комірці з рівним або більшим значенням, то небезпека зіткнення з нею буде залежати від орієнтації МР. У даному випадку використовується термін «можливо-», оскільки така вартість визначає безпосередньо перешкоду, конкретні значення встановлюються користувачем на власний розсуд. Наприклад, користувач може обумовити уникнення МР певних частин будівлі шляхом завдання значень для таких областей на costmap;

- unknown cost або «Невідома» вартість - означає відсутність інформації про дану комірку costmap.

Усім іншим можливим вартостям присвоюється значення між «freespace cost» та «possibly circumscribed cost», в залежності від відстані до «lethal cost» комірки.

Система навігації використовує costmap для зберігання інформації про існуючі перешкоди у навколишньому середовищі. Для того, щоб інформація була актуальною та правильною, необхідно вказати теми («шини» для передавання певного типу даних до середовища ROS) для costmap, повідомлення аналіз яких вони мають виконати для оновлення карт. Загальна конфігурація costmap описана у файлі common_costmap.yaml (зміст common_costmap.yaml наведено в Додатках). Далі представлено опис деяких параметрів та значення в табличному вигляді.

Налаштування глобальної costmap наведено у файлі global_costmap.yaml який містить параметри, специфічні для глобальної costmap. Отримані визначення наведено в табличному вигляді та в Додатках.

Таблиця 4.7 - Загальна конфігурація costmap [72]

| Назва параметру | Значення | Примітка |
|---------------------|---|---|
| footprint | [[[-0.556, -0.325],[-0.556, 0.325],[0.556, 0.325],[0.556, -0.325]] | Проекція МР на площину |
| transform_tolerance | 0,1 | Визначення допустимої затримки в перетворенні (tf) даних. Даний параметр використовується як гарантія втрати посилання в дереві tf, водночас залишаючи певний час очікування, достатній для користувача |
| footprint_padding | 0,0 | Додаткова відстань між проекцією МР на площину і реальною перешкодою |
| robot_base_frame | footprint | Визначення системи координат бази МР, на яку повинна орієнтуватися costmap |
| update_frequency | 5,0 | Визначення частоти, з якою costmap буде запускати власний цикл оновлення |
| global_frame | map | Визначення, в якій координатній системі повинна працювати costmap |
| publish_frequency | 1,0 | Визначення частоти, з якою costmap буде публікувати інформацію візуалізації |
| resolution | 0,05 | Роздільна здатність (метр/комірку) costmap |
| rolling_window | true | Встановлення параметра в значенні Істина означає, що costmap буде залишатиметься центрованою навколо МР, поки МР рухається навколишнім середовищем |
| obstacles_laser | observation_sources: laser laser data_type: LaserScan clearing: true marking: true topic: rectificated_scan inf_is_valid: true | Структура для опису давачів МР, які необхідні для побудови та оновлення costmap. Параметр «selection_sources» визначає список давачів розділених пробілами, які будуть передавати інформацію в costmap |
| inflation_layer | inflation_radius: 0.7 enabled: true | Опис структури інфляції. Радіус інфляції повинен бути встановлений на максимальній відстані від реальних перешкод, за яких мають бути понесені витрати |

Таблиця 4.8 – Налаштування глобальної costmap [72]

| Назва параметру | Значення | Примітка |
|-------------------|----------|---|
| update_frequency | 5,0 | Визначення частоти, з якою costmap буде запускати власний цикл оновлення |
| resolution | 0,1 | Роздільна здатність (метр/комірку) costmap |
| publish_frequency | 5,0 | Визначення частоти, з якою costmap буде публікувати інформацію візуалізації |

Налаштування локальної costmap, файл local_costmap.yaml володіє параметрами, специфічними для локальної costmap. Значення представлено в табличному вигляді та в додатках.

Таблиця 4.9 – Налаштування локальної costmap [72]

| Назва параметру | Значення | Примітка |
|-----------------|----------|--|
| height | 6,0 | Висота costmap в метрах |
| width | 6,0 | Ширина costmap в метрах |
| static_map | false | Параметр «static_map» визначає, чи повинна ініціалізуватися costmap на основі карти, яку обслуговує map_server |
| resolution | 0,05 | Роздільна здатність (метр/комірка) costmap |

4.5 Апробація системи навігації в Rviz

Для апробації системи навігації МР в Rviz [43, 44] було запропоновано наступну процедуру.

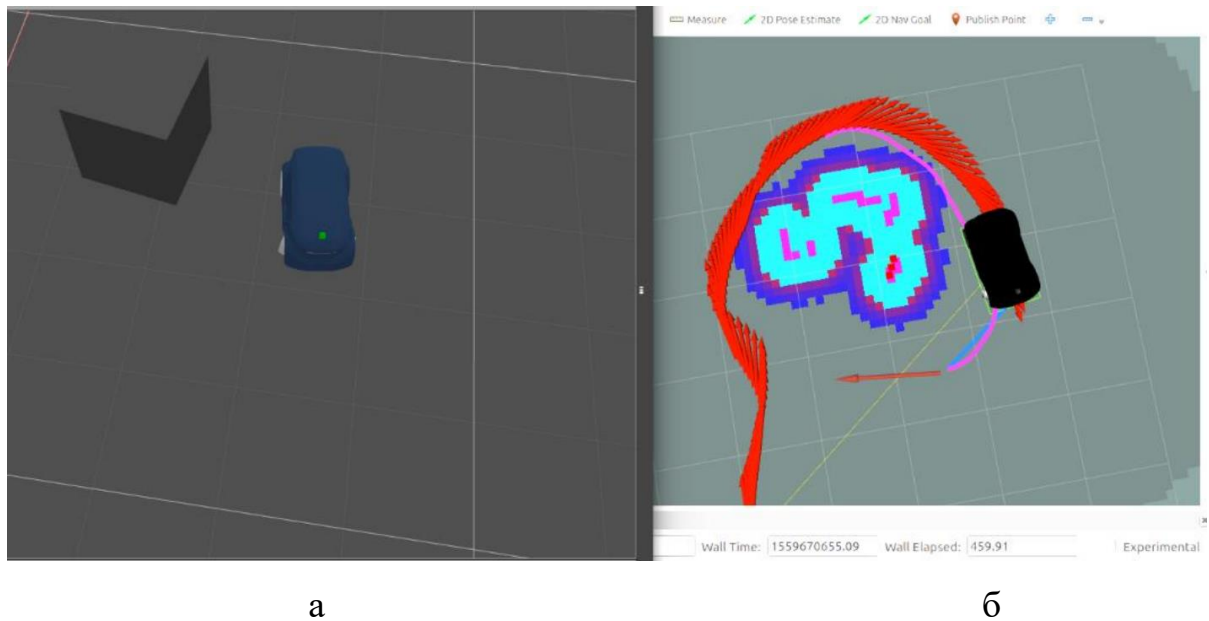
1. Перевірка навігації МР в порожньому навколишньому середовищі. Вибір кінцевої точки визначається таким чином, щоб траєкторія шляху прокладалася строго прямо, з мінімальною зміною орієнтації МР в кінцевій (цільовій) точці (рис. 4.4).

2. Перевірка навігації МР в порожньому навколишньому середовищі. Вибір мети визначається таким чином, щоб кінцева точка була в орієнтації, відмінній від початкової орієнтації МР.

3. Перевірка навігації МР у навколишньому середовищі із існуючими перешкодами. Мета обирається таким чином, щоб траєкторія шляху до неї вибудовувалася з урахуванням обходу реальних перешкоди.

Дослідження виконувались згідно із вказаними процедурами. Для апробації потрібні були наступні кроки:

- запуск моделі МР в симуляторі Gazebo;
- запуск контролерів моделі;
- запуск СН;



а - візуалізація симуляції; б - дані МР через Rviz

Рисунок 4.4 – Загальний вигляд навігації МР у навколишньому середовищі із існуючими перешкодами [43, 44]:

- запуск Rviz із файлом конфігурації, представленим у Додатках;
- налаштування процедур виконували під час відстеження повідомлень, надісланих на теми контролерів, `move_base` (рис. 4.5).

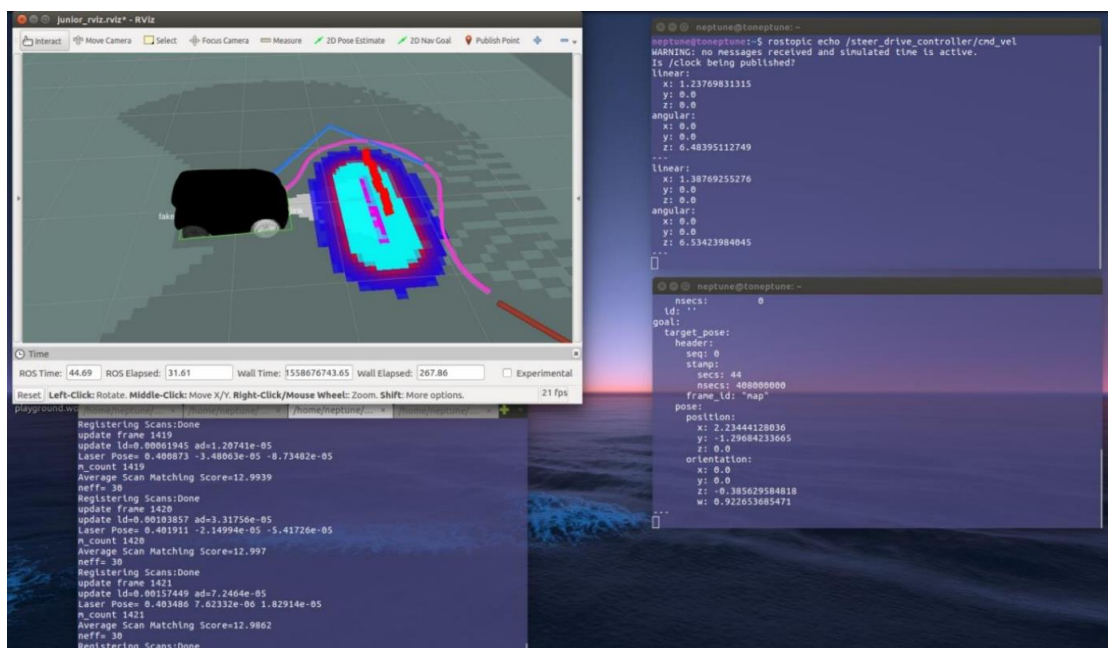


Рисунок 4.5 – Загальний вигляд налаштування СН

Концепція вирішення поставленої проблеми складається із 2 етапів.

Перший етап - підготовка до апробації, яка включає в себе: моделювання навколишнього середовища в симуляторі Gazebo [43], що відповідає міській забудові; створення node, яка буде відправляти в СН МР кінцеві точки.

Другий етап – апробація, процедура наступна - у запущеному віртуальному навколишньому середовищі МР виконуватиме автономну навігацію в кінцеві точки; після досягнення МР встановленої кінцевої точки, node відправляти на СН наступну кінцеву точку.

Для виконання апробації автономної навігації моделі в кінцеві точки необхідно створити відповідний світ у середовищі Gazebo [43], топологія якого була б схожою на міську забудову. Для створення навколишнього середовища використовувалися стандартні 3D моделі, доступні в базі моделей симулятора Gazebo [43], на рис. 4.6-4.7 представлено отриману модель світу.

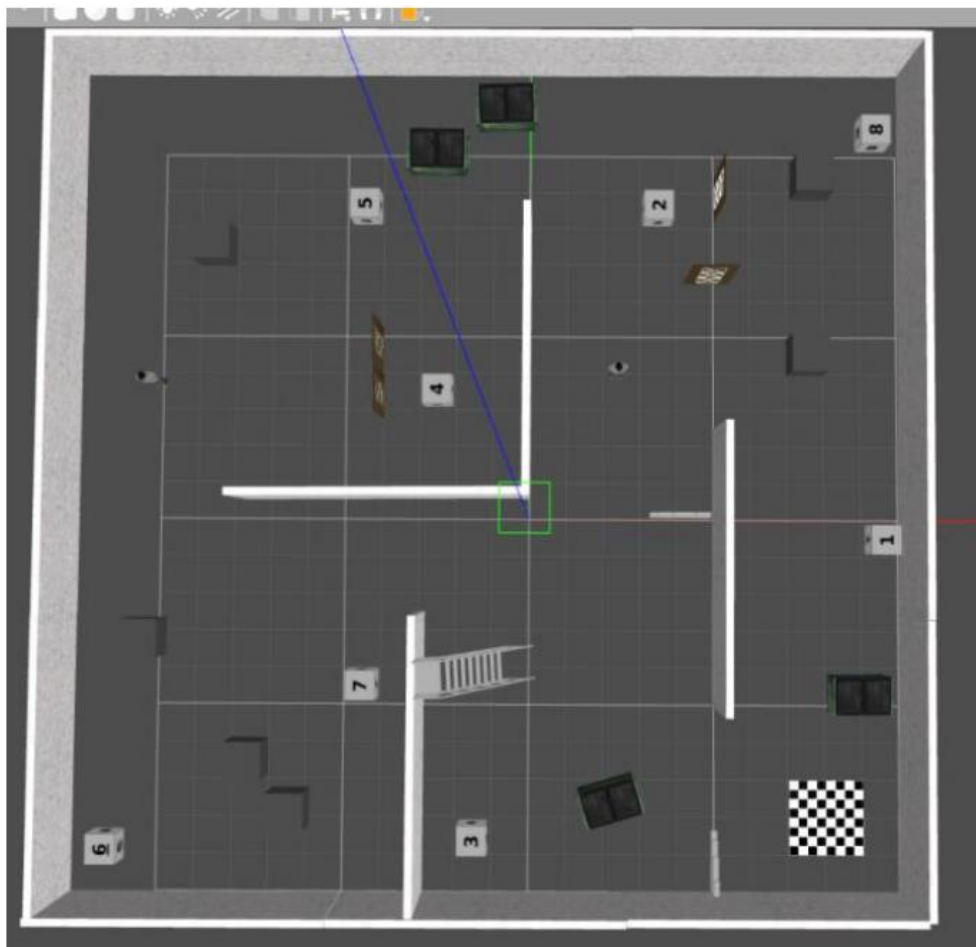


Рисунок 4.6 – Загальний вигляд створеної моделі світу в 2D

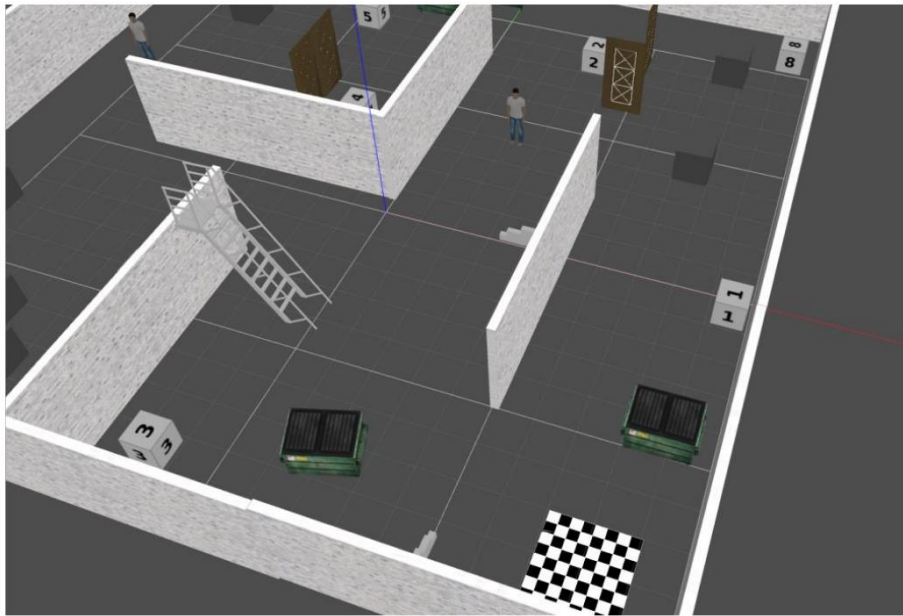


Рисунок 4.7 - Загальний вигляд створеної моделі світу в 3D

У змодельованому світі знаходяться спеціальні об'єкти (куби), пронумеровані від 1 до 9, які є кінцевими точками (див. рис. 4.8). Розмір моделі світу 23x23 метри. У середині змодельованого світу додано панелі для імітації міської забудови. Ширина прорізів буде рівною 3 м, що відповідає ширині дороги із двома дорожніми смугами. У змодельований світ також довільним чином додано різноманітні перешкоди для збільшення складності побудови траєкторії маршруту.



Рисунок 4.8 – Загальний вигляд змодельованої кінцевої точки

Як апробацію ідеї останнього кілометра використовували змодельований світ та СН моделі МР. У Gazebo [43] завантажувалися модель МР і змодельований світ із міською забудовою. Далі запускалася СН та обирали кінцеві точки для їх досягнення. СН вибудовувала траєкторію шляху, враховуючи існуючі перешкоди, які виявляє МР. Приклади апробації у змодельованому середовищі представлено на рис. 4.9.

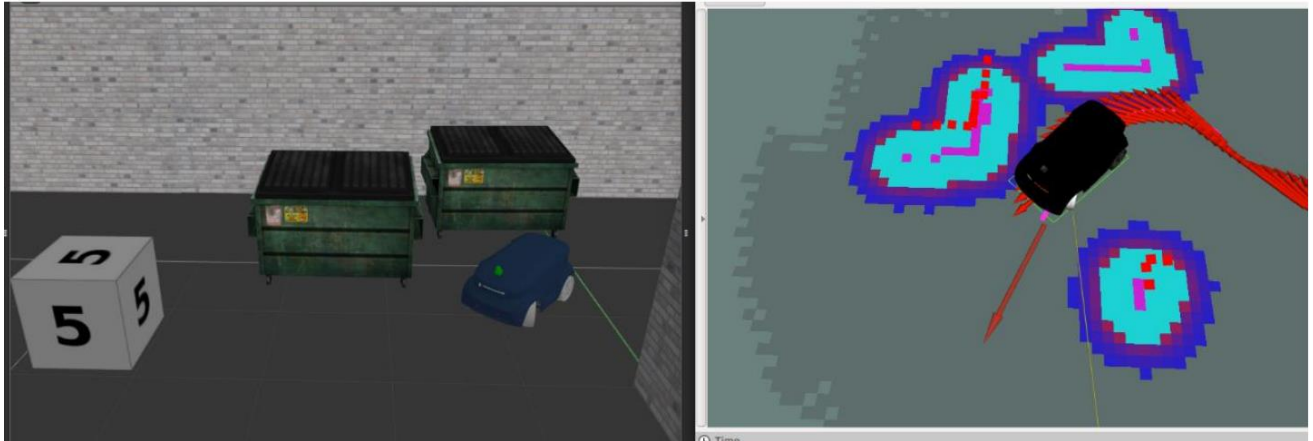


Рисунок 4.9 – Приклад апробації поставленого завдання

Після апробації ідеї розв'язання поставленого завдання в симуляції, можливим продовженням дослідження є підготовка МР «Junior» до дослідження у реальному середовищі. Під підготовкою мається на увазі низка робіт, пов'язаних з імплементацією розроблених для симуляційної моделі пакетів і різних налаштувань на реальному МР. Особливість даної підготовки полягає в тому, що в програмному забезпеченні робота (або ROS) містяться лише базові пакети для роботи з датчиками МР та керуванням його електродвигунами. Подальший розвиток проекту наступний:

- розробка СК рухом ІТЗ;
- впровадження розроблених пакетів на ІТЗ;
- налаштування СН для ІТЗ;
- випробовування СН в лабораторних умовах;
- створення штучної міської забудови для апробації задачі останнього

кілометра;

- апробація ІТЗ для розв'язання задачі останнього кілометра в штучній міській забудові.

Розробка СК рухом ІТЗ містить у собі розробку `node` для контролю ІТЗ. Даний етап необхідний, оскільки поточний пакет контролера ІТЗ `ur_hardware_driver` не надає точного керування ІТЗ. Тема `node ur_hardware_driver` приймає повідомлення типу `std_msgs/Float64`; такі повідомлення мають абстрактний характер. У разі, якщо користувач відправляє повідомлення на задні електродвигуни ІТЗ, то необхідно підбирати значення з діапазону $[6.0; 100.0]$; при цьому користувач повинен самостійно визначити, як збільшення значення повідомлення, що надсилається, впливає на зростання швидкості ІТЗ. У випадку, якщо користувач надсилає повідомлення на електродвигун рульової рейки, діапазон значень, який повертає на певний кут рульову систему ІТЗ приблизно складає $[20.0; 56.0]$. В такому випадку користувач також не може припустити, яке точне значення величини забезпечить поворот рульової системи ІТЗ на встановлений кут.

Є два підходи до реалізації даного етапу. Перший підхід полягає у тому, щоб створити новий контролер керування електродвигунами ІТЗ, замінюючи пакет, наданий компанією-виробником. Другий підхід полягає в створенні пакета-адаптера, `node` якого б приймала від користувача (або СН) чіткі команди рівня швидкості (у м/с) і величини кута повороту коліс (у радіанах), і конвертувала їх у потрібні значення для спеціального пакета `ur_hardware_driver`.

Імплементація ROS пакетів СН МР. Даний підрозділ об'єднує в собі одразу два етапи, які полягають у налаштуванні ІТЗ на базі пакетів і конфігурацій моделі. Розроблений для моделі пакет налаштування даних з лазерного далекоміра повинен бути аналогічно використовуваний для коригування даних реального давача.

Наступним завданням є налаштування параметрів моделі для реалістичності представлення МР «Junior» у середовищі Gazebo [43], бо в

задачах автономної навігації реальний МР буде оцінювати власне положення, сприйняття світу через віртуальну модель. Дане завдання є необхідним для будь-яких майбутніх досліджень, пов'язаних із ІТЗ.

Після налаштування СН на реальному МР його апробація може бути виконана також і в лабораторних умовах. Під лабораторними умовами маються на увазі офісні кабінети чи коридори навчального корпусу, площа яких дає змогу ІТЗ вільно пересуватися і виконувати розвороти.

4.6 Висновки до четвертого розділу

В даному розділі наведено дослідження системи навігації для мобільного робота «JUNIOR», а саме дослідження системи навігації в програмному середовищі ROS, коригування отриманих даних з лазера моделі мобільного робота, описано налаштування локального планувальника та карт вартості планувальника та представлено апробацію системи навігації в Rviz.

ВИСНОВКИ

Кваліфікаційна робота пропонує концепцію вирішення, так званої, проблеми останнього кілометра. Оскільки серйозні витрати на виконання доставок у кінцеву точку є проблемою для багатьох транспортних корпорацій, зараз розробляються шляхи вирішення даної проблеми. Один з інноваційних напрямків - використання МР в якості автономних інтелектуальних агентів, які будуть доставляти товари в кілька точок у населеному пункті.

МР «JUNIOR» є малорозмірним МР з рульовою системою керування. Така система керування також присутня в повнорозмірних автомобілях. Додатково, в ІТЗ присутня верхня кришка, під якою розташовується простір для можливого зберігання товарів для доставки. Такий ІТЗ є потенційним рішенням для поставленого завдання, проте для його розв'язання необхідно розробити відповідне базове і специфічне ПЗ МР: створення моделі МР, налаштування СН, апробація рішення останнього кілометра в симуляції.

Необхідність розробки базового ПЗ полягає в тому, що компанія-виробник надає мінімальний набір ROS-пакетів, які дають змогу лише отримувати дані із датчиків і посилати команди на електродвигуни МР.

Для створення моделі МР використовувався симулятор Gazebo, інтегрований у ROS. Створена модель МР відповідає реальним розмірам МР, його габаритним характеристикам, наявності датчиків.

Далі для моделі МР було налаштовано СН (локальний планувальник, глобальний планувальник, конфігурації costmap), проведено первинні експерименти в симуляції. Для апробації поставленого завдання було створено віртуальний світ з імітацією міської забудови і дев'ятьма різними кінцевими цілями. Запропоноване рішення демонструє концепт автоматизованого вирішення проблеми останнього кілометра і може бути розвинене в якості повноцінного проєкту на реальному МР «JUNIOR».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Робототехніка та мехатроніка: навч. посіб. / Л.І. Цвіркун, Г. Грулер ; під заг. ред. Л.І. Цвіркуна ; М-во освіти і науки України, Нац. гірн. ун-т. – 3-тє вид., переробл. і доповн. – Дніпро: НГУ, 2017. – 224 с. ISBN 978–966–350–645–6
2. Основи автоматики та робототехніки: Навчальний посібник/ А. М. Гуржій, А. Т. Нельга, В. М. Співак, О. С. Ітякін:–Дніпро:«Гарант СВ», 2021.- 243с. ISBN 978-617-627-156 -7
3. Мехатронні та телематичні системи автомобіля : навч. посіб. / В.Д. Мигаль. – Х. : Майдан, 2017. – 314 с. ISBN 978-966-372-665-6
4. BMW says self-driving car to be Level 5 capable by 2021 [Електронний ресурс]. – Режим доступу: <https://www.autonews.com/article/20170316/MOBILITY/170319877/bmw-says-self-driving-car-to-be-level-5-capable-by-2021> .
5. Future of Driving [Електронний ресурс]. – Режим доступу: <https://www.tesla.com/autopilot?redirect=no>
6. Uber [Електронний ресурс]. – Режим доступу: <https://www.uber.com/info/atg/technology/>
7. Apollo Open Platform [Електронний ресурс]. – Режим доступу: <http://apollo.auto/>
8. Next generation robotics // Christensen H., Okamura A., Mataric M., Kumar V., Hager G., Choset H. - *Procedia CIRP*. Volume 63, 2017, Pages 89-94.
9. Swarm Robotics Control and Communications: Imminent Challenges for Next Generation Smart Logistics // Wen J., He L., Zhu F. - *IEEE Communications Magazine*, Vol. 56, No. 7, 2018. pp. 102-107.
10. The Urban Last Mile Problem: Autonomous Drone Delivery to Your Balcony // Brunner G., Szebedy B., Tanner S., Wattenhofer R. - *IFAC-PapersOnLine*. Volume 55, Issue 10, 2022, Pages 3142-3147.
11. Robots in Space: Sharing Our World with Autonomous Delivery Vehicles //

Marks M. - SSRN Electronic Journal, 2019. DOI:10.2139/ssrn.3347466

12. Adoption of next generation robotics: A case study on Amazon // Jain D., Sharma Y. - A Case Research Journal, Vol. 3, 2017. pp. 9-23.

13. Амазон [Электронный ресурс]. – Режим доступа:
<https://www.amazon.com/>

14. Kiva Systems [Электронный ресурс]. – Режим доступа:
https://robotics.ua/robototekhnika-v-roznychnoi-torhovle/?srsltid=AfmBOorhhoJxXI3G_r-iAoPVGaVW5BYSMLm_Rmid2uL4Qp1n3KeP31aH

15. Marble [Электронный ресурс]. – Режим доступа:
<https://www.marble.io/>

16. Starship [Электронный ресурс]. – Режим доступа:
<https://www.starship.xyz/>

17. Kiwibot: Bite the future [Электронный ресурс]. – Режим доступа:
<https://www.kiwicampus.com/>

18. The Urban Last Mile Problem: Autonomous Drone Delivery to Your Balcony // Brunner G., Szebedy B., Tanner S., Wattenhofer R. - International Conference on Unmanned Aircraft Systems (ICUAS), 2019. DOI:10.1109/ICUAS.2019.8798337

19. Autonomous personal vehicle for the first and last-mile transportation services // Chong Z., Qin B., Bandyopadhyay T., Wongpiromsarn T., Rankin E., Ang M., Frazzoli E., Rus D., Hsu D., Low K. - IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS). 2011. pp. 253-260.

20. A novel visible light communication system for enhanced control of autonomous delivery robots in a hospital // Murai R., Sakai T., Kawano H., Matsukawa Y., Kitano Y., Honda Y., Campbell K.- IEEE/SICE International Symposium on System Integration (SII). 2012. pp. 510-516.

21. Design and implementation of an omnidirectional mobile robot platform with unified I/O interfaces // Wu J., Chaoshun L., Lijun Z., Ruifeng L., Guanglin W. -

In Mechatronics and Automation (ICMA), 2017 IEEE International Conference on. 2017. pp. 410-415.

22. Optimal Design for PID Controller Based on DE Algorithm in Omnidirectional Mobile Robot // Wu P., Kai W., Juzhong Z., Qi Z. - MATEC Web of Conferences, 2017. Vol. 95. P. 8-14.

23. Integrating Mecanum wheeled omnidirectional mobile robots in ROS // Feng Y., Chengjun D., Xia L., Xinghua Z. - In Robotics and Biomimetics (ROBIO), 2016. pp. 643-648.

24. Development of an omnidirectional mobile robot using a RGB-D sensor for indoor navigation // Yip H.M., Ho K.K., Chu M.H.A., Lai K.W. . - The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent. 2014. pp. 162-167.

25. Omnidirectional mobile robot-design and implementation // Doroftei I., Grosu V., Spinu V. - Bioinspiration and Robotics Walking and Climbing Robots. 2007.

26. Колеса Mecanum [Электронный ресурс]. – Режим доступа:
<https://robotgeeks.ru/collection/mecanum>

27. Omnidirectional wheeled mobile robots: wheel types and practical applications // Kanjanawanishkul K. - International Journal of Advanced Mechatronic Systems, Vol. 6, No. 6, 2015. pp. 289-302.

28. Analysis, design, and implementation of an omnidirectional mobile robot platform // Ismael O.Y., Hedley J. - American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), Vol. 22, No. 1, 2016. pp. 195-209.

29. Analysis and control of an omnidirectional mobile robot // Ren C., Ma S. - IEEE ISR 2013. 2013. pp. 1 -6.

30. Dynamic modeling and analysis of an omnidirectional mobile robot // Ren C., Ma S. - 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013. pp. 4860-4865.

31. Improved mecanum wheel design for omni-directional robots // Diegel O., Aparna B., Glen B., Johan P., Sylveste T. - In Proc. 2002 Australasian Conference on Robotics and Automation. 2002. pp. 117-121.

32. Modified mecanum wheels for traversing rough terrains // Ramirez-Serrano A., Roman K. - 2010 Sixth International Conference on Autonomic and Autonomous Systems. 2010. pp. 97-103.

33. Designing omnidirectional mobile robot with mecanum wheel // Salih J.E.M., Mohamed R., Sazali Y., Abdul H.A., Mohd R.M. - American Journal of Applied Sciences, Vol. 3, No. 5, 2006. pp. 1831 -1835.

34. Autonomous mobile robot platform with multi-variant task-specific endeffector and voice activation // Tapia J., Eric W., Patrick B., Aldo J., Ethan C., John P., Dan C., Mo J., Benjamin C. - 2016 World Automation Congress (WAC). 2016. pp. 1 -6.

35. Remote control system for a mobile platform with four Mecanum wheels // Sarmiento L., Francisco N., Ricardo S.M., João S., João, Sena Esteve. - International Journal of Mechatronics and Applied Mechanics, No. 1, 2017. pp. 274-281.

36. Doing good across organizational boundaries: Sustainable supply chain practices and firms' financial risk // Lam H.K. - International Journal of Operations & Production Management, Vol. 38, No. 12, 2018. pp. 2389-2412.

37. What does industry 4.0 mean to supply chain? // Tjahjono B., Esplugues C., Ares E., Pelaez G. - Procedia Manufacturing, Vol. 13, 2017. pp. 1175-1182.

38. Emerging technologies to support supply chain management // Singh N. - Communications of the ACM, Vol. 46, No. 9, 2003. pp. 243-247.

39. Supply chain 4.0 in consumer goods // Alicke K., Rexhausen D., Seyfert A.- Mckinsey & Company, 2017.

40. Стартап Nuro, який планує запуснути безпілотну доставку товарів, отримав \$92 млн інвестицій [Електронний ресурс]. – Режим доступу:

<https://mind.ua/news/20181105-startap-nuro-yakij-planue-zapustiti-bezpilotnu-dostavku-tovariv-otrimav-92-mln-investicij>

41. Мобільний робот Robby 2 [Електронний ресурс]. – Режим доступу:
<https://roby.com.ua/?srsltid=AfmBOoqvap5QboP05snp5U8p6nTGSYA6U9irpvCniX0KB3yAmazGqmkm>
42. ROS: an open-source Robot Operating System // Quigley M., Conley K., Gerkey B., Faust J., Foote T., Leibs J., Wheeler R., Ng A.Y. - ICRA workshop on open source software. Kobe, Japan. 2009. Vol. 3. P. 5.
43. Simulate before you build [Електронний ресурс]. – Режим доступу:
<https://gazebosim.org/home>
44. Використання rviz и rqt [Електронний ресурс]. – Режим доступу:
<https://clover.coex.tech/ru/rviz.html>
45. Павленко І.І., Мажара В.А. Роботизовані технологічні комплекси: Навчальний посібник. – Кіровоград: КНТУ, 2010. – 392 с.
46. Робототехнічні системи: проєктування і моделювання [Електронний ресурс]: навч. Посіб. для студ. спеціальності 126 «Інформаційні системи та технології» / М. М. Поліщук, М.М. Ткач; КПІ ім. Ігоря Сікорського. Електронні текстові дані (1 файл: 41,6 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2021. - 112 с.
47. Синтез робототехнічних систем в машинобудуванні: Підручник / Л. Є. Пелевін, К. І. Почка, О. М. Гаркавенко, Д. О. Міщук, І. В. Русан. – К.:ТОВ «НВП «Інтерсервіс»», 2016. – 258 с.
48. Основи мехатроніки [Електронний ресурс] : навчальний посібник для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / КПІ ім. Ігоря Сікорського ; уклад.: С. М. Пересада, М. В. Пушкар. – Електронні текстові дані (1 файл: 1,87 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 137 с.
49. Towards DJI Phantom 4 Realistic Simulation with Gimbal and RC Controller in ROS/Gazebo Environment // Sagitov A., Gerasimov Y. - 10th International Conference on Developments in eSystems Engineering (DeSE). 2017. pp. 262-266.
50. ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D

model of indoor environment // Afanasyev I., Sagitov A., Magid E. - International Conference on Advanced Concepts for Intelligent Vision Systems. 2015. pp. 273-283.

51. Modelling a crawler-type UGV for urban search and rescue in Gazebo environment // Sokolov M., Afanasyev I., Lavrenov R., Sagitov A., Sabirova L., Magid E. - Artificial Life and Robotics (ICAROB 2017), International Conference on. 2017. pp. 360-362.

52. Robot simulation physics validation // Pepper C., Balakirsky S., Scrapper C. - Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems. 2007. pp. 97-104.

53. Fast simulation of vehicles with nondeformable tracks // Pecka M., Zimmermann K., Svoboda T. - 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017. pp. 6414-6419.

54. A simulation system based on ros and gazebo for robocup middle size league // Yao W., Dai W., Xiao J., Lu H., Zheng Z.- IEEE international conference on robotics and biomimetics (ROBIO). 2015. pp. 54-59.

55. Manipulation task simulation using ROS and Gazebo // Qian W., Xia Z., Xiong J., Gan Y., Guo Y., Weng S., Deng H., Hu Y., Zhang J. - IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). 2014. pp. 2594-2598.

56. Robocuprescue 2014-robot league team hector darmstadt (Germany) // Kohlbrecher S., Meyer J., Graber T., Petersen K., Von Stryk O., Klingauf U. RoboCupRescue 2014.

57. A dynamic simulator for underwater vehicle-manipulators // Kermorgant O. - International Conference on Simulation, Modeling, and Programming for Autonomous Robots. 2014. pp. 25-36.

58. V-REP: A versatile and scalable robot simulation framework // Rohmer E., Singh S.P., Freese M. - 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013. pp. 1321 -1326.

59. Aurora Driver for Freight —coming to Texas in 2024 [Электронный ресурс]. – Режим доступа:

<https://aurora.tech/>

60. Ackermann steering geometry [Електронний ресурс]. – Режим доступу:

https://en.wikipedia.org/wiki/Ackermann_steering_geometry

61. Ладанюк А. П., Архангельська К. С., Власенко Л. О. Теорія автоматичного керування технологічними об'єктами: Навч. посіб. / – К.: НУХТ, 2014. – 274 с.

62. Автоматизація технологічних процесів і системи автоматичного керування: Навчальний посібник /Барало О.В., Самойленко П.Г., Гранат С.Є., Ковальов В.О. – К.: Аграрна освіта, 2010. – 557 с. ISBN 978-966-2007-12-1.

63. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К.: Либідь, 2007. – 656 с.

64. Optimum settings for automatic controllers // Ziegler J.G., Nichols N.B. - trans. ASME, Vol. 64, No. 11, 1942.

65. Move_base [Електронний ресурс]. – Режим доступу:

http://wiki.ros.org/move_base?distro=melodic

66. Kinodynamic trajectory optimization and control for car-like robots // Rosmann C., Hoffmann F., Bertram T. - 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017. pp. 5681 -5686.

67. Trajectory modification considering dynamic constraints of autonomous robots // Rosmann C., Feiten W., Wosch T., Hoffmann F., Bertram T. ROBOTIK 2012; 7th German Conference on Robotics. 2012. pp. 1 -6.

68. Efficient trajectory optimization using a sparse model // Rosmann C., Feiten W., Wosch T., Hoffmann F., Bertram T. 2013 European Conference on Mobile Robots. 2013. pp. 138-143.

69. Integrated online trajectory planning and optimization in distinctive topologies // Rosmann C., Hoffmann F., Bertram T. - Robotics and Autonomous Systems, Vol. 88, 2017. pp. 142-153.

70. Planning of multiple robot trajectories in distinctive topologies // Rosmann C., Hoffmann F., Bertram T. - 2015 European Conference on Mobile Robots (ECMR).

2015. pp. 1 -6.

71. Limpert N., Schiffer S., Ferrein A. A local planner for Ackermann-driven vehicles in ROS SBPL // 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2015. pp. 172-177.

72. costmap_2d [Электронный ресурс]. – Режим доступа:
wiki.ros.org/costmap_2d

Додаток А

Стаття

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»

ISSN 2219-9365

<https://doi.org/10.31891/2219-9365-2024-79-28>

УДК 681.5:629.3

СПОРИЙ Олександр
Хмельницький національний університет

МАЙДАН Павло
Хмельницький національний університет
<https://orcid.org/0000-0003-3319-8730>
maidanps@gmail.com

МАКАРИШКІН Денис
Хмельницький національний університет
<https://orcid.org/0000-0003-3447-811X>
makaryshkind@khmnu.edu.ua

СОКОЛАН Юлія
Хмельницький національний університет
<https://orcid.org/0000-0002-0273-5719>
sokolan.julia@gmail.com

МОДЕЛЮВАННЯ ОСНОВНИХ ЕЛЕМЕНТІВ КОНСТРУКЦІЇ НА ПРИКЛАДІ РОБОМОБІЛЯ «JUNIOR»

Загально відомо, що використання симуляторів допомагає здешевити та прискорити розробку робото-технічних систем, є безпечнішим інструментом первинної верифікації ідей та алгоритмів, а також надає можливість виконувати тестування нових концепцій та алгоритмів, незалежно від наявності необхідного обладнання у наукової групи. Моделювання може бути використане для всіх типів робототехнічних систем та завдань на початкових етапах розвитку проекту.

Моделювання робомобіля було проведене в середовищі ROS Gazebo. Підхід до моделювання був наступний: врахування фізичних характеристик робомобіля, врахування давачів та характеристик, використання CAD-моделей робомобіля для моделювання. При створенні моделі використовувалися середовища для програмування: ROS, Gazebo. Для того, щоб модель візуально відповідала реальному робомобілю, були використані частини для даних елементів та деяких давачів, САПР-модель є у відкритому доступі. Усі зчленування моделі були задані у пропорціях, що відповідають пропорціям та розмірам реального робомобіля. Щоб адаптувати інерційні характеристики робомобіля до моделі, було розраховано спрощений інерційний тензор на зчленування.

В роботі виконано моделювання роботи робомобіля для переміщення виробів, а саме на прикладі робомобіля «Junior» фірми AURORA, створена модель відповідає реальним розмірам, його габаритним характеристикам, наявності встановлених давачів. Необхідність розробки базового програмного забезпечення полягає в тому, що компанія-виробник надає мінімальний набір ROS-пакетів, які дають змогу лише отримувати дані із давачів і посилати команди на електродвигуни робомобіля.

Ключові слова: робомобіль, Junior, Ackermann drive, симулятор Gazebo, моделювання, давачі.

SPORYI Oleksandr, MAIDAN Pavlo, MAKARYSHKIN Denys, SOKOLAN Iuliia
Khmelnitskyi National University

MODELING OF THE MAIN STRUCTURAL ELEMENTS ON THE EXAMPLE OF THE ROBOT VEHICLE «JUNIOR»

It is well known that the use of simulators helps to reduce the cost and speed up the development of robotic systems, is a safer tool for the initial verification of ideas and algorithms, and will allow testing of new concepts and algorithms, regardless of the availability of the necessary equipment for the research team. Modeling can be used for all types of robotic systems and tasks in the initial stages of project development.

The robot vehicle was modeled in the ROS Gazebo environment. The approach to modeling was as follows: taking into account the physical characteristics of the robot vehicle, taking into account sensors and characteristics, using CAD models of the robot vehicle for modeling. When creating the model, the following programming environments were used: ROS, Gazebo. To make the model visually consistent with a real robot vehicle, parts for these elements and some sensors were used; the CAD model is publicly available. All the joints of the model were set in proportions that correspond to the proportions and dimensions of the real robot vehicle. To adapt the inertial characteristics of the robot vehicle to the model, a simplified inertial tensor for the joints was calculated.

In this work, we model the operation of a robot vehicle for moving products, namely, on the example of the AURORA Junior robot vehicle, the created model corresponds to the real size, its overall characteristics, and the presence of installed sensors. The need to develop basic software is due to the fact that the manufacturer provides a minimal set of ROS packages that only allow you to receive data from sensors and send commands to the robot's electric motors.

Keywords: robot vehicle, Junior, Ackermann drive, Gazebo simulator, modeling, sensors.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОКІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

В останні десятиліття широкое використання симуляції стало невіддільною частиною досліджень у галузі робототехніки [1-4]. Використання симуляторів допомагає здешевити та прискорити розробку робото-

International Scientific-technical journal
«Measuring and computing devices in technological processes» 2024, Issue 3

технічних систем (PТС), є безпечнішим інструментом первинної верифікації ідей та алгоритмів, а також надає можливість виконувати тестування нових концепцій та алгоритмів, незалежно від наявності необхідного обладнання у науковій групі. Моделювання може використовуватись для всіх типів PТС та завдань на початкових етапах розвитку проєкту.

Симулятори стали більш прогресивними інструментами, які надають можливість виконати відтворення складних середовищ та використовувати фізику, яку визначив сам розробник, що, в свою чергу, дає можливість створювати моделі PТС, які поведуться максимально близько до реальної поведінки робомобілів (PМ). Gazebo є одним із найпопулярніших 3D-симуляторів PМ, який успішно використовується для симуляції БПЛА [5] та безпілотного наземного робота (БНР) [6-9], та для проведення різноманітних експериментів, включно з тестуванням базових рухів PМ, плануванням траєкторії шляху та колаборативною взаємодією з іншими PМ [10], експериментами з маніпуляторами [11] та моделюванням сценаріїв пошуково-рятувальних операцій [12]. Інші спеціалізовані симулятори (наприклад, UWSim [13], V-REP [14]) дають змогу моделювати водне середовище для автономних підводних роботів.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

PМ «Junior» є розробкою фірми «AURORA ROBOTICS», загальний вигляд, якого представлено на рисунку 1. За своєю зовнішньою будовою PМ «Junior» є класичною машиною із Ackermann drive або системою приводів та рульовою рейкою для здійснення керування рухом машини [15]. Даний PМ було розроблено з метою навчання студентів принципів автономного руху БНР. Такий навчально-налагоджувальний комплекс є зручним та оптимальним рішенням для роботи студентів: невеликі габарити та загальна вага машини (відносно повнорозмірних машин, які використовуються для розробки автономного функціонування компанії Uber чи Google), відкрита програмна система на базі ОС Linux Ubuntu 16.04 LTS та ROS, а також відкрита апаратна система.

Відкрита апаратна система передбачає вільний фізичний доступ до бортових давачів PМ, влаштування електричного ланцюга, можливості зміни апаратного складу конструкції, вдосконалення та заміни будь-яких компонентів (як давачів, і базових елементів – акумулятора, коліс, мотор-коліс тощо) [16].



Рис. 1. Загальний вигляд PМ «Junior» фірми AURORA ROBOTICS

З PМ передбачається використання спеціального полігону, побудованого на кшталт реальної проїжджої частини міста з відповідними елементами: дорожніми знаками, світлофорами, тунелем, дорожньою розміткою та, відповідно, пішохідним переходом. Завдяки всьому комплексу даного навчального комплексу студенти можуть розробляти алгоритми автономного руху PМ у штучному середовищі дорожнього оточення [16].

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою роботи є: розроблення алгоритмів переміщення та автономної навігації для PМ «Junior», для чого побудовано модель у симуляторі Gazebo і проведено тестування алгоритмів керування.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Моделювання PМ було проведене в середовищі ROS Gazebo. Підхід до моделювання був наступний: врахування фізичних характеристик PМ, врахування давачів PМ та характеристик, використання САD-моделей PМ для моделювання. При створенні моделі PМ використовувалися наступні середовища для програмування: ROS, Gazebo [17].

Основними елементами PМ «Junior», що представляють модель, були обрані наступні: зовнішній каркас PМ - `base_link`, переднє праве колесо - `right_steer_wheel`, переднє ліве колесо - `left_steer_wheel`, права частина рульової рейки, частина майбутнього механізму системи підвіски PМ згідно Ackermann drive - `right_steer_link`, ліва частина рульової рейки, частина майбутнього механізму системи підвіски PМ згідно з Ackermann drive - `drive_right_drive_left_steer_link`, заднє ліве колесо - `left_drive_wheel`.

Для того, щоб модель РМ візуально відповідала реальному РМ, були використані частини для даних елементів та деяких давачів, розглянутих нижче (САПР-модель є у відкритому доступі) (рис. 2).

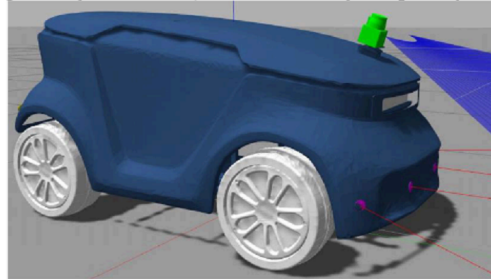


Рис. 2. Загальний вигляд моделі РМ «Junior»

Усі зчленування моделі були задані у пропорціях, що відповідають пропорціям та розмірам реального РМ. Щоб адаптувати інерційні характеристики РМ до моделі, було розраховано спрощений інерційний тензор на зчленування. У процесі було замінено складну структуру каркаса РМ (base_link) суцільним кубоїдним об'єктом (solid cuboid); структури рульових коліс РМ (right_steer_wheel, left_steer_wheel), задніх коліс замінені на суцільний циліндр (solid cylinder) для обчислення наближених тензорів інерції зчленувань моделі. Колізії зчленувань були зазначені з урахуванням структури всіх зчленувань і в тому ж розмірі, що й зчленування.

```
#include "ros/ros.h"
#include "std_msgs/Float64.h"
#include <utility>
#include <cmath>
#include <sstream>
#include <string>
#include "ackermann_msgs/AckermannDriveStamped.h"
ros::Publisher left_pub;
ros::Publisher right_pub;
std_msgs::Float64 fi_r;
std_msgs::Float64 fi_l;
double const lateral_distance = 0.7;
double const front_distance = 0.54;
double const pi2 = 1.5708;
double radiusCalc (double angle)
{ return lateral_distance/tan(angle); }
double formulaCalc (double radius, bool sign)
{ if (sign)
{ return lateral_distance / (radius + (front_distance/2)); }
else
{ return lateral_distance / (radius - (front_distance/2)); } }
void steeringCallback(const ackermann_msgs::AckermannDriveStamped::ConstPtr &msg)
{ double radius = radiusCalc(msg->drive.steering_angle);
ROS_INFO("---Calculated radius : %.4f", radius);
if (msg->drive.steering_angle >= 0 && msg->drive.steering_angle <= 1.20)
{ fi_r.data = atan(formulaCalc(radius, false));
fi_l.data = atan(formulaCalc(radius,true));
ROS_INFO("---Calculated angles : left wheel %.4f, right wheel %.4f", fi_r.data,fi_l.data);
left_pub.publish(fi_r)
right_pub.publish(fi_l); }
if (msg->drive.steering_angle <= 0 && msg->drive.steering_angle >= -1.20)
{ ROS_INFO("---Calculated angles : right wheel %.4f, left wheel %.4f", fi_r.data,fi_l.data);
fi_r.data = atan(formulaCalc(radius,true));
fi_l.data = atan(formulaCalc(radius, false));
left_pub.publish(fi_l);
right_pub.publish(fi_r); }
if (msg->drive.steering_angle > 1.20)
{ double radius_exception(1.2);
ROS_INFO("---Calculated radius exception : %.4f", radius);
```

```

fi_r.data = atan(formulaCalc(radius_exception, false));
fi_l.data = atan(formulaCalc(radius_exception, true));
ROS_INFO("---Calculated angles : left wheel %.4f, right wheel %.4f", fi_r.data, fi_l.data);
left_pub.publish(fi_r);
right_pub.publish(fi_l); }
if (msg->drive.steering_angle < -1.2)
{ double radius_exception(-1.2);
ROS_INFO("---Calculated radius exception : %.4f", radius);
fi_r.data = atan(formulaCalc(radius_exception, false));
fi_l.data = atan(formulaCalc(radius_exception, true));
ROS_INFO("---Calculated angles : left wheel %.4f, right wheel %.4f", fi_r.data, fi_l.data);
left_pub.publish(fi_r);
right_pub.publish(fi_l); }
}
int main(int argc, char **argv)
{ ros::init(argc, argv, "steer_remap");
ros::NodeHandle n;
std::string lw_topic, rw_topic, central_steer_topic;
central_steer_topic = "/junior_car/ackermann cmd";
lw_topic = "/junior_car/SteerLeft_controller/command";
rw_topic = "/junior_car/SteerRight_controller/command";
if (n.getParam("left_wheel_steer_topic", lw_topic))
{ ROS_INFO("Got param: %s", lw_topic.c_str()); }
if (n.getParam("right_wheel_steer_topic", rw_topic))
{ ROS_INFO("Got param: %s", rw_topic.c_str()); }
if (n.getParam("right_wheel_steer_topic", central_steer_topic))
{ ROS_INFO("Got param: %s", central_steer_topic.c_str()); }
left_pub = std::move(n.advertise<std_msgs::Float64>(lw_topic, 1024));
right_pub = std::move(n.advertise<std_msgs::Float64>(rw_topic, 1024));
ros::Subscriber sub = n.subscribe(central_steer_topic, 1024, steeringCallback);
ros::spin();
return 0;
}

```

Рис. 3. Приклад програми контролю моделі РМ «Junior»

Моделювання суглобів, моторів та трансмісія РМ.

URDF формат дозволяє описувати наступні типи суглобів: призматичний (prismatic) – ковзний суглоб, що зміщується вздовж вказаної осі та володіє обмеженим діапазоном руху, заданими верхньою та нижньою границями, обертальний без лімітів (continuous) – обертальний суглоб (або шарнірний), який обертається щодо вказаної осі та не має верхніх та нижніх границь, плаваючий (floating) – суглоб, що володіє рухом у всіх шести ступенях свободи, обертальний з лімітами (revolute) – обертальний суглоб (шарнірний), який обертається щодо заданої осі та має обмежений діапазон руху, заданий верхньою та нижньою границею і нерухомий (fixed) – фіксований суглоб, який насправді не є суглобом. Усі ступеня свободи даного типу суглоба заблоковані, він вимагає вказівки осі руху, границь. Даний тип суглобу використовується для жорсткого зчеплення зчленування РМ між собою, а також приєднання давачів РМ до його зчленування.

Обертальний тип суглоба без обмежень був використаний до суглобів передніх та задніх коліс (left_steer_wheel_joint, left_drive_wheel_joint, right_steer_wheel_joint, right_drive_wheel_joint). Обертальний тип суглоба із обмеженням був використано до кермових механізмів моделі (left_steer_joint and right_steer_joint), для подальшої розробки системи підвіски та керування моделлю згідно з принципом Ackermann drive [15].

Таким чином, для здійснення обчислень величини кута повороту передніх коліс (α_1 та α_0), необхідно було знати лінійні геометричні параметри РМ: відстань між передніми колесами ($T = 0,54\text{ м}$) і бічну відстань між переднім та заднім колесом ($L = 0,70\text{ м}$). Дані параметри були використані як константи при розрахунку величин кутів повороту коліс (рис. 4).

Так як на реальному РМ використовуються два мотори для виконання руху задніх коліс та один мотор для організації рульового механізму, були налаштовані наступні ROS-контролери для суглобів: SteerRight controller і SteerLeft controller - контролери положення (або ROS JointPositionController), EffortDriveRight Control - контролери зусиль.

Контролери положення відносяться до суглобів кермового механізму, контролери зусиль відносяться до суглобам задніх коліс (оскільки на реальному РМ моторами оснащені два задніх колеса). Рисунок 5 демонструє фінальну схему роботи контролерів середовища ROS.

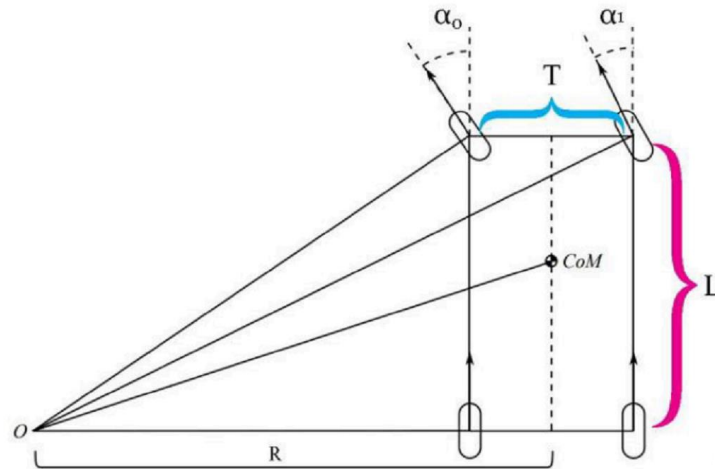


Рис. 4. Загальний вигляд геометрії Ackermann drive для PM [15];
L - бічна відстань між серединами переднього та заднього коліс PM; R - відстань від поздовжньої осі PM до центру повороту; T - відстань між серединою передньої пари коліс

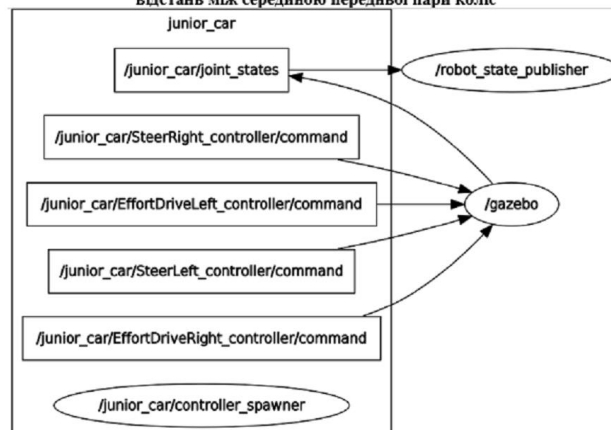


Рис. 5. Загальний вигляд візуалізації в rqt_graph тем повідомлень моделі PM «Junior»
(теми повідомлень давачів не відображені)

Моделювання давачів наведено табл. 1 та представляє список давачів PM «Junior» та їх імплементація в програмних середовищах ROS та Gazebo.

Таблиця 1.

Давачі PM та плагіни в середовищах ROS, Gazebo

| Давачі PM та їх плагіни | Пакет в ROS | Пакет в Gazebo |
|-----------------------------------|-----------------------|------------------|
| Лідар | hokuyo node | laser controller |
| Microsoft Kinect | freenect stack | camera plugin |
| GPS | ur nmea driver | novatel gps sim |
| УЗ | ur rangefinder driver | sonar |
| Вимірювальний інерційний пристрій | myahrs driver | imu plugin |

Моделі давачів були поміщені в тому ж положенні та орієнтації, як і на реальному PM, для отримання точних даних із давачів. Всі плагіни давачів були налаштовані згідно з технічним описом існуючих давачів, для виконання візуалізації давачів використовувалося ПЗ Rviz (входить у фреймворк ROS).

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

В роботі виконано моделювання роботи робомобіля для переміщення виробів, а саме на прикладі робомобіля «Junio» фірми AURORA, створена модель відповідає реальним розмірам, його габаритним характеристикам, наявності давачів.

Необхідність розробки базового програмного забезпечення полягає в тому, що компанія-виробник надає мінімальний набір ROS-пакетів, які дають змогу лише отримувати дані із давачів і посылати команди на електродвигуни робомобіля.

Далі для моделі робомобіля буде налаштовано систему навігації (локальний планувальник, глобальний планувальник, конфігурації costmap), проведено первинні дослідження в симуляції. Для апробації поставленого завдання буде також створено віртуальний світ з імітацією міської забудови і дев'ятьма різними кінцевими цілями. Запропоноване рішення може продемонструвати концепт автоматизованого завдання останнього кілометру і може бути розвинене в якості повноцінного проєкту на реальному робомобілі «Junio».

Література

1. Павленко І.І., Мажара В.А. Роботизовані технологічні комплекси: Навчальний посібник. – Кіровоград: КНТУ, 2010. – 392 с.
2. Робототехнічні системи: проєктування і моделювання [Електронний ресурс]: навч. Посіб. для студ. спеціальності 126 «Інформаційні системи та технології» / М. М. Поліщук, М.М. Ткач; КПІ ім. Ігоря Сікорського. Електронні текстові дані (1 файл: 41,6 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2021. - 112 с.
3. Синтез робототехнічних систем в машинобудуванні: Підручник / Л. Є. Пелевін, К. І. Почка, О. М. Гаркавенко, Д. О. Міщук, І. В. Русан. – К.:ТОВ «НВП «Інтерсервіс»», 2016. – 258 с.
4. Основи мехатроніки [Електронний ресурс] : навчальний посібник для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / КПІ ім. Ігоря Сікорського ; уклад.: С. М. Пересада, М. В. Пушкар. – Електронні текстові дані (1 файл: 1,87 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 137 с.
5. Towards DJI Phantom 4 Realistic Simulation with Gimbal and RC Controller in ROS/Gazebo Environment // Sagitov A., Gerasimov Y. - 10th International Conference on Developments in eSystems Engineering (DeSE). 2017. pp. 262-266.
6. ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment // Afanasyev I., Sagitov A., Magid E. - International Conference on Advanced Concepts for Intelligent Vision Systems. 2015. pp. 273-283.
7. Modelling a crawler-type UGV for urban search and rescue in Gazebo environment // Sokolov M., Afanasyev I., Lavrenov R., Sagitov A., Sabirova L., Magid E. - Artificial Life and Robotics (ICAROB 2017), International Conference on. 2017. pp. 360-362.
8. Robot simulation physics validation // Pepper C., Balakirsky S., Scrapper C. - Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems. 2007. pp. 97-104.
9. Fast simulation of vehicles with nondeformable tracks // Pecka M., Zimmermann K., Svoboda T. - 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017. pp. 6414-6419.
10. A simulation system based on ros and gazebo for robocup middle size league // Yao W., Dai W., Xiao J., Lu H., Zheng Z.- IEEE international conference on robotics and biomimetics (ROBIO). 2015. pp. 54-59.
11. Manipulation task simulation using ROS and Gazebo // Qian W., Xia Z., Xiong J., Gan Y., Guo Y., Weng S., Deng H., Hu Y., Zhang J. - IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). 2014. pp. 2594-2598.
12. Robocuprescue 2014-robot league team hector darmstadt (Germany) // Kohlbrecher S., Meyer J., Graber T., Petersen K., Von Stryk O., Klingauf U. RoboCupRescue 2014.
13. A dynamic simulator for underwater vehicle-manipulators // Kermorgant O. - International Conference on Simulation, Modeling, and Programming for Autonomous Robots. 2014. pp. 25-36.
14. V-REP: A versatile and scalable robot simulation framework // Rohmer E., Singh S.P., Freese M. - 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013. pp. 1321 -1326.
15. Ackermann steering geometry [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Ackermann_steering_geometry
16. Aurora Driver for Freight —coming to Texas in 2024 [Електронний ресурс]. – Режим доступу: <https://aurora.tech/>
17. Simulate before you build [Електронний ресурс]. – Режим доступу: <https://gazebosim.org/home>

References

1. Pavlenko I.I., Mazhara V.A. Robotyzovani tekhnolohichni komplekxy: Navchalnyi posibnyk. – Kirovohrad: KNTU, 2010. – 392 s.
2. Robototekhnichni systemy: proiektuvannya i modeliuvannya [Elektronnyi resurs]: navch. Posib. dlia stud. spetsialnosti 126 «Informatsiini systemy ta tekhnolohii» / M. M. Polishchuk, M.M. Tkach; KPI im. Ihoria Sikorskoho. Elektronni tekstovi dani (1 fail: 41,6 Mбайт). Kyiv: KPI im. Ihoria Sikorskoho, 2021. - 112 s.

3. Syntez robototekhnichnykh system v mashynobuduvanni: Pidruchnyk / L. Ye. Pelevin, K. I. Pochka, O. M. Harkavenko, D. O. Mishchuk, I. V. Rusan. – K.:TOV «NVP «Interservis», 2016. – 258 s.
4. Osnovy mekhatroniky [Elektronnyi resurs] : navchalnyi posibnyk dlia studentiv spetsialnosti 141 «Elektroenerhetyka, elektrotehnika ta elektromekhanika» / KPI im. Ihoria Sikorskoho ; uklad.: S. M. Peresada, M. V. Pushkar. – Elektronni tekstovi dani (1 fail: 1,87 Mбайт). – Kyiv : KPI im. Ihoria Sikorskoho, 2020. – 137 s.
5. Towards DJI Phantom 4 Realistic Simulation with Gimbal and RC Controller in ROS/Gazebo Environment // Sagitov A., Gerasimov Y. - 10th International Conference on Developments in eSystems Engineering (DeSE). 2017. pp. 262-266.
6. ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment // Afanasyev I., Sagitov A., Magid E. - International Conference on Advanced Concepts for Intelligent Vision Systems. 2015. pp. 273-283.
7. Modelling a crawler-type UGV for urban search and rescue in Gazebo environment // Sokolov M., Afanasyev I., Lavrenov R., Sagitov A., Sabirova L., Magid E. - Artificial Life and Robotics (ICAROB 2017), International Conference on. 2017. pp. 360-362.
8. Robot simulation physics validation // Pepper C., Balakirsky S., Scrapper C. - Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems. 2007. pp. 97-104.
9. Fast simulation of vehicles with nondeformable tracks // Pecka M., Zimmermann K., Svoboda T. - 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017. pp. 6414-6419.
10. A simulation system based on ros and gazebo for robocup middle size league // Yao W., Dai W., Xiao J., Lu H., Zheng Z. - IEEE international conference on robotics and biomimetics (ROBIO). 2015. pp. 54-59.
11. Manipulation task simulation using ROS and Gazebo // Qian W., Xia Z., Xiong J., Gan Y., Guo Y., Weng S., Deng H., Hu Y., Zhang J. - IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). 2014. pp. 2594-2598.
12. Robocuprescue 2014-robot league team hector darmstadt (Germany) // Kohlbrecher S., Meyer J., Graber T., Petersen K., Von Stryk O., Klingauf U. RoboCupRescue 2014.
13. A dynamic simulator for underwater vehicle-manipulators // Kermorgant O. - International Conference on Simulation, Modeling, and Programming for Autonomous Robots. 2014. pp. 25-36.
14. V-REP: A versatile and scalable robot simulation framework // Rohmer E., Singh S.P., Freese M. - 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013. pp. 1321-1326.
15. Ackermann steering geometry [Elektronnyi resurs]. – Режим доступу: https://en.wikipedia.org/wiki/Ackermann_steering_geometry
16. Aurora Driver for Freight —coming to Texas in 2024 [Elektronnyi resurs]. – Режим доступу: <https://aurora.tech/>
17. Simulate before you build [Elektronnyi resurs]. – Режим доступу: <https://gazebosim.org/home>

Додаток В
(вибірковий)

Таблиця В.1 – Лінійні розміри МР

| Найменування | Розмір, м |
|--|-----------|
| Діаметр колеса диска | 0,205 |
| Діаметр всього колеса | 0,26 |
| Поздовжня відстань між колесами | 0,71 |
| Передня та задня відстань між колесами | 0,527 |
| Поздовжня довжина всієї машини | 1,105 |
| Висота машини | 0,535 |

Таблиця В.2 - Вагові характеристики МР

| Найменування | Вага (кг) |
|----------------------------------|-----------|
| Колесо | 0,9 |
| Зовнішній корпус МР | 8,5 |
| Кришка МР | 3,0 |
| Вага лазерного далекоміра Нокуйо | 0,16 |
| Вага камери Kinect | 0,45 |
| Вага всього МР | 43,5 |

Таблиця В.3 - Давачі робомобіля та плагіни в середовищах ROS, Gazebo

| Датчики робомобіля та їх плагіни | Пакет в ROS | Пакет в Gazebo |
|-----------------------------------|-----------------------|------------------|
| Лідар | hokuyo_node | laser_controller |
| Microsoft Kinect | freenect_stack | camera_plugin |
| GPS | ur_nmea_driver | novatel_gps_sim |
| УЗ | ur_rangefinder_driver | sonar |
| Вимірювальний інерційний пристрій | myahrs_driver | imu_plugin |

Таблиця В.4 – Метод Зиглера-Нікольса для налаштування ПД-регулятора

| Тип регулятора | K_p | T_i | T_d |
|---|-----------|-----------|-----------|
| П | $0.5K_u$ | - | - |
| ПІ | $0.45K_u$ | $T_u/1.2$ | - |
| ПД | $0.8K_u$ | - | $T_u/8$ |
| Класичний ПД | $0.6K_u$ | $T_u/2$ | $T_u/8$ |
| Інтегральне правило Пессена | $0.7K_u$ | $T_u/2.5$ | $3T_u/20$ |
| Допущення перевищення бажаної границі | $0.33K_u$ | $T_u/2$ | $T_u/3$ |
| Недопущення перевищення бажаної границі | $0.2K_u$ | $T_u/2$ | $T_u/3$ |

Додаток Г
(вибірковий)

Програмування контролю моделі МР «Junior»

```

#include "ros/ros.h"
#include "std_msgs/Float64.h"
#include <utility>
#include <cmath>
#include <sstream>
#include <string>
#include "ackermann_msgs/AckermannDriveStamped.h"
ros::Publisher left_pub;
ros::Publisher right_pub;
std_msgs::Float64 fi_r;
std_msgs::Float64 fi_l;
double const lateral_distance = 0.7;
double const front_distance = 0.54;
double const pi2 = 1.5708;
double radiusCalc (double angle)
{ return lateral_distance/tan(angle); }
double formulaCalc (double radius, bool sign)
{ if (sign)
  { return lateral_distance / (radius + (front_distance/2)); }
  else
  { return lateral_distance / (radius - (front_distance/2)); } }
void steeringCallback(const
ackermann_msgs::AckermannDriveStamped::ConstPtr &msg)
{ double radius = radiusCalc(msg->drive.steering_angle);
  ROS_INFO("---Calculated radius : %.4f,", radius);
  if (msg->drive.steering_angle >= 0 && msg->drive.steering_angle <= 1.20)
  { fi_r.data = atan(formulaCalc(radius, false));
    fi_l.data = atan(formulaCalc(radius,true));
    ROS_INFO("---Calculated angles : left wheel %.4f, right wheel %.4f",
fi_r.data,fi_l.data);
    left_pub.publish(fi_r)
    right_pub.publish(fi_l); }
  if (msg->drive.steering_angle <= 0 && msg->drive.steering_angle >= -1.20)
  { ROS_INFO("---Calculated angles : right wheel %.4f, left wheel %.4f",
fi_r.data,fi_l.data);
    fi_r.data = atan(formulaCalc(radius,true));
    fi_l.data = atan(formulaCalc(radius, false));
  }
}

```

```

left_pub.publish(fi_l);
right_pub.publish(fi_r); }
if (msg->drive.steering_angle > 1.20)
{ double radius_exception(1.2);
ROS_INFO("---Calculated radius exception : %.4f,", radius);
fi_r.data = atan(formulaCalc(radius_exception, false));
fi_l.data = atan(formulaCalc(radius_exception,true));
ROS_INFO("---Calculated angles : left wheel %.4f, right wheel %.4f",
fi_r.data,fi_l.data);
left_pub.publish(fi_r);
right_pub.publish(fi_l); }
if (msg->drive.steering_angle < -1.2)
{ double radius_exception(-1.2);
ROS_INFO("---Calculated radius exception : %.4f,", radius);
fi_r.data = atan(formulaCalc(radius_exception, false));
fi_l.data = atan(formulaCalc(radius_exception,true));
ROS_INFO("---Calculated angles : left wheel %.4f, right wheel %.4f",
fi_r.data,fi_l.data);
left_pub.publish(fi_r);
right_pub.publish(fi_l); }
}
int main(int argc, char **argv)
{ ros::init(argc, argv, "steer_remap");
ros::NodeHandle n;
std::string lw_topic, rw_topic, central_steer_topic;
central_steer_topic = "/junior_car/ackermann_cmd";
lw_topic = "/junior_car/SteerLeft_controller/command";
rw_topic = "/junior_car/SteerRight_controller/command";
if (n.getParam("left_wheel_steer_topic", lw_topic))
{ ROS_INFO("Got param: %s", lw_topic.c_str());}
if (n.getParam("right_wheel_steer_topic", rw_topic))
{ ROS_INFO("Got param: %s", rw_topic.c_str()); }
if (n.getParam("right_wheel_steer_topic", central_steer_topic))
{ ROS_INFO("Got param: %s", central_steer_topic.c_str()); }
left_pub = std::move(n.advertise<std_msgs::Float64>(lw_topic, 1024));
right_pub = std::move(n.advertise<std_msgs::Float64>(rw_topic, 1024));
ros::Subscriber sub = n.subscribe(central_steer_topic, 1024, steeringCallback);
ros::spin();
return 0;
}

```

Додаток Д
(вибірковий)

Система навігації (СН) моделі МР «Junior»

Зміст файлу junior_hector.launch

```
<launch>
  <node pkg="move_base" type="move_base" respawn="false"
name="move_base" output="screen">
    <roscpp param file="$(find junior_2dnav)/config/dagny_nav.yaml"
command="load" />
    <roscpp param file="$(find junior_2dnav)/config/teb_local_planner.yaml"
command="load"/>
    <param name="SBPLLatticePlanner/primitive_filename" value="$(find
junior_2dnav)/primitives/dagny.mprim" />
    <roscpp param file="$(find junior_2dnav)/config/common_costmap.yaml"
command="load"
ns="local_costmap"/>
    <roscpp param file="$(find junior_2dnav)/config/local_costmap.yaml"
command="load"
ns="local_costmap"/>
    <roscpp param file="$(find junior_2dnav)/config/common_costmap.yaml"
command="load"
ns="global_costmap"/>
    <roscpp param file="$(find junior_2dnav)/config/global_costmap.yaml"
command="load"
ns="global_costmap"/>
    <remap from="cmd_vel" to="/steer_drive_controller/cmd_vel" />
    <remap from="odom" to="/steer_drive_controller/odom" />
  </node>
</launch>
```

Фрагмент файлу hector.launch

```
<launch>
  <arg name="tf_map_scanmatch_transform_frame_name"
default="scanmatcher_frame"/>
  <arg name="base_frame" default="footprint"/>
  <arg name="odom_frame" default="odom"/>
  <arg name="pub_map_odom_transform" default="false"/>
  <arg name="scan_subscriber_queue_size" default="5"/>
  <arg name="scan_topic" default="rectified_scan"/>
  <arg name="map_size" default="2048"/>
```

```

<node pkg="hector_mapping" type="hector_mapping"
name="hector_mapping" output="screen">
  <!-- Frame names -->
  <param name="map_frame" value="map" />
  <param name="base_frame" value="$(arg base_frame)" />
  <param name="odom_frame" value="$(arg odom_frame)" />
  <!-- Tf use -->
  <param name="use_tf_scan_transformation" value="true"/>
  <param name="use_tf_pose_start_estimate" value="false"/>
  <param name="pub_map_odom_transform" value="$(arg
pub_map_odom_transform)"/>

```

Зміст файлу junior_nav.launch

base_global_planner: SBPLLatticePlanner

recovery_behaviors:

```

[
  {
    name: conservative_reset,
    type: clear_costmap_recovery/ClearCostmapRecovery,
  },
]

```

planner_patience: 15.0

clearing_rotation_allowed: false

base_local_planner: teb_local_planner/TebLocalPlannerROS

controller_frequency: 10.0

SBPLLatticePlanner:

allocated_time: 15

initial_epsilon: 10.0

Фрагмент файлу teb_local_planner.yaml

TebLocalPlannerROS:

odom_topic: odom

map_frame: /map

Trajectory

teb_autosize: True

dt_ref: 0.6

dt_hysteresis: 0.1

global_plan_overwrite_orientation: True

max_global_plan_lookahead_dist: 3.0

feasibility_check_no_poses: 0

Robot

max_vel_x: 53

max_vel_x_backwards: -53

```

max_vel_theta: 53
acc_lim_x: 53
acc_lim_theta: 53
acc_lim_y: 53
min_turning_radius: 1.0
footprint_model:
  type: "polygon"
  vertices: [[-0.556, -0.325],[-0.556, 0.325],[0.556, 0.325],[0.556, -0.325]]
# GoalTolerance
xy_goal_tolerance: 0.2
yaw_goal_tolerance: 0.3
free_goal_vel: False
# Obstacles
min_obstacle_dist: 0.7
include_costmap_obstacles: True
costmap_obstacles_behind_robot_dist: 1.5
obstacle_poses_affected: 30
inflation_dist: 0.6
costmap_converter_plugin: ""
costmap_converter_spin_thread: True
costmap_converter_rate: 7 #was 5

```

Зміст файлу `common_costmap.yaml`

```

footprint: [[-0.556, -0.325],[-0.556, 0.325],[0.556, 0.325],[0.556, -0.325]]
footprint_padding: 0.0
transform_tolerance: 0.1
robot_base_frame: footprint
global_frame: map
update_frequency: 5.0
publish_frequency: 1.0
rolling_window: true
static:
  map_topic: map
  subscribe_to_updates: true
plugins:
  - { name: obstacles_laser, type: "costmap_2d::ObstacleLayer" }
  - { name: inflation, type: "costmap_2d::InflationLayer" }
resolution: 0.05
obstacles_laser:
  observation_sources: laser
  laser:
    {
      data_type: LaserScan,

```

```

clearing: true,
marking: true,
topic: rectificated_scan,
inf_is_valid: true,
}
inflation_layer:
inflation_radius: 0.7
enabled: true
footprint_layer:
enabled: true
sonar_layer:
enabled: true

```

Зміст файлу local_costmap.yaml

```

static_map: false
width: 6.0
height: 6.0
resolution: 0.05

```

Зміст файлу global_costmap.yaml

```

track_unknown_space: true
resolution: 0.10
update_frequency: 5.0
publish_frequency: 5.0

```

Зміст файлу rviz_config.launch

```

<?xml version="1.0"?>
<launch>
  <arg name="rviz" default="true" />
  <arg name="rviz_file" default="$(find junior_2dnav)/rviz/junior_rviz.rviz" />
  <node pkg="rviz" type="rviz" name="$(anon rviz)" args="-d $(arg rviz_file)"
output="screen" if="$(arg
rviz)"/>
</launch>

```

Фрагмент файлу junior_rviz.rviz

Panels:

- Class: rviz/Displays

Help Height: 78

Name: Displays

Property Tree Widget:

Expanded:

- /Global Options1

- /Status1
- /TF1/Frames1
- /Odometry1/Shape1
- Splitter Ratio: 0.5
- Tree Height: 775
- Class: rviz/Selection
- Name: Selection
- Class: rviz/Tool Properties
- Expanded:
- /2D Pose Estimate1
- /2D Nav Goal1
- /Publish Point1
- Name: Tool Properties
- Splitter Ratio: 0.588679016
- Class: rviz/Views

Додаток Е
(вибірковий)

Node коригування даних з лазерного далекоміра моделі МР «Junior»

```

#include "ros/ros.h"
#include "std_msgs/Float64.h"
#include <utility>
#include <cmath>
#include <sstream>
#include <string>
#include "sensor_msgs/LaserScan.h"
ros::Subscriber sub;
ros::Publisher pub;
const double laser_incline = 0.261799;
const double laser_height = 0.4;
const double max_obstacle = laser_height/tan(laser_incline);
void scanCallback(const sensor_msgs::LaserScan::ConstPtr& scan)
{
    sensor_msgs::LaserScan msg;
    msg.header.stamp = scan->header.stamp;
    msg.header.frame_id = "fake_laser";
    msg.angle_min = scan->angle_min;
    msg.angle_max = scan->angle_max;
    msg.angle_increment = scan->angle_increment;
    msg.time_increment = scan->time_increment;
    msg.range_min = scan->range_min;
    msg.range_max = scan->range_max;
    msg.ranges.resize(scan->ranges.size());
    msg.intensities.resize(scan->intensities.size());
    for (int i=0; i<scan->ranges.size() ; i++)
    {
        double r = scan->ranges[i] * cos(laser_incline);
        if (r > max_obstacle)
            r = std::numeric_limits<double>::infinity();
        msg.ranges[i] = r;
        msg.intensities[i] = scan->intensities[i];
    }
    pub.publish(msg);
}
int main(int argc, char **argv)
{
    ros::init(argc, argv, "rectificated_laser");

```

```

ros::NodeHandle n;
std::string scan_topic, remap_topic;
remap_topic = "/rectificated_scan";
scan_topic="/scan";
pub = n.advertise<sensor_msgs::LaserScan>(remap_topic, 50);
sub = n.subscribe(scan_topic, 1, scanCallback);
ros::spin();
return 0;
}

```

Опис моделі МР «Junior»

E.1 Фрагмент файлу junior_car.urdf.xacro

```

<?xml version="1.0"?>
<robot name="junior_car"
  xmlns:xacro="http://www.ros.org/wiki/xacro">
  <xacro:include filename="$(find junior_car)/urdf/junior_car.gazebo.xacro" />
  <!-- inertial macros -->
  <xacro:include filename="$(find junior_car)/urdf/inertia.xacro" />
  <!-- add transmissions -->
  <xacro:include
    junior_car)/urdf/junior_car.transmission.xacro" />
    filename="$(find
  <link name="footprint">
  </link>
  <link name="base_link">
  <visual>
  <origin xyz="0 -0.02 0.165" rpy="0 0 0" />
  <geometry>
  <mesh filename="package://junior_car/meshes/base_link.STL" />
  </geometry>
  <material name="">
  <!--<color rgba="0.25098 0.25098 0.25098 1" />-->
  </material>
  </visual>
  <collision>
  <origin xyz="0 -0.02 0.165" rpy="0 0 0" />
  <geometry>
  <mesh filename="package://junior_car/meshes/base_link.STL"/>
  </geometry>
  </collision>
  <xacro:box_inertial_origin x="${b_height}" y="${b_width}" z="${b_depth}"
mass="${b_mass}"
  origin="0 -0.025 0.2" rpy="0 0 1.5708"/>
  </link>

```

```

<joint name="footprint_joint" type="fixed">
  <parent link="footprint"/>
  <child link="base_link"/>
  <origin xyz="0 0 0" rpy="0 0 -1.5709" />
</joint>
<link name="hokuyo_sensor">
  <xacro:box_inertial      x="{hokuyo_depth}"      y="{hokuyo_depth}"
z="{hokuyo_width}"
  mass="{hokuyo_mass}"/>
  <visual>
  <origin xyz="0 0 0" rpy="0 -${pi/2} ${pi}" />
  <geometry>
  <mesh filename="package://junior_car/meshes/hokuyo_sensor.STL" />
  </geometry>
  <material name="">
  <color rgba="0.250980392156863 0.250980392156863 0.250980392156863 1"
/>
</material>
</visual>
<collision>
  <origin xyz="0 0 0" rpy="0 -${pi/2} ${pi}" />
  <geometry>
  <mesh filename="package://junior_car/meshes/hokuyo_sensor.STL" />
  </geometry>
</collision>
</link>

```

E.2 Фрагмент файла junior_car.transmission.xacro

```

<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
  <xacro:macro name="junior_transmission">
    <transmission name="junior_car_tran_1">
      <robotNamespace>/junior_car</robotNamespace>
      <type>transmission_interface/SimpleTransmission</type>
      <joint name="right_steer_joint">
        <hardwareInterface>PositionJointInterface</hardwareInterface>
      </joint>
      <actuator name="junior_car_motor_1">
        <hardwareInterface>PositionJointInterface</hardwareInterface>
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
    </transmission>
  </macro>

```

E.3 Фрагмент файла junior_car.gazebo.xacro

```

<?xml version="1.0"?>

```

```

<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
  <gazebo>
    <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
    </robotNamespace></robotNamespace>

```

```

<robotSimType>steer_bot_hardware_gazebo/SteerBotHardwareGazebo</robotSimType>

```

```

  </plugin>
  </gazebo>
  <gazebo reference="imu_link">
    <material>Gazebo/Yellow</material>
    <gravity>true</gravity>
    <sensor name="imu_sensor" type="imu">
      <always_on>true</always_on>
      <update_rate>100</update_rate>
      <visualize>true</visualize>
      <topic>/avrora_imu</topic>
      <plugin filename="libgazebo_ros_imu_sensor.so" name="imu_plugin">
      <topicName>imu</topicName>
      <bodyName>imu_link</bodyName>
      <updateRateHZ>10.0</updateRateHZ>
      <gaussianNoise>0.0</gaussianNoise>
      <xyzOffset>0 0 0</xyzOffset>
      <rpyOffset>0 0 0</rpyOffset>
      <frameName>imu_link</frameName>
    </plugin>
    <pose>0 0 0 0 0 0</pose>
  </sensor>
</gazebo>

```

E.4 Фрагмент файла inertia.xacro

```

<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
  <property name="M_PI" value="3.1415926535897931" />
  <xacro:property name="hokuyo_mass" value="0.16"/>
  <xacro:property name="hokuyo_height" value="0.05"/>
  <xacro:property name="hokuyo_depth" value="0.05"/>
  <xacro:property name="hokuyo_width" value="0.07"/>
  <xacro:macro name="sphere_inertial" params="radius mass">
    <inertial>
      <mass value="{mass}" />
      <origin xyz="0 0 0" />
      <inertia ixx="{0.4 * mass * radius * radius}" ixy="0.0" ixz="0.0" iyy="{0.4
* mass * radius *

```

```

radius}" iyz="0.0" izz="{0.4 * mass * radius * radius}" />
</inertial>
</xacro:macro>
<xacro:macro name="cylinder_inertial" params="radius length mass origin
rpy">
<inertial>
<mass value="{mass}" />
<origin xyz="{origin}" rpy="{rpy}" />
<inertia ixx="{0.0833333 * mass * (3 * radius * radius + length * length)}"
ixy="0.0" ixz="0.0"
iyy="{0.0833333 * mass * (3 * radius * radius + length * length)}" iyz="0.0"
izz="{0.5 * mass * radius *
radius}" />
</inertial>
</xacro:macro>

```

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Спорий Олександр Володимирович

Тема: Метод керування автономним рухом мобільного робота

Спеціальність: 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Обсяг кваліфікаційної роботи:

Кількість сторінок записки 77

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розв'язання задачі останнього кілометра на основі мобільного робота (MR), що включає створення реалістичної моделі MR, і подальша апробація моделі MR для вирішення завдання доставки

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі описано проблематику переміщення вантажу із використанням безпілотних транспортних засобів. У другому розділі виконано моделювання системи керування переміщенням вантажу із використанням безпілотних транспортних засобів, проаналізовано програмне забезпечення для керування автономним рухом мобільного робота. У третьому розділі виконано моделювання роботи мобільного робота для переміщення виробів. У четвертому розділі наведено дослідження системи навігації для мобільного робота «JUNIOR».....

4. Позитивні сторони роботи: висока практична цінність роботи, досліджено систему навігації в програмному середовищі ROS, виконано коригування отриманих даних з лазера моделі мобільного робота, описано налаштування локального планувальника та карт вартості планувальника та представлено апробацію системи навігації в Rviz

5. Негативні сторони роботи: у роботі недостатньо уваги приділяється аналізу існуючих технічних рішень

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації

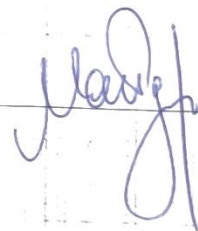
7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: відмінно (5,0/А)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Майдан Павло Сергійович, канд. техн. наук, доцент кафедри МАЕЕС

"12" 12 2024 р.

 (підпис)

Завідувачу кафедри АКІТтаР
д-ру техн.наук. проф. Мартинюку В.В.

Спорого О.В.

ПІБ здобувача вищої освіти

ФІТ, 2 курс, групи АКІТРМ-23-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

9.12.2024

дата



підпис

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Олександр СПОРИЙ

Співавтор:

Назва: МКР Спорий

Науковий керівник: Денис МАКАРИШКІН

Підрозділ: Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Коефіцієнт подібності 1: 1.3%

Коефіцієнт подібності 2: 0%

Мікропробіли: 0

Заміна букв: 3

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2024-12-10 06:03:16.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

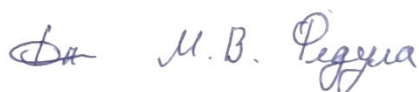
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2024-12-10

Дата



експерт

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилоч в документах: 10%**

| | | | | |
|---|----------|---------|-----------------------------|---------|
| ID: 157131 Назва: МКР Метод керування автономним рухом безпілотною транспортною засобу Додано в БД: 2024-12-10 Автора: Олександр СПОРИЙ Керівники: Денис МАКАРИШКІН Консультанти: Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 105123 | 1539 | 2996 (3%) | 40 (3%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |
| | | | |

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ АВТОМАТИЗАЦІЇ, КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ТА
РОБОТОТЕХНІКИ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод керування автономним рухом мобільного робота

Автор: Спорий Олександр Володимирович

Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітня програма: Освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології»

Науковий керівник: Макаришкін Денис Анатолійович, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|--|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розмішені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розмішені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |
| 5 | Інше: | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 1,3% і адресується до 17 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи





Валерій МАРТИНЮК

Валерій МАРТИНЮК

Денис МАКАРИШКІН