

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРҒЫ БІЛІМ МИНИСТРЛІГІ
«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық
емес акционерлік қоғамы



Автоматика және ақпараттық технологиялар институты
«Робототехника және автоматиканың техникалық құралдары» кафедрасы

Игибай Данияр Галымбекулы

«Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі»

МАГИСТРЛІК ДИССЕРТАЦИЯ

7М07107 – Робототехника және мехатроника

Алматы 2025

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРҒЫ БІЛІМ МИНИСТРЛІГІ

«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық
емес акционерлік қоғамы



SATBAYEV
UNIVERSITY

Автоматика және ақпараттық технологиялар институты

ӘОЖ 665.622.43.046.6-52 (043)

Қолжазба құқығында

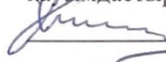
Магистр академиялық дәрежесін алу үшін
МАГИСТРЛІК ДИССЕРТАЦИЯ (ЖОБА)

Диссертацияның атауы: Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі
Дайындау бағыты: 7М07107 – Робототехника және мехатроника

Ғылыми жетекші

ф-м.ғ.к.,

қауымдастырылған профессор

 Бактыбаев М.К

«11» 06 2025 ж.

Рецензент

Плазма физикасы, нанотехнология

және компьютерлік физика кафедрасы

ф-м.ғ.к., қауымдастырылған

профессор



Муратов М.М

«11» 06 2025 ж.

Норма бақылаушы

Тех. ғылым. маг., оқытушы,

 Кальменов Е.Т

«09» 06 2025 ж.

ҚОРҒАУҒА ЖІБЕРІЛДІ

Кафедра меңгерушісі РТЖАТК

техника ғылымының кандидаты,

профессор



Қ.Ә. Өжікенов

«11» 06 2025 ж.

Алматы 2025

«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық
емес акционерлік қоғамы



SATBAYEV
UNIVERSITY

Автоматика және ақпараттық технологиялар институты

«Робототехника және автоматиканың техникалық құралдары» кафедрасы

7M07107 – Робототехника және мехатроника



Кафедра меңгерушісі РТЖАТҚ
техника ғылымының кандидаты
Қ.Ә. Өжікенов
«11» 06 2025 ж.

Магистрлік диссертацияны орындауға арналған ТАПСЫРМА

Магистрант Игибай Данияр Галымбекулы

Тақырыбы: Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі

Университет ректорының 2022 ж. « 25 » 03 № 430 - М бұйрығымен бекітілген

Аяқталған жұмысты тапсыру мерзімі: «02» маусым 2025 ж.

Магистрлік диссертацияның бастапқы деректері:

Магистрлік диссертацияда қарастырылатын мәселелер тізімі:

- а) *Жер сілкінісі жағдайларын модельдеу*
- б) *Көп агентті жүйе архитектурасын жобалау*
- в) *ROS2, CoppeliaSim және Gazebo орталарында жүйені модельдеу*
- г) *Жүйенің тиімділігін бағалау*

Графикалық материалдың тізімі (міндетті сызбаларды дәл көрсете отырып):

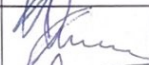
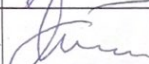
жұмыс презентациясы слайтарда 23 көрсетілген

Ұсынылатын негізгі әдебиеттер: 23 әдебиеттер тізімі және 1 қосымша

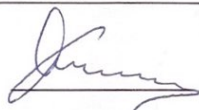
Магистрлік диссертацияны дайындау
КЕСТЕСІ

Бөлімдер атауы, әзірленетін сұрақтар тізбесі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Жер сілкінісі жағдайларын модельдеу	15.01-13.02.2025 ж	Орындалды
Көп агентті жүйе архитектурасын жобалау	13.02-21.03.2025 ж.	Орындалды
ROS2, CoppeliaSim және Gazebo орталарында жүйені модельдеу	21.03-19.04.2025 ж.	Орындалды
Жүйенің тиімділігін бағалау	19.04-15.05.2025 ж.	Орындалды
Қорытынды бөлім	16.05-19.05.2025 ж	Орындалды

Аяқталған магистрлік диссертация үшін, оған қытысты бөлімдердегі диссертациялар кеңесшілері мен қалып бақылаушының қолдары

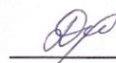
Бөлімдердің атауы	Кеңесшілер, тегі, аты, әкесінің аты, (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қол
Қалып бақылаушы	Кальменов Е.Т., техника ғылымдарының магистрі, оқытушы	09.06.2025	
Негізгі бөлім	Бактыбаев М.К., ф-м.ғ.к., қауымдастырылған профессор	10.06.2025	
Есептеу бөлім	Бактыбаев М.К., ф-м.ғ.к., қауымдастырылған профессор	10.06.2025	

Ғылыми жетекшісі



Бактыбаев М.К

Білім алушы тапсырманы орындауға алды



Игибай Д.Г

Күні

« 11 » 06 2025 ж.

АҢДАТПА

Бұл магистрлік диссертацияда жер сілкінісі сияқты табиғи апаттардан кейінгі төтенше жағдайлар кезінде қолдануға арналған көп агентті роботтық жүйені әзірлеу және модельдеу мәселелері қарастырылады. Жүйенің басты мақсаты — зардап шеккен аймақтарда құтқару операцияларын жылдам әрі қауіпсіз жүзеге асыру, құтқарушылардың өмірін қатерге тікпей, роботтар арқылы іздестіру және алғашқы көмек жеткізу жұмыстарын автоматтандыру.

Жобада бірнеше автономды мобильді агенттердің үйлесімді жұмыс істеуі, бір-бірімен байланыс орнатуы, тапсырмаларды бөлісуі және кедергілерді айналып өтуі қарастырылады. ROS2 және Gazebo платформалары қолданылып, әр агенттің автономды қозғалысы, SLAM технологиясы арқылы карта құру және навигациялық алгоритмдер енгізілді. Сонымен қатар, апатты ортада сенсорлардан (LiDAR, камера, т.б.) алынған ақпарат негізінде әрекет ету қабілеті тестіленді.

Симуляция нәтижелері арқылы көп агентті жүйенің тиімділігі, тапсырмаларды орындау жылдамдығы, навигациялық дәлдігі және апатты аймақты қамту ауқымы бағаланды. Жұмыс нәтижелері болашақта нақты құтқару операцияларына енгізу үшін ғылыми-тәжірибелік негіз бола алады.

АННОТАЦИЯ

Данная магистерская диссертация посвящена разработке и моделированию многоагентной робототехнической системы, предназначенной для проведения поисково-спасательных операций в зонах, пострадавших от землетрясений. Целью проекта является создание эффективной, автономной и устойчивой системы, способной выполнять задачи поиска пострадавших, картографирования разрушенной местности и доставки первой помощи без риска для жизни спасателей.

Разрабатываемая система состоит из нескольких мобильных роботов (агентов), взаимодействующих друг с другом и координирующих свои действия в реальном времени. Используются современные технологии ROS2 и Gazebo, в которых моделируются автономная навигация, построение карты при помощи SLAM, обмен данными между агентами, распределение задач и избегание препятствий. Каждый агент оснащён сенсорной системой, включающей LiDAR, камеру и модули связи, позволяющие ориентироваться в сложных условиях.

В рамках исследования проведены симуляции различных сценариев спасения: от поиска жертв в замкнутых пространствах до координации агентов при развертывании помощи. Эффективность системы была проанализирована по таким метрикам, как скорость выполнения задач, точность построения карты, охват зоны бедствия и устойчивость системы к отказам отдельных компонентов. Результаты исследования демонстрируют потенциал внедрения многоагентных систем в реальные спасательные операции и открывают возможности для дальнейших научных исследований и разработок в области робототехники.

ABSTRACT

This master's thesis focuses on the design, development, and simulation of a Multi-agent Robotic System aimed at performing search and rescue operations in post-earthquake disaster environments. The primary goal of the system is to autonomously assist in locating survivors, mapping collapsed areas, and delivering aid supplies without endangering human rescuers.

The system architecture is based on multiple mobile robots (agents) that collaborate through wireless communication, task sharing, and real-time coordination. The robots are equipped with sensors such as LiDAR, cameras, and IMUs, enabling them to navigate challenging terrain and operate independently. The framework is implemented in ROS2 and simulated in Gazebo, leveraging SLAM (Simultaneous Localization and Mapping) for environmental mapping and Nav2 for autonomous navigation and path planning.

A variety of disaster response scenarios are simulated, including building debris navigation, narrow passage exploration, and distributed area coverage. The system demonstrates robustness through redundancy and fault-tolerance, where agents can dynamically adapt their roles if one or more units fail. Performance is measured by mission time, area coverage, mapping accuracy, and inter-agent communication efficiency. This work contributes to the advancement of intelligent rescue robotics and opens future directions for integrating swarm behaviors and real-world field deployment in emergency response applications.

ҚЫСҚАРТЫЛҒАН ТЕРМИНДЕР ТІЗІМІ

Қысқартылған сөздер	Толық атауы	Түпнұсқа
ROS	Робот операциялық жүйесі	Robot Operating System
ROS2	Робот операциялық жүйесі 2-нұсқасы	Robot Operating System 2
SLAM	Бір уақытта локализация және карта жасау	Simultaneous Localization and Mapping
LiDAR	Лазерлік қашықтық өлшеу және нысан тану жүйесі	Light Detection and Ranging
GPS	Жаһандық позициялау жүйесі	Global Positioning System
ҰҰА	Ұшқышсыз ұшу аппараты	Unmanned Aerial Vehicle
ҰЖК	Ұшқышсыз жерүсті көлігі	Unmanned Ground Vehicle
SLAR	Қабатты іздеу және құтқару алгоритмі	Layered Search and Rescue Algorithm
RViz	ROS визуализация құралы	ROS Visualization
Nav2	Навигация жүйесі 2-нұсқасы	Navigation Stack 2
CRASAR	Робот көмегімен іздеу-құтқару орталығы	Center for Robot-Assisted Search and Rescue
DMCsim	Машиналық оқыту негізіндегі командалық модель симуляциясы	Decision-Making Command simulation
TF	Координат жүйесі трансформациясы	Transform Frames
RTK-GPS	Дәл нақты координаттарды анықтауға арналған GPS	Real-Time Kinematic GPS
ҰСК	Ұшқышсыз суасты көлігі	Unmanned Underwater Vehicle
ЖИ	Жасанды интеллект	Artificial Intelligence
GUI	Графикалық интерфейс	Graphical User Interface

МАЗМҰНЫ

КІРІСПЕ	10
1 Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі	11
1.1 Әдебиеттік шолу және тақырыптың теориялық негіздері	11
1.2 Қазіргі заманғы көп агентті жүйелердің талдауы	14
1.3 Жер сілкінісі кезіндегі құтқару операцияларының ерекшеліктері	18
1.4 Қазақстандағы төтенше жағдайлар жүйесі: цифрлық жаңғырту	19
1.5 Робототехникалық жүйелердің құтқару операцияларында қолданылуы: халықаралық тәжірибелердің талдауы	20
1.6 Мақсаттарды іздеу мен анықтаудың қолданыстағы әдістері мен алгоритмдері	21
2 Зерттелетін жүйенің әдіснамасы және моделі	26
2.1 Роботтық жүйенің негізгі деңгейлері	26
2.2 Технологияға шолу. ROS2, RViz және Gazebo сипаттамасы	28
2.3 Дифференциалды роботтың жұмыс принциптері	31
2.4 Навигация үшін LiDAR және SLAM пайдалану	33
2.5 Бір роботтың іздеу тиімділігін есептеу	36
3. Жүйелік дизайн	37
3.1 Жүйе архитектурасы	37
3.2 Роботтың дизайны	39
4. Жүйені енгізу	42
4.1 SLAM орнату және іске қосу	42
4.2 Офлайн шарлау үшін Nav2-мен интеграция	45
4.3 Басқару коды және деректерді ұйымдастыру	49
4.4 Тестілеу ортасын ұйымдастыру	52
Пайдаланылған әдебиеттер тізімі	56
А Қосымшасы	58

КІРІСПЕ

Жер сілкіністері бұл қазіргі заманғы әлеуметтік-экономикалық дамуға зор қауіп төндіретін табиғи апаттардың ең қауіпті түрлерінің бірі. Әсіресе, қалалық инфрақұрылымдардың жиі қирауы, адамдардың көпшілігінің қиралған ғимараттар астында қалуы және құтқару жұмыстарын жүргізудегі қиындықтар осы мәселенің өзектілігін арттырады. Дәстүрлі құтқару әдістерінің шектеулілігіне байланысты, көп агентті роботтық жүйелерді енгізу бұл апат аймақтарында тиімді және қауіпсіз операцияларды жүргізудің инновациялық әдісі.

Бұл зерттеуде ROS2 (Robot Operating System 2), Gazebo симуляторы және автономды навигациялық жүйелер (SLAM, Nav2) негізінде құрастырылған көп агентті роботтық жүйенің жер сілкінісі кезіндегі құтқару операцияларына қолданылуы қарастырылады. Жобаның негізгі мақсаты – апат аймақтарын тез арада сканерлеу, қиралған объектілердің картасын жасау және құтқарушыларға нақты уақытта ақпарат беру арқылы операциялардың тиімділігін арттыру.

Зерттеудің өзектілігі:

- а) бірнеше роботтың бір ортада өзара кедергісіз және үйлесімді жұмыс істеуін қамтамасыз ету;
- б) қиын әрі белгісіз ортада навигация мен карта құруды жүзеге асыру;
- в) агенттер арасында тапсырмаларды автоматты түрде бөлісу мен басқару әдістерін жетілдіру.

Жобаның зерттеу нысаны жер сілкінісінен кейінгі апатты ортада жұмыс істейтін көп агентті автономды роботтық жүйе.

Жүйенің зерттеудің мақсаты - жер сілкінісінен кейінгі апат аймақтарында қолдануға арналған автономды көп агентті роботтық жүйенің моделін жасау, симуляциялық ортада сынау және оның тиімділігін бағалау.

1 Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі

1.1 Әдебиеттік шолу және тақырыптың теориялық негіздері

Жер сілкінісі табиғи апаттардың ең қауіпті және көп құрбан әкелетін түрлерінің бірі болып табылады. Оны үш негізгі кезеңге бөлуге болады: жер сілкінісіне дейінгі кезең, жер сілкінісі сәті және жер сілкінісінен кейінгі кезең. Әр кезеңде әрекет ету тәсілдері әртүрлі, ал көп агентті роботтық жүйелердің рөлі негізінен апаттан кейінгі кезеңде анық байқалады.

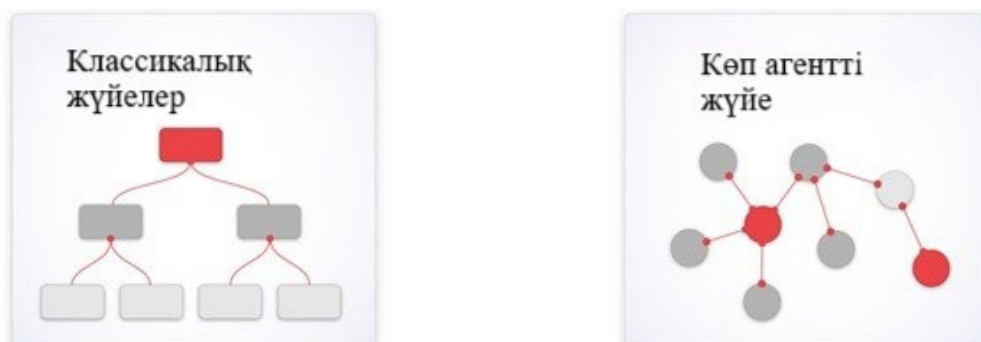
Жер сілкінісіне дейінгі кезең бұл кезеңде басты назар халықтың ақпараттылығы мен төтенше жағдайларға дайындығына бағытталады. Төтенше жағдайлар жөніндегі халықаралық сарапшылардың пікірінше, адамдардың жер сілкінісіне психологиялық және физикалық дайындығы апат салдарын 30-40%-ға дейін төмендетуі мүмкін. Негізгі шаралар:

- a) Қауіпсіз орындарды білу (мысалы, ғимараттағы есік жақтаулары немесе берік үстел астына тығылу);
- b) Төтенше жағдай кезінде қолданылатын әрекеттер алгоритмін жаттау;
- c) Алдын ала дайындалған жеке қауіпсіздік жинағын (су, фонарь, медициналық заттар) сақтау;
- d) Эвакуация маршрутын үйрену.

Жер сілкінісі сәті бұл кезең өте қысқа және қауіпті. Мұнда адамның әрекеті автоматты түрде және нақты болуы тиіс. Қауіпсіздік бойынша халықаралық нұсқаулықтар жер сілкінісі кезінде қолдануға ұсынылған негізгі модель.

Мысалы, АҚШ-тың FEMA агенттігі 2021 жылы шығарған нұсқаулықта: «Жер сілкінісі кезінде ең тиімді әдіс құлап қалудан сақтанып, мықты құрылым астына тығылу және сілкініс тоқтағанша қозғалмау».

Жер сілкінісінен кейінгі кезең - бұл кезең жер сілкінісі тоқтағаннан кейін басталады және ең маңызды, себебі тірі қалғандарды құтқару мүмкіндігі алғашқы 72 сағат ішінде ең жоғары. Осы уақыт аралығында киранды ғимараттардың ішінде тірі қалғандар болуы мүмкін, кайталама дүмпулер қосымша қауіп төндіреді, құтқарушыларға апат аймағына жету қиындауы ықтимал.[2]



1.1-сурет - Жүйелер

Дәл осы кезеңде көп агентті роботтар ерекше маңызды. Олар адамдар кіре алмайтын, қауіпті аймақтарды сканерлеуге, тірі адамдарды табуға, карта жасауға және көмек көрсетуге бейімделген. Мысал ретінде, Fukushima-2011 кезінде радиация деңгейі жоғары ортада PackBot және Quince роботтары қолданылып, ғимарат ішіндегі жағдайды бақылауға мүмкіндік берді. 2001 жылғы 11 қыркүйек оқиғасынан кейін Нью-Йоркте CRASAR орталығы роботтарды ғимарат қирандыларында іздеу және құжаттау үшін пайдаланды.[3]

Қазіргі уақытта роботты агенттер мен көп агентті робот жүйелері адамдарға қауіп төндіретін салаларда кеңінен қолданылады, мысалы, химиялық және ядролық өнеркәсіпте техногендік және табиғи апаттар кезінде және т.б. 1.1-суретке сәйкес, көп агентті роботтық жүйелерді қолданудың артықшылықтары айқын, өйткені әрекет ауқымы артады, орындалатын тапсырмалар жиынтығы кеңейеді және тапсырманы сәтті аяқтау ықтималдығы артады, бұл агенттер арасында өзгерістер болған жағдайда тапсырмаларды қайта бөлу арқылы қол жеткізіледі. мульти-агенттік жүйеге енгізілген роботтардың саны. Жер сілкінісі адамдарға қауіп төндіретін ең ауыр апаттардың бірі. Сонымен қатар, жер сілкінісі көбінесе ғимараттардың қирауы, өрттер мен су тасқыны сияқты екінші реттік апаттарды тудырады, бұл қосымша шығындарға әкеледі. Қазір жүргізіліп жатқан құтқару жұмыстары онша тиімді емес. Осыған байланысты өзара әрекеттесетін роботты агенттердің командаларын пайдалану қажет. Осылайша, диссертация тақырыбы өзекті болып табылады. Бұл мәселені шешу қолайсыз немесе қауіпті жағдайларда жұмыс істегенде адам өмірі мен денсаулығына қауіп-қатерді айтарлықтай төмендететін қолдануды іс жүзінде жүзеге асыруға бағытталған.

Роботтандырылған жер сілкінісінен құтқару жүйесін жобалау, талдау және енгізуде бірқатар қиындықтар туындайды. Біріншіден, авариялық-құтқару тобын құру және құтқару процесінің кезеңдерін жоспарлау кезінде төтенше жағдайлардың ерекшеліктерін ескеру қажет. Екіншіден, іздеу мәселелерін талдау және шешу кезінде сенсорлардың және жұмыс аймағының қасиеттерін ескеру қажет. Үшіншіден, құтқару тапсырмаларын орындау процесінде агенттер арасында мақсаттарды оңтайлы бөлу мәселесін шешу қажет, оның ішінде жүйеге енгізілген роботтардың санын өзгерту кезінде. Соңында, көп агентті роботтық жүйені басқаруды жүзеге асыру агенттерден сымсыз желіні пайдаланып бірлесе әрекет етуді талап етеді.

Статистикаға сүйенсек, әлемде жыл сайын 5 миллион жер сілкінісі тіркеледі, бірақ олардың көпшілігі мұхиттарда немесе жер қыртысының тереңдігінде болады немесе олардың қарқындылығы тым төмен және адам сезбейді. Дегенмен, үлкен шығындар мен материалдық шығындарға әкелетін күшті жер сілкінісі де орын алады. Тек өткен ғасырдың аяғынан бері жер сілкінісі мен оның зардаптары салдарынан әлемде 1,2 миллионнан астам адам қаза тапты. Осыған байланысты авариялық-құтқару жұмыстарын ұйымдастыру мен өрістетуге үлкен маңызға ие. Мұндай операциялардың негізгі

мақсаттары - зардап шеккендердің орнын тез арада анықтау және өлім санын азайту және материалдық шығынды азайту үшін оларға ықтимал көмек көрсету.

Жер сілкінісі кенеттен болатыны мен болжауға болмайтындығына байланысты ең қауіпті табиғи апаттардың бірі болып табылады. Сейсмикалық қауіпті аймаққа жататын елдер мен аудандар Тынық мұхит, Азия-Еуропа және Үнді тектоникалық қабаттарының түйіскен жеріндегі Тынық және Үнді мұхиттарының аймақтары болып табылады. Жапония, Қытай, Индонезия, Таиланд, Филиппин, Бангладеш және басқа да Азия елдеріне үнемі қауіп төніп тұр. Сонымен қатар, АҚШ-тың батысы да сейсмикалық аймақта орналасқан. Болған бірнеше жер сілкіністерін және олардың салдарын қарастырайық.

Диссертацияда құтқару кезеңдерін жоспарлау әдістемесін әзірлеу, авариялық-құтқару тобын құру, сондай-ақ кедергілер бар ортада іздеуді жүзеге асыратын, тапсырмаларды тарататын және траекторияларды жоспарлайтын роботоагенттерді басқару модульдерін құрумен байланысты мәселелер қарастырылады. Бұл мәселелерді шешу құтқару жұмыстарын тиімді жүргізуге мүмкіндік береді.

Жер сілкінісінен құтқарудың көп агентті роботтық жүйелері бұл апаттық жағдайларда құтқару және көмек көрсету жұмыстарының тиімділігін айтарлықтай арттыра алатын инновациялық технологиялар. Міне, осындай жүйелердің кейбір негізгі аспектілері:

Автономды роботтар бұл жүйелерге жер сілкінісінен кейін қауіпті және қол жетпейтін жерлерге кіруге қабілетті әртүрлі автономды роботтар кіреді. Олар дрондар, жорғалағыштар, роботты су астындағы көліктер және т.б. Бұл роботтар тірі қалғандарды іздеу және құрылымдар мен қауіпті аймақтарды бағалау үшін сенсорлармен және камералармен жабдықталған. Байланыс және байланыс жүйелері роботтар мен құтқару топтары арасындағы тиімді үйлестіру сенімді байланыс жүйелерін қажет етеді. Бұл сымсыз желілерді, спутниктік байланыстарды және т.б. Мұндай жүйелер операторларға роботтардың әрекеттерін басқаруға және нақты уақытта кері байланыс алуға мүмкіндік береді.[4]

Басқару және жоспарлау алгоритмдері тиімді басқару және жоспарлау алгоритмдерін әзірлеу көп агентті роботты құтқару жүйелерінде маңызды рөл атқарады. Бұл алгоритмдер роботтарға өз әрекеттерін үйлестіруге, қақтығыстарды болдырмауға және ресурстарды барынша тиімді пайдалануға мүмкіндік береді.

Географиялық ақпараттық жүйелер пайдалану географиялық деректерді талдауға және авариялық-құтқару жұмыстарының басым бағыттарын анықтауға көмектесетін қауіп пен осалдық карталарын жасауға мүмкіндік береді.

Деректер туралы ғылым және жасанды интеллект: роботтардан және басқа көздерден деректерді жинау, талдау және интерпретациялау нақты уақытта шешім қабылдауда маңызды рөл атқарады. Машиналық оқытуды және жасанды интеллект әдістерін қолдану шешім қабылдаудың дәлдігі мен

жылдамдығын арттырады. Көп агентті құтқару роботтарының жүйелері адамдардың өзара әрекеттесуін ескере отырып жасалғаны маңызды. Бұған операторлар үшін интуитивті басқару интерфейстері, сондай-ақ адамдармен бірлескен командаларда жұмыс істеу мүмкіндігі кіреді.

Тұтастай алғанда, жер сілкінісінен құтқарудың көп агентті роботтық жүйелерін әзірлеу және пайдалану қиын жағдайларда құтқару жұмыстарының тиімділігі мен қауіпсіздігін арттыру үшін маңызды күш болып табылады.

1.2 Қазіргі заманғы көп агентті жүйелердің талдауы

Іздестіру құтқару және жою жұмыстарын жүргізу құтқарушылар үшін күрделі де қауіпті міндет болып табылады. Тиісінше, робот-агенттерді пайдалану өзекті болып табылады. Қазіргі уақытта іздестіру-құтқару тапсырмаларын орындай алатын әртүрлі робот-агенттер әзірленді.

Құтқару роботтарының төрт түрі бар: ұшқышсыз ұшатын аппараттар (ұшқышсыз ұшу аппараттары), ұшқышсыз жерүсті көліктері, ұшқышсыз жерүсті көліктері және ұшқышсыз теңіз немесе су астындағы көліктер. Роботтардың осы түрлерін толығырақ қарастырайық. Төмендегі 1.2-суретте Laikago роботы қытайлық Unitree Robotics компаниясы жасаған төртаяқты робот. Ол электр жетектерімен және қозғалысты басқарудың жетілдірілген жүйесімен жабдықталған, ол тегіс емес беттерде қозғалуға, тепе-теңдікті сақтауға және әртүрлі жағдайларға бейімделуге мүмкіндік береді. Лаикагоның дизайны жануарлардың биомеханикасынан шабыттандырылған және оның денесі жеңіл, берік материалдардан жасалған, бұл оны мобильді және тұрақты етеді. Ол өрескел жерлерде жүре алатын болса да, оның тұрақтылығы ауыр жол талғамайтын, ылғалды немесе борпылдақ беттерде (күм, қар) төмендеуі мүмкін.[3]



1.2-сурет – Құтқару Laikago роботы -

Келесі 1.3-суретте электр жетегі және алға қарай бағытталған камерасы бар планер типіндегі ұшқышсыз ұшу аппараты көрсетілген, болжам бойынша әскери немесе барлау мақсаттары бар. Бұл дрондарды армия бөлімшелері барлау, атысты реттеу, жаудың қозғалысын бақылау және соққылардың салдарын бағалау үшін пайдаланады. Дизайнының арқасында (қанаттарының кеңдігі, салмағы аз және аэродинамикалық) мұндай дрондар ұзақ қашықтыққа байқалмай ұшып, ауада ұзақ уақыт бола алады. Олар көбінесе қолмен іске қосылады, ал басқару және деректерді жинау ноутбук немесе портативті станция арқылы жүзеге асырылады. Мұндай ұшқышсыз ұшу аппараттарының кемшіліктері арасында олардың ауа-райына тәуелділігін атап өткен жөн: қатты жел, жаңбыр немесе тұман олардың ұшуын бұзуы немесе құрылғыны зақымдауы мүмкін. Сондай-ақ, қуат көзі шектеулі болғандықтан (батареялар жиі пайдаланылады) ұшу ауқымы мен жұмыс уақыты шектеулі. Қазіргі уақытта тікұшақтар мен ұшқышсыз ұшақтар, мысалы, квадрокоптер немесе мультикоптер, ғарыштық тұрақтандыру мүмкіндіктері бар, қатты желге төтеп бере алатын және базалық станциядан бірнеше шақырым қашықтықта ұшатын ұшақтар ретінде пайдаланылады. ұшқышсыз ұшу аппараттары құтқару жұмыстарында өте пайдалы. [1]



1.3-сурет – Әскери жағдайдағы дрондар жүйесі

Оларды жоғарыдан апат аймақтарын тексеру, түсіру және қоршаған ортадағы өзгерістерді қалпына келтіру үшін пайдалануға болады. Олардың позициясының тік құрамдас бөлігі зардап шеккендерді іздеу және құтқару кезінде де, азық-түлік пен дәрі-дәрмекті тасымалдау кезінде де, сондай-ақ апат аймағынан/қайта ақпарат беру кезінде де пайдалы болатын үлкен шолуды қамтамасыз етеді. Тағы бір жүйе 1.4-суреттегі орман өртін бақылау үшін пайдаланылатын көп роторлы дрон (квадрокоптер) айып кетуге болады. Мұндай дрондар азаматтық қауіпсіздік қызметтерінде, қоршаған ортаны

бақылау орталықтарында және өртке қарсы қызметтерде өрт ауқымын жылдам бағалау, өрттің таралуын бақылау және тепловизор камералары арқылы көздерді іздеу үшін белсенді түрде қолданылады. Олардың артықшылығы апат аймақтарында және жету қиын аймақтарда өте маңызды болып табылатын нақты уақыт режимінде әуеден мәліметтерді жылдам қабылдау мүмкіндігі. Мұндай ұшқышсыз ұшақтардың кемшіліктерінің қатарында ұшудың шектеулі уақыты (әдетте 20-40 минут), әсіресе тепловизор қосылған кезде. Жоғары температура мен түтін жағдайында жұмыс істеу электрониканың қызып кетуіне немесе байланыс ақауларына әкелуі мүмкін.



1.4-сурет - Ұшқышсыз ұшатын мониторингке арналған жүйе

Тағы бір модель 1.5-суреттегі сәлемдемелерді, соның ішінде DPD логистикалық компаниясынан жеткізу үшін пайдаланылатын жүк ұшқышсыз ұшу аппараты. Бұл қысқа және орташа қашықтыққа шағын жүктерді автоматтандырылған тасымалдау үшін арнайы әзірленген көп 8 роторлы дронның типтік мысалы. Мұндай дрондар электронды коммерция, дәрі-дәрмектің шұғыл жеткізілімі, жету қиын аймақтардағы логистика салаларында және жерүсті көлігі шектеулі жағдайларда танымал бола түсуде.

Жүк ұшатын ұшақтардың негізгі кемшіліктері жүктің салмағына және ауа райы жағдайына байланысты олардың шектеулі ауқымы мен ұшу уақыты болып табылады. Ағымдағы аккумуляторлық технология салмағы 2-5 кг-ға дейінгі сәлемдемелерді зарядтау қажет болғанға дейін 20-30 км-ге дейінгі қашықтыққа тасымалдауға мүмкіндік береді.

Заманауи жер үсті мобильді роботтар қозғалуға әртүрлі беттерде қабілетті. Тасымалдаудың ең көп тараған құралы - дөңгелектер, жолдар, аяқтар немесе жылан тәрізді механизмдер. Құтқару роботтарының көпшілігі жер сілкінісі, цунами, циклон және басқа да табиғи немесе техногендік апаттардың салдарынан болатын тегіс немесе қиын беттерді еңсеру үшін дөңгелектерді немесе жолдарды пайдаланады.[5]



1.5-сурет - Ұшқышсыз ұшатын тасымалдаушы жүйе

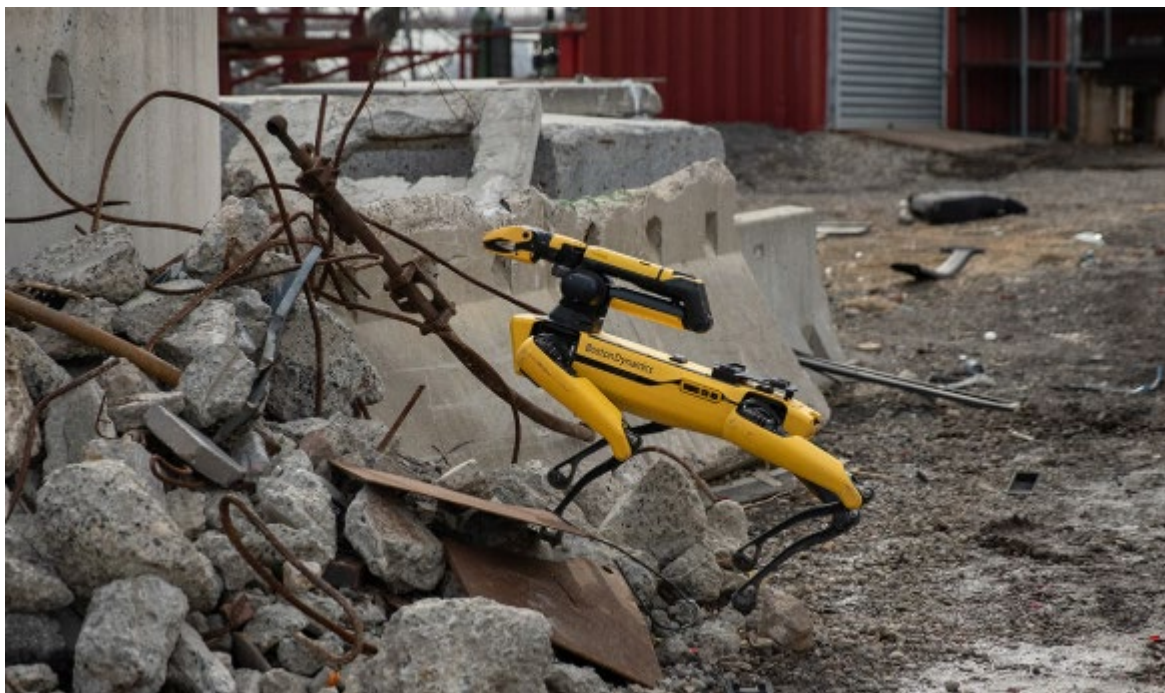
Ұшқышсыз жер үсті көліктері қайықтар немесе құтқару салдары сияқты су бетінде жұмыс істей алатын мобильді роботтар. Роботтың бұл түрі океанографиядағы, әскери және гуманитарлық зерттеулердегі мәселелерді шешуге арналған. Күн батареяларымен жабдықталған мұндай роботтар бірнеше ай бойы үздіксіз жұмыс істей алады.

2017 жылдың 9 тамызында Қытайдың солтүстік-батысында, Шыңжаң-Ұйғыр автономиялық ауданында шамамен 15-35 шақырым тереңдікте 7 баллдық жер сілкінісі болды. Қытайдың жекеменшік DJI компаниясы зардап шеккен аймаққа сенімді аэротүсірілімдер жүргізу және апат аймағындағы жағдайдың өзгеруін бақылау үшін өз конструкциясы бойынша бірнеше ұшқышсыз ұшу аппараттарын жіберді.

Келесі 1.6-суретте Boston Dynamics компаниясының Spot, манипулятормен жабдықталған төртаяқты роботы көрсетілген. Бұл робот іздестіру-құтқару жұмыстарында, өндірістік инспекцияда, әскери тапсырмаларда, сондай-ақ адамдар үшін қауіпті аймақтарда - мысалы, ғимараттардың құлауы, химиялық заттардың ағуы немесе радиациялық апаттар кезінде белсенді қолданылады. Жоғары маневрлік және кедергілерді болдырмау мүмкіндіктерімен Spot жету қиын жерлерге еніп, бейне/сенсор ақпаратын нақты уақытта жібере алады.

Spot-тың кемшіліктері оның жоғары құнын он мыңдаған доллар құрайды, бұл оның кең таралуын шектейді. Сонымен қатар, оның тиімділігі бақылау сапасына тікелей байланысты - автономды функциялар күрделі ортадағы адам операторын әлі толықтай алмастырмайды. Сондай-ақ Spot батареясының қызмет ету мерзімі шектеулі (бір зарядта шамамен 90 минут)

және төтенше жағдайларда (лай, қар, қатты өрт) оның қозғалғыштығы мен сенсорлары оның тиімділігін төмендетуі мүмкін(сурет 1.6).[6]



1.6-сурет – Төтенше жағдай кезіндегі жұмыс жасаушы робот ит

Зардап шеккен аймақта талшықты-оптикалық кабельдер зақымданғандықтан, базалық станциялар бұзылып, байланыс болмағандықтан, әуе байланысының базалық станцияларына айналған және 100 шегінде үздіксіз Интернетке қол жеткізуді және қауіпсіз деректерді беруді қамтамасыз ететін байланыстырылған ұшқышсыз ұшу аппараттары да пайдаланылды.

1.3 Жер сілкінісі кезіндегі құтқару операцияларының ерекшеліктері

Жер сілкінісі кезіндегі құтқару операциялары басқа төтенше жағдайлармен салыстырғанда бірқатар ерекшеліктерге ие. Бұл апаттың динамикалық сипаты, құрылыс құрылымдарының қирауы, инфрақұрылымның зақымдануы және қосымша қауіптердің (өрттер, газдың ағып кетуі, су тасқыны) пайда болуы құтқару жұмыстарын айтарлықтай қиындатады.

Жер сілкінісінің басты ерекшелігі оның кенеттен пайда болуы. Бұл апат түрін алдын ала дәл болжау мүмкін емес, сондықтан дайындық шаралары шектеулі сипатта болады. Жер сілкінісі болғаннан кейін алғашқы 24-48 сағат ішінде қайталама дүмпулер болуы ықтимал. Бұл дүмпулер құтқару жұмыстарын жүргізіп жатқан мамандар мен әлі құтқарылмаған жәбірленушілерге қосымша қауіп төндіреді.

1.7-суретке сәйкес, жер сілкінісінен кейін көбінесе ғимараттар қирап, жолдар мен инженерлік желілер зақымданады. Қираған ғимараттардың астында қалған адамдарды іздеу үшін термографиялық камералар, дрондар,

және қуыстарды тексеруге арналған арнайы құрылғылар қолданылады. Апат аймағына баратын жолдардың, көпірлердің бұзылуы құтқару топтарының жетоне қиындатады, ал электр және байланыс желілерінің үзілуі ақпарат алмасу мен координацияны әлсіретеді. Жер сілкінісімен қатар, бірқатар қосымша қауіптер туындауы мүмкін.



1.7-сурет – Жер сілкініс көрінісі

Өрт қаупі газ құбырлары мен электр желілерінің зақымдануы өрт шығуына себеп болуы мүмкін. Улы газдардың таралуы химиялық зауыттар немесе өндірістік объектілердің зақымдануы қоршаған ортаға қауіпті заттардың таралуына әкеледі. Су тасқыны су қоймалары мен дамбалардың бұзылуы апаттың ауқымын арттырып, құтқару жұмыстарын қиындатады. Осы кезеңде құтқару жұмыстары бірнеше кезеңге бөлінеді:[7]

1.4 Қазақстандағы төтенше жағдайлар жүйесі: цифрлық жаңғырту

Қазақстан Республикасының Төтенше жағдайлар министрлігі (ТЖМ) төтенше жағдайларды болжау және басқару үшін 2024 жылы Геоақпараттық жүйені (ГАЗ ТЖ) енгізді. Бұл жүйе табиғи және техногендік апаттарды, мысалы, өрттерді, метеорологиялық апаттарды және мұнай төгілулерін бақылауға мүмкіндік береді. Сондай-ақ, жүйеде аймақтардың қауіпсіздік паспорттары мен тарихи деректер негізінде төтенше жағдайлардың атласы қамтылған. Сел қаупі жоғары кезеңдерде ТЖМ Түрген көлдері мен Талғар өзеніндегі бөгеттерге РТЗ-камералар орнатты. Сонымен қатар, мореналық көлдерді автоматтандырылған бақылау жүйесі мен ұшқышсыз ұшу

аппараттары арқылы мониторинг жүргізіледі. Орман және дала өрттерін ерте анықтау үшін «Сергек» бейнебақылау жүйесі мен арнайы өртке қарсы камералар енгізілді. Бұл жүйелер «Ертіс Орманы», «Бурабай», «Семей орманы» және «Медеу» сияқты орман шаруашылықтарында қолданылады.

Алматы және оған жақын аймақтарда сейсмикалық белсенділікті бақылау үшін GeoRAPTOR жүйесі пайдаланылады. Бұл жүйе физикалық датчиктер арқылы жер сілкіністерін нақты уақыт режимінде тіркейді. ТЖМ халықты төтенше жағдайлар туралы жедел хабардар ету үшін отандық аппараттық-бағдарламалық кешен әзірлеуде. Бұл кешен ел бойынша сирена-жариялау құрылғыларын басқаруға мүмкіндік береді. Сонымен қатар, пайдаланушылардың орналасқан жерін анықтап, төтенше жағдайлар кезінде әрекет ету нұсқауларын беретін мобильді қосымша әзірленуде.

ТЖМ Ішкі істер министрлігі, Ұлттық гвардия, Қылмыстық-атқару жүйесі комитеті, орман шаруашылықтары және "Qarmet" ұйымдарымен жедел өзара әрекеттесу үшін бірыңғай радиожелі құру жұмыстарын жүргізуде. Аймақтардың қауіпсіздік паспорттарын цифрландыру арқылы ТЖМ деректерді жаңарту және пайдалану процесін жеңілдетті. «Цифрлық қауіпсіздік паспорты» және «Интерактивті карта» қосалқы жүйелері енгізілді.

ТЖМ қызметкерлерінің 20 мыңнан астам жеке ісі цифрландырылды. Бұл кадрлық жұмыстарды автоматтандыруға, есептілікті жеңілдетуге және бюрократиялық жүктемені азайтуға мүмкіндік берді.[2]

1.5 Робототехникалық жүйелердің құтқару операцияларында қолданылуы: халықаралық тәжірибелердің талдауы

Төтенше жағдайлар кезінде құтқару операцияларын жүргізу адам өмірін құтқарудың аса жауапты және күрделі процесі. Бұл салада робототехникалық жүйелерді қолдану соңғы жылдары кең таралып келеді. Олар адамдар үшін қауіпті немесе қолжетімсіз аймақтарда жұмыс істей алады, құтқару жылдамдығын арттырып, адам шығынын азайтуға септігін тигізеді. Әлем елдерінің тәжірибесі бұл бағыттың өзектілігі мен болашағын көрсетіп отыр.

Жапония табиғи апаттарға бейім елдердің бірі, сондықтан бұл мемлекет құтқару жұмыстарында роботтарды қолдануға ерекше мән береді. 2011 жылы Фукусима-Даичи атом электр станциясындағы апат кезінде адам өміріне қауіпті ортада жұмыс істеу үшін арнайы роботтар қолданылды. Олар ғимарат ішін зерттеді, радиациялық деңгейлерді өлшеді және камералар арқылы бейне бақылау жүргізді. Бұл роботтардың нақты апатты жағдайда сәтті қолданылған алғашқы ірі мысалдарының бірі.

АҚШ-та робототехниканы құтқару саласында қолдану кеңінен зерттеліп жатыр. Texas A&M университеті жанындағы CRASAR (Center for Robot-Assisted Search and Rescue) іздеу-құтқару роботтарын дамытумен айналысатын жетекші орталықтардың бірі. Бұл орталықтың мамандары роботтарды 2001 жылғы 11 қыркүйек оқиғасынан кейін құлаған ғимараттардың астында тірі

қалғандарды іздеуге пайдаланған. Бұдан кейін де көптеген дауыл, жер сілкінісі мен техногендік апаттар кезінде роботтар қолданылды.

Германияның Fraunhofer институты және TU München университеті құтқару роботтарын SLAM және автономды навигациямен қамтамасыз ету бағытында жұмыс істейді. Бұл мекемелер DARPA Robotics Challenge сияқты халықаралық жарыстарға қатысып, күрделі ортада өзін-өзі басқару қабілеті жоғары роботтар жасап шығарды. Олардың кейбір модельдері адамдармен бірлесіп жұмыс істей алады, яғни аралас ортада тиімді кооперация жасай алады.

Оңтүстік Кореяда робототехника дамуының маңызды бағыты интеллектуалды іздеу жүйелері мен дрондарды құтқару жұмыстарында пайдалану. Мұнда KAIST университетінің зерттеушілері роботтарға жасанды интеллект пен машиналық оқыту енгізіп, тірі адамдарды жылу сигналдары арқылы анықтайтын жүйелерді сынақтан өткізуде.

Біріккен Ұлттар Ұйымы мен Қызыл Крест сияқты халықаралық ұйымдар да роботтарды төтенше жағдайларда қолдану мүмкіндігін зерттеп жатыр. Олар табиғи апат аймақтарына автономды дрондар мен роботтарды жіберу арқылы алдын ала бағалау жұмыстарын жүргізіп, апаттың ауқымын нақты анықтауға көмектеседі.[8]

1.6 Мақсаттарды іздеу мен анықтаудың қолданыстағы әдістері мен алгоритмдері

Роботтар тобын іздеу әртүрлі қолданбаларда қолданылады: робот шаңсорғыштың жолын жоспарлау, жер жырту, егін жинау, шөп шабу, бояу және тазалау және өрт сөндіру. Жер сілкінісінен аман қалғандарды табу зардап шеккен аймақтарды іздеуді және мүмкіндігінше қысқа мерзімде аман қалғандарды табуды қамтиды. Иттер іздеу мәселелерін өте жақсы шешеді, бірақ олар қысқа уақыт ішінде жұмыс істей алады және жануарларға кедергі келтірмеу үшін басқа жұмыстарды тоқтату керек. Бірақ иттер әрқашан жету қиын жерлерге жете алмайды. Бұл жағдайда өмір белгілерін анықтауға арналған камералар мен сенсорлармен жабдықталған миниатюралық роботтар пайдалырақ болуы мүмкін. Іздеу процесіне төрт талап қойылады. Біріншісі - іздеу аймағын максималды қамту. Екіншісі, жабу кезінде қайталануларды жою. Үшіншіден, соқтығыспай кедергілерден құтылу. Төртіншіден, резервті көбейту арқылы ақауларға төзімділікті қамтамасыз ету. Негізінде, жоғарыда аталған төрт ережені сақтау жабынның оңтайлы траекторияларын табуға кепілдік береді. Әрі қарай іздеу аймағын қамту мәселелерін шешудің әртүрлі алгоритмдері қарастырылады.

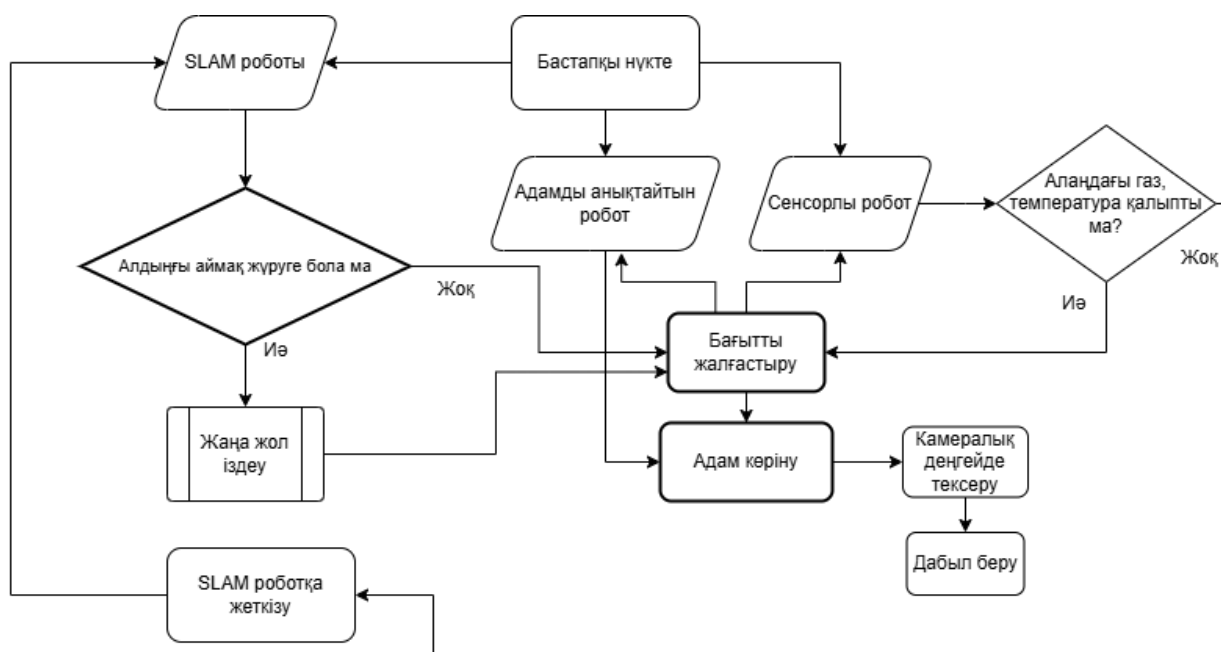
Белгісіз аймақта стохастикалық іздеу әдісін қолданатын, іздеу аймағында роботтарды тарату арқылы тиімділікті арттыратын Balch және Arki сияқты көптеген алгоритмдер бар. Бірақ бұл пассивті іздеу әдісі қайталанудан аулақ бола алмайды, бұл іздеу жылдамдығын төмендетеді. Паркер іздеу жылдамдығын жақсарту үшін жаңа бөлінген кооперативтік көп агенттік

үлгісін жасады. Бұл әдіс сенімді және ақауларға төзімді толық таратылған іздеу болып табылады.

Бұл жүйелер қауіпті және қол жетпейтін аймақтарға жылдам қол жеткізуді қамтамасыз ету арқылы тірі қалғандарды іздеу және құтқару уақытын айтарлықтай қысқартуға мүмкіндік береді. Олар сондай-ақ құтқарушылардың қауіпсіздігін оларға объектілердің жағдайы мен қоршаған орта туралы ақпарат беру арқылы жақсарта алады.

Дегенмен, мұндай жүйелерді сәтті енгізу және орналастыру инженерлердің, зерттеушілердің, мемлекеттік органдардың және гуманитарлық ұйымдардың бірлескен күш-жігерін талап етеді. Сондай-ақ апаттық жағдайларда роботтық жүйелерді пайдаланудың этикалық, құқықтық және әлеуметтік аспектілерін ескеру маңызды.

Жер сілкінісінен құтқарудың көп агентті роботтық жүйелері қоғамның төтенше жағдайларға әрекет ету және адамдардың өмірін сақтап қалу қабілетін айтарлықтай жақсарта алатын қуатты құрал болып табылады. Егер олар дұрыс әзірленсе және пайдаланылса, олар біздің әлемнің қауіпсіздік және қорғаныс инфрақұрылымының ажырамас бөлігі бола алады.



1.8-сурет – Жүйенің функционалдық блок схемасы

Осы мақсатқа жету үшін 1.8-суреттегідей зерттеу аясында келесі міндеттер қойылады:

- Құтқару жұмыстарында қолданылатын роботтық жүйелерді зерттеп, олардың артықшылықтары мен кемшіліктерін анықтау.
- Роботтардың өзара әрекеттесу және үйлестіру алгоритмдерін қоса алғанда, көп агенттік жүйенің архитектурасын әзірлеу.
- Қоршаған ортаның өзгеруін ескере отырып, агенттер арасында тапсырмаларды оңтайлы бөлу алгоритмдерін құру.

д) Жүйені модельдеу және оның тиімділігін бағалау үшін имитацияланған үйінділерде эксперименттік сынақтарды жүргізу.

е) Жүйені нақты құтқару жұмыстарында практикалық қолдану бойынша ұсыныстар әзірлеу.

Ғылыми жаңалық

Күрделі үйінділер мен қираулар жағдайында жұмыс істеуге бейімделген көп агентті роботтық жүйенің архитектурасы ұсынылды. Агенттер арасында олардың ағымдағы жағдайы мен қолда бар ресурстарын ескере отырып, тапсырмаларды динамикалық бөлу алгоритмі әзірленді. Жүйенің жұмысы әртүрлі сценарийлерде имитацияланды, бұл роботтардың өзара әрекеттесуінің ең тиімді стратегияларын анықтауға мүмкіндік берді.

Зерттеу объектісі

Зерттеу нысаны іздестіру-құтқару жұмыстарын орындауға арналған мобильді роботтар қамтитын көп агентті роботтық жүйе болып табылады.

Зерттеу пәні

Зерттеу пәні күрделі орта жағдайында тапсырмаларды орындау тиімділігін арттыруға бағытталған мультиагенттік жүйенің агенттері арасындағы өзара әрекеттесу және үйлестіру алгоритмдері болып табылады.

Зерттеу әдістемесі

Зерттеу келесі әдістерге негізделген:

а) Әдебиеттерді талдау: құтқару жұмыстарында қолданылатын заманауи роботтық жүйелер мен алгоритмдерді зерттеу.

б) Модельдеу: сценарийлер модельдеу және агенттер арасындағы өзара әрекеттесуді талдау үшін ROS2 (Robot Operating System 2), және Gazebo бағдарламалық орталарын пайдалану.

с) Эксперименттік зерттеу: бұзылу жағдайларын имитациялайтын стендте тәжірибелік жүйені сынау.

д) Алгоритмдік модельдеу: тапсырмаларды тарату және роботтар арасындағы қайшылықтарды болдырмау алгоритмдерін әзірлеу және сынау.

1.1-кесте – Ғылыми зерттеу көрсеткіші

Ғылыми зерттеу	Жауап беру уақыты	Командалық конфигурация	Табыс деңгейі	Қоршаған ортаның күрделілігі
Алотойби және т.б., 2019	Көптеген құтқарулар $\leq$ 04:50:02 (95% сенімділік)	Бірнеше ҰҰА (көрсетілмеген)	74% құтқарылды	Белгіленген жоқ
Балакирски й және т.б., 2007 ж	Деректер жоқ	12 бірдей жерүсті роботтарға дейін	Деректер жоқ	Қалалық үйінділер (> 11 000 м²)
Бек және т.б., 2016 ж	Қолданыстағы әдістермен салыстырғанда 20-25% төмендету	Гетерогенді (анықталмаған)	Деректер жоқ	Белгіленген жоқ
Бақши және т.б., 2023	30 минуттан аз	Гетерогенді (ҰЖА және ҰҰА)	80% (имитациялық), 100% (далалық сынақтар)	Ұйымдастырылған табиғи орта (75 000 м²)
Велагапуд и және т.б., 2008 ж	Деректер жоқ	4, 8, 12 бірдей NGR	Деректер жоқ	Бөлмелері мен кедергілері бар қалалық лабиринт
Такахаш и және т.б., 2001	Деректер жоқ	Гетерогенді (өрт, жедел жәрдем, полиция)	Деректер жоқ	Қала үйіндісі
Сouseiro және т.б., 2013 ж	Деректер жоқ	Деректер жоқ	Деректер жоқ	Деректер жоқ
Хашемипу р және т.б., 2018	Операцияның аяқталу уақыты (көрсетілмеген)	Деректер жоқ	Деректер жоқ	Деректер жоқ

1.1-кестеге сәйкес, ғылыми зерттеулерді талдау олардың тек үшеуі ғана жер сілкінісінен құтқару жағдайында көп агенттік жүйелердің әрекет ету

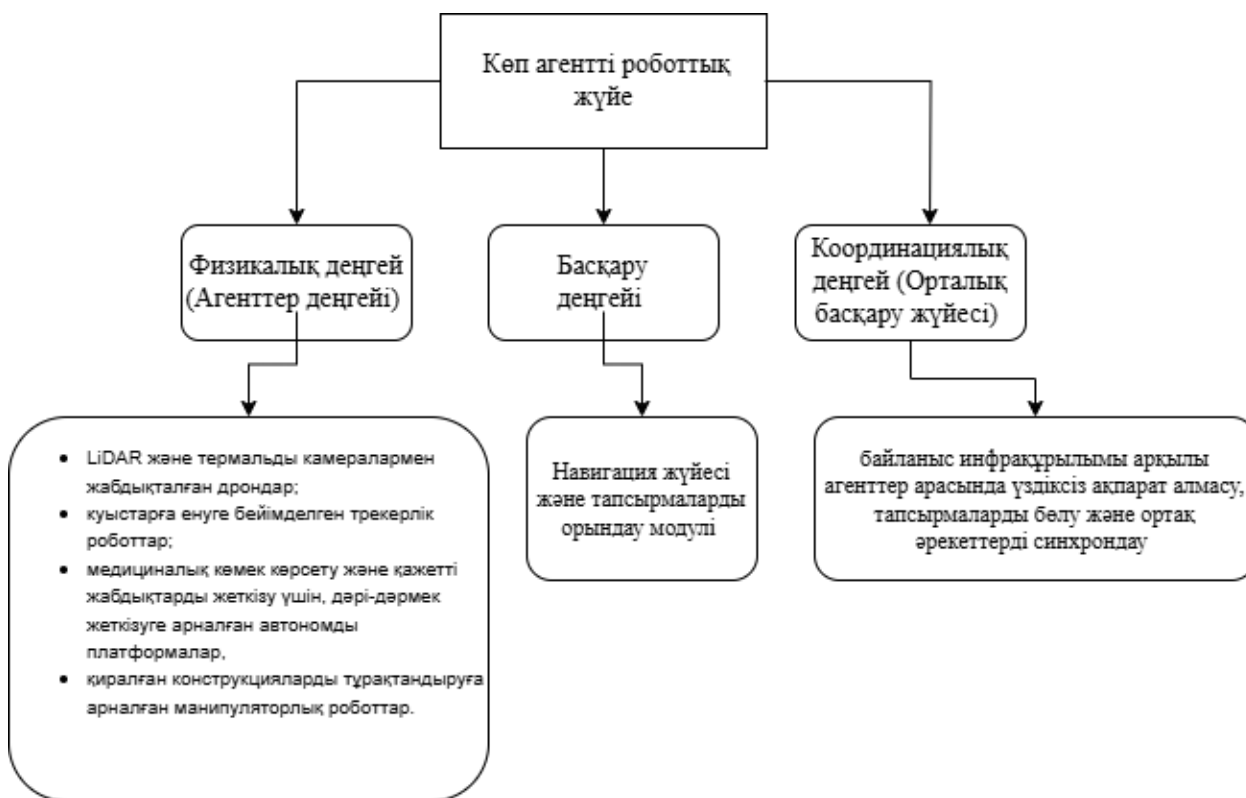
уақытын сандық түрде анықтағанын көрсетті. Бір жағдайда уақыт аталды, бірақ саны анықталмады, ал алты зерттеуде ақпарат жоқ болды. Командалық конфигурациясы алты зерттеуде нақтыланды: екеуінде тек қана ұшқышсыз ұшу аппараттары, үшеуінде жердегі роботтар (ҰҰА) және үшеуінде гетерогенді командалар қолданылды. Құтқару сәттілігі екі зерттеуде ғана сандық түрде бағаланды, ал бір жағдайда нақты деректерсіз жақсару байқалды. Қоршаған ортаның күрделілігі он дереккөздің тек төртеуінде ғана егжей-тегжейлі сипатталған: үшеуі қалалық қоқыстарды зерттеді, біреуі құрылымсыз табиғи орталарды зерттеді.

Үйлестіру тәсілдерін масштабтауға және жүзеге асыруға қатысты бірнеше негізгі қорытындылар пайда болды. Мысалы, Велагапуди және т.б. (2008) роботтар саны 4-тен 8-ге дейін өскен сайын жүйе өнімділігі өсті, бірақ олардың саны одан әрі 12-ге дейін өскен сайын төмендеді. Бакстер және т.б. (2007) әлеуетті өрістері ортақ жүйелер (6+ роботтары бар) қосылмаған жүйелерден асып түсетінін көрсетті. LSAR алгоритмі (Алатойби және т.б., 2019) ҰАО санының ұлғаюымен құтқару пайызының сызықты дерлік өсуін қамтамасыз етті. Тапсырмаларды бөлу тәсілдерінің ішінде мыналар ерекшеленді: LSAR (Қабатты іздеу және құтқару) тапсырмаларды дрондар арасында тиімді бөлу; DMCsim Командалық конфигурацияны оңтайландыру үшін машиналық оқыту пайдалану; және мульти-агенттік жүйені басқаруда операторға қолдау көрсететін автоматтандырылған кеңесшілер (Розенфельд және т.б., 2015 ж). [12-19]

2 Зерттелетін жүйенің әдіснамасы және моделі

2.1 Роботтық жүйенің негізгі деңгейлері

2.1-суретте көрсетілгендей, жер сілкінісінен құтқару операцияларына арналған ұсынылатын көп агентті роботтық жүйенің архитектурасы үш негізгі деңгейден тұрады: физикалық деңгей, басқару деңгейі және координациялық деңгей. Бұл архитектура апат аймағында тиімді және үйлесімді жұмыс істеуге мүмкіндік береді.



2.1-сурет – Жүйе сәулеті

Физикалық деңгей (Агенттер деңгейі) бұл деңгейде әртүрлі мақсаттарға арналған роботтық агенттер орналасады. Қиралған ғимараттарға енуге және құрбандарды іздеуге арналған шағын, икемді роботтар. Мысалы, LiDAR және термальды камералармен жабдықталған дрондар, қуыстарға енуге бейімделген трекерлік роботтар, медициналық көмек көрсету және қажетті жабдықтарды жеткізу үшін, дәрі-дәрмек жеткізуге арналған автономды платформалар, су және тамақ тасымалдауға арналған роботтар, қиралған конструкцияларды тұрақтандыруға арналған манипуляторлық роботтар

Әрбір агент өзінің сенсорлық жүйесімен (LiDAR, камера, GPS), процессорлық блокпен және байланыс модулімен жабдықталған.

Басқару деңгейі бұл деңгейде әрбір агенттің тәуелсіз және дербес басқарылуы қамтамасыз етіледі. Жүйе автономды жұмыс істеуге қабілетті, яғни әр агент өз ортасын тану, шешім қабылдау және мақсатқа жету

әрекеттерін орталықсыз жүзеге асыра алады. Мұндай басқару архитектурасы агенттің нақты уақыт режимінде сенімді әрі тиімді әрекет етуіне мүмкіндік береді.

Агенттің негізгі басқару блогына автономды навигация жүйесі және тапсырмаларды орындау модулі енеді.

Автономды навигация жүйесі келесі негізгі компоненттерден тұрады. SLAM (Simultaneous Localization and Mapping) технологиясы агенттің өз орнын анықтау және айналадағы ортаның картасын құру қабілетін қамтамасыз етеді. Кедергілерден айналып өту алгоритмдері нақты уақыт режимінде жұмыс істеп, тосын кедергілерге жауап береді. Динамикалық маршруттық жоспарлау өзгермелі орта жағдайында тиімді бағыттарды есептеп, агенттің мақсатқа жетуін қамтамасыз етеді.[1]

Тапсырмаларды орындау модулі мынадай функцияларды орындайды. Агенттің ағымдағы мақсатын анықтау берілген миссия негізінде локалды мақсаттарды ажырату және приоритеттер қою. Сенсорлық ақпаратты өңдеу LiDAR, камера, IMU секілді құрылғылардан алынған деректерді интерпретациялау. Жеке шешім қабылдау механизмдері алынған ақпарат негізінде тәуелсіз әрекеттерді жоспарлау және орындау.

Бұл басқару деңгейі ROS 2 (Robot Operating System) платформасына негізделген. ROS 2 модульдік архитектурасы мен ашық интерфейстері арқылы жүйеге жаңа мүмкіндіктерді оңай қосуға, агенттер санын кеңейтуге және масштабтауға мүмкіндік береді. Сонымен қатар, ROS 2 платформасының шынайы уақыттағы өңдеу мүмкіндігі құтқару операцияларының сенімділігін арттыруға зор ықпал етеді.

Координациялық деңгей (Орталық басқару жүйесі) көп агентті роботтық жүйеде барлық агенттердің тиімді және үйлесімді әрекет етуі үшін координациялық деңгей маңызды рөл атқарады. Бұл деңгейде орталық басқару жүйесі мен байланыс инфрақұрылымы арқылы агенттер арасында үздіксіз ақпарат алмасу, тапсырмаларды бөлу және ортақ әрекеттерді синхрондау жүзеге асырылады.

Орталықтандырылған басқару пункті келесі функцияларды атқарады. Барлық агенттерден түсетін деректерді жинап, талдау. Ортақ жағдайдың 3D моделін құру арқылы бүкіл апат аймағын визуализациялау. Жүктемені теңдей бөлу мақсатында әр агентке нақты тапсырмалар тағайындау.

Байланыс желісі жүйенің тұрақты және сенімді жұмыс істеуі үшін арнайы ұйымдастырылған. Бұл желіге Mesh (торлы) байланыс құрылымы, яғни агенттер арасында бір-бірімен тікелей байланыс орнату. Кажетті жағдайда жергілікті байланыс түйіндерін (ретранслятор, базалық станция) қолдану, қосымша инфрақұрылымсыз жұмыс істеу үшін арнайы шешімдер (мысалы, өздігінен ұйымдасатын желілер) жатады.

Желілік архитектураның бірнеше нұсқасы қарастырылады. P2P (Peer-to-Peer) моделі барлық агенттер тең дәрежеде бір-бірімен тікелей ақпарат алмаса алады. Орталық серверлік модель басқару мен ақпарат өңдеу орталық түйінге

шоғырланады. Гибридітк архитектура жоғарыда аталған екі әдістің үйлесімі ретінде икемді әрі масштабталатын құрылым ұсынады.

Жүйе жер сілкінісі болғаннан кейінгі апатты жағдайда келесі реттілік бойынша жұмыс істейді. Дрондар апат аймағын жоғарыдан сканерлеп, оның жалпы конфигурациясын анықтайды. Алынған мәліметтер негізінде бастапқы карта құрылады.

Орталық басқару жүйесі дрондардан алынған картографиялық ақпаратты талдап, әрбір агентке өзінің мүмкіндігіне қарай тиімді тапсырмаларды тағайындайды.

Агенттер өз тапсырмаларын орталықтан тыс, дербес орындайды. Олар кедергілерден айналып өтіп, SLAM арқылы навигация жасап, қажетті нысандарды (адамдар, жабдықтар, қауіпті аймақтар) анықтайды.

Әрбір агент өз сенсорларынан алған мәліметтерді орталық жүйеге жібереді. Бұл деректер негізінде жалпы 3D карта мен жағдай бейнесі үнемі жаңарып отырады.

Жаңа алынған мәліметтерге сүйене отырып, жүйе стратегияны өзгертіп, жаңа тапсырмалар тағайындай алады. Бұл жүйенің динамикалық ортаға икемделу қабілетін көрсетеді.

Жүйе келесі заманауи технологияларға сүйенеді:

- a) ROS 2 роботтарды басқару үшін ашық бастапқы платформа
- b) Gazebo нақты жағдайларды модельдеуге арналған симулятор

2.2 Технологияға шолу. ROS2, RViz және Gazebo сипаттамасы

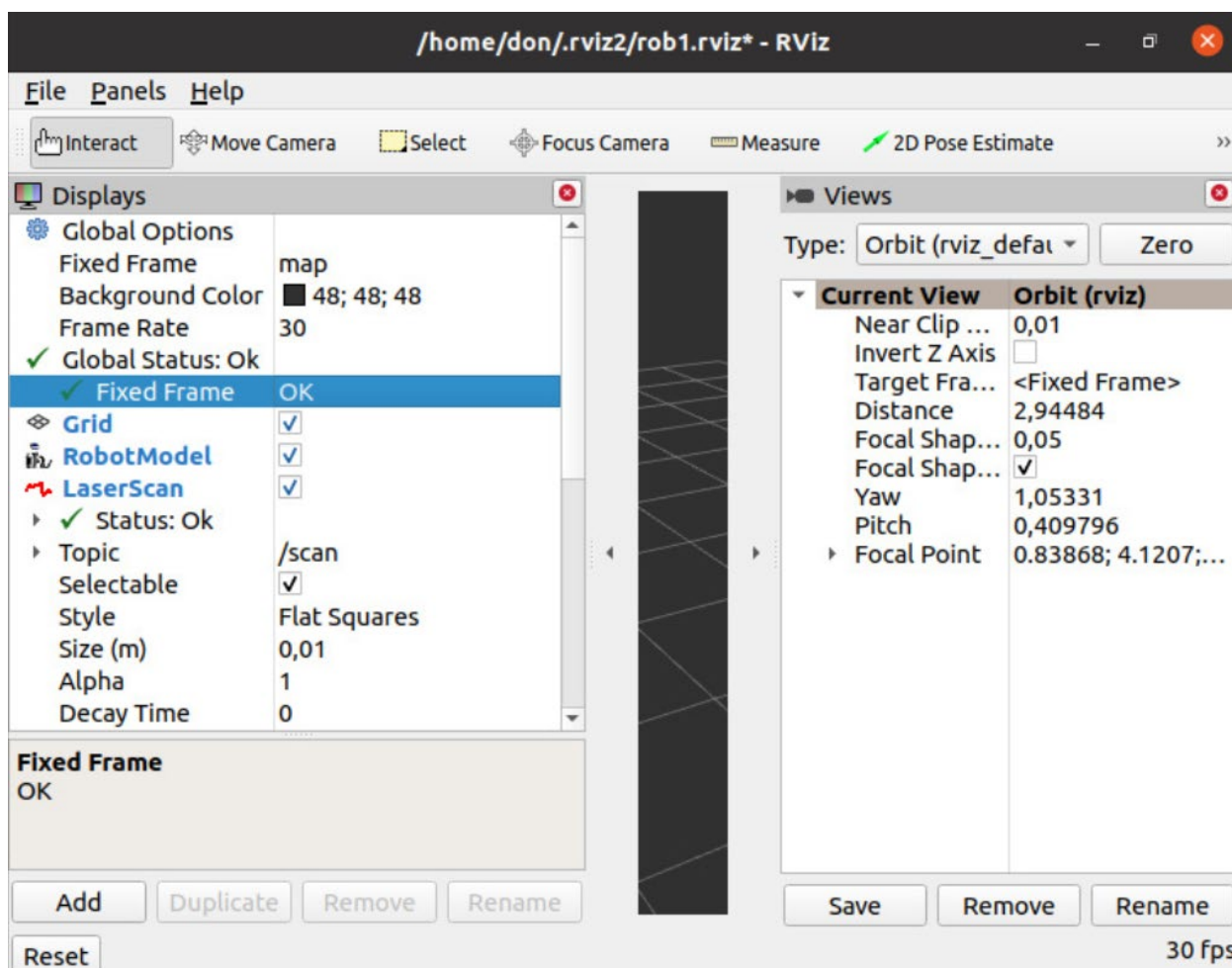
ROS2 Foxy Fitzroy робототехниканы дамытуға арналған заманауи құралдарды ұсынатын, қауіпсіздікті арттыруға, бөлінген жүйелерді басқаруды жақсартуға және процессаралық байланысты оңтайландыруға бағытталған Robot Operating System 2 (ROS2) шығарылымы.

Негізгі ROS2 компоненттері және олардың функциялары түйіндер (Nodes) ROS2 жүйесіндегі негізгі технологиялық блоктар болып табылады. Әрбір түйін сенсорларды басқару, есептеулерді орындау немесе басқару сияқты нақты тапсырмаларды орындайтын бөлек орындалатын процесс. Түйіндер басқаруға ыңғайлы болу үшін топтарға ұйымдастырылған және тақырыптарды жариялау және жазылу арқылы бір-бірімен әрекеттесе алады.

Тақырыптар (Topics) түйіндер арасында хабар алмасу арналары ретінде қызмет етеді. Мысалы, LiDAR түйіні белгілі бір тақырыпқа қашықтық деректерін жариялай алады, ал SLAM түйіні деректерді алу және орта картасын құру үшін осы тақырыпқа жазылады. ROS2 тақырыптары бір бағытты да, екі жақты да хабар алмасуды қолдайды.

Баспагерлер (Publishers) мен (Subscribers) тақырыптарға хабарламалар жіберетін, оларды басқа түйіндерге қолжетімді ететін түйін құрамдас бөліктері. Жазылушылар, сәйкесінше, хабарламаларды алу үшін тақырыптарға жазылады. Бұл орталықтандырылмаған және жоғары теңшелетін деректер алмасу жүйесін құруға мүмкіндік береді.[6]

RViz бұл ROS2 жүйесінде роботтардың жұмысын визуалды түрде бақылауға арналған қуатты құрал. Ол роботтың әртүрлі сенсорларынан алынатын деректерді (мысалы, лазердік сканерлерден (/laser_scan), карталардан (/map), роботтың қазіргі күйінен (/robot_model) және координаталар айналымынан (/tf)) 3D-кеңістікте нақты уақыт режимінде көрсетеді. RViz арқылы роботтың ортаны қалай түсінгенін, оның қозғалыс траекториясын және басқа маңызды ақпараттарды тез талдауға болады. Мысалы, SLAM (Simultaneous Localization and Mapping) технологиясын қолдану кезінде робот қоршаған ортаның картасын жасап, өз орнын анықтайды, ал бұл процестің барлық кезеңдері RViz интерфейсінде анық көрінеді.



2.2-сурет – Rviz басты жұмыс орны

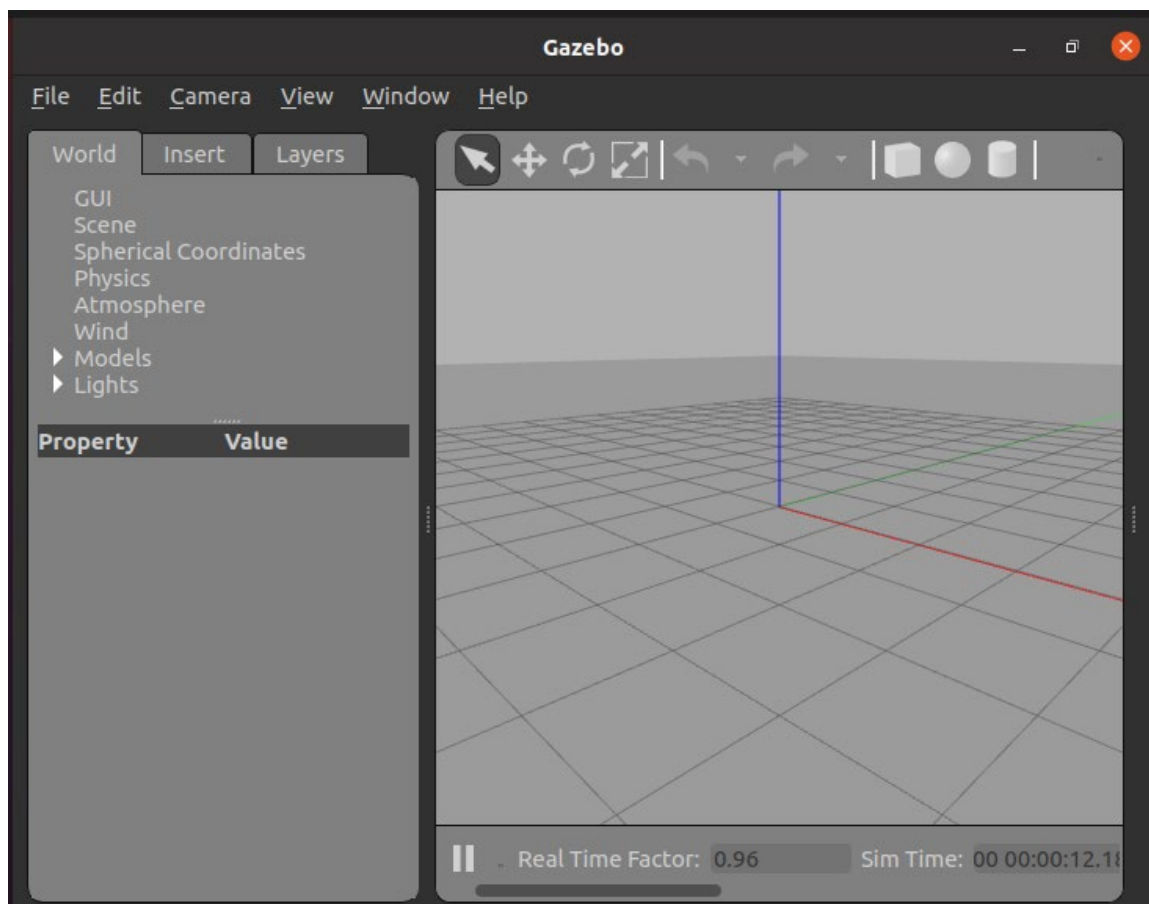
2.2-суретке сәйкес, жобада RViz көбінесе роботтың навигациялық міндеттерін орындауын тексеру үшін қолданылады. Мысалы, 2D Pose Estimate және 2D Nav Goal функциялары арқылы роботқа мақсатты нүктені белгілеуге және оның жолын бақылауға болады. Бұл құрал арқылы роботтың қозғалысындағы қателерді тез анықтап, алгоритмдерді түзетуге болады. Сонымен қатар, RViz мультироторлы дрондардың ұшу траекториясын немесе автономды көліктердің жолды тану процесін сипаттау үшін де пайдалы. Ол

деректерді интуитивті түрде көрсететіндіктен, зерттеушілерге роботтың жұмыс істеу механизмін тереңірек түсінуге мүмкіндік береді.[4]

Қызметтер ROS2 қызметтері түйіндер арасындағы байланыс үшін сұрау-жауап әдісін қамтамасыз етеді. Мысалы, қозғалтқышты басқаратын түйін жылдамдықты өзгерту қызметін ұсына алады, ал басқа түйін жылдамдықты өзгертуге сұраныс жіберу үшін осы қызметке қоңырау шалуы мүмкін.

Хабарламалар тақырыптар немесе қызметтер арқылы түйіндер арасында берілетін деректер құрылымдары. Олар сандар, жолдар, массивтер және тіпті күрделі құрылымдар сияқты әртүрлі деректер түрлерін қамтуы мүмкін.

Gazebo_ros_pkg бұл басқарылатын виртуалды ортада роботтарды тиімді тестілеуге және дамытуға мүмкіндік беретін Gazebo симуляторын ROS2-мен біріктіруге арналған құралдар жиынтығы. Gazebo роботтарды, олардың LiDAR сияқты сенсорларын және жетектерді физикалық тұрғыдан дәл модельдеуді қамтамасыз етеді.



2.3-сурет - Gazebo басты беті

2.3-суретке сәйкес, SLAM тестілеу Gazebo симуляциясында виртуалды LiDAR сенсорларының деректерін пайдаланып SLAM алгоритмдерін сынауға болады, бұл алгоритмдерді нақты роботтарға енгізбес бұрын жақсартуға көмектеседі. Диагностика және жөндеу: түйіндердің өзара әрекеттесуін, хабарлама алмасуын және қызметтің орындалуын Gazebo ортасында

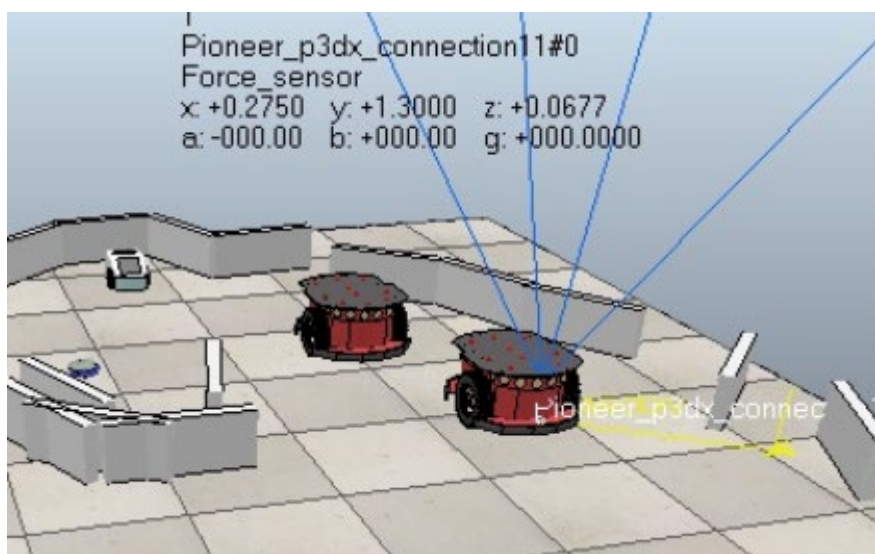
бақылауға және диагностикалауға болады, бұл қателерді табуды және түзетуді жеңілдетеді.

2.3 Дифференциалды роботтың жұмыс принциптері

Дифференциалды робот бұл әрқайсысы бөлек қозғалтқышпен жабдықталған екі тәуелсіз дөңгелекті пайдаланатын мобильді роботтың бір түрі. Дифференциалды роботтың жұмыс принциптері осы дөңгелектердің айналу жылдамдығын басқаруға негізделген, бұл роботқа алға-артқа қозғалуға, бір жерде бұрылуға немесе әртүрлі маневрлер жасауға мүмкіндік береді.[3]

Төмендегі 2.4-суреттегі Pioneer 3-DX роботы робототехника саласындағы зерттеулер мен білім беру жобалары үшін өте ыңғайлы және дәлелденген мобильді платформа. Модульдік архитектурасы мен сенсорларды (УДЗ, камералар, лидар) кең қолдауының арқасында ол автономды навигация, SLAM, карта жасау және көп агенттік өзара әрекеттесу тапсырмаларын орындауға тамаша мүмкіндіктер береді. Оның дифференциалды жетек жүйесі үй ішінде жоғары жүргізу дәлдігі мен маневрлілігін қамтамасыз етеді.

ROS қолдауы, симулятор үйлесімділігі және үлкен халықаралық пайдаланушы базасы Pioneer 3-DX алгоритмін жылдам әзірлеу және тестілеу үшін ыңғайлы платформаға айналдырады. Ол тұрақтылық, ашық архитектура және заманауи бағдарламалық шешімдермен интеграцияның қарапайымдылығы маңызды болып табылатын білім беру мақсаттары мен қолданбалы зерттеулер үшін өте қолайлы. Біздің жобада Pioneer P3-DX роботын негіз ретінде алып, оған әртүрлі модульдер қостық: SLAM үшін лидар, адамды тану үшін камера және қауіптілік (газ немесе температура) үшін арнайы сенсорлар. Бұл роботтар өзара ақпарат алмасып, көп агентті жүйе ретінде әрекет етеді.

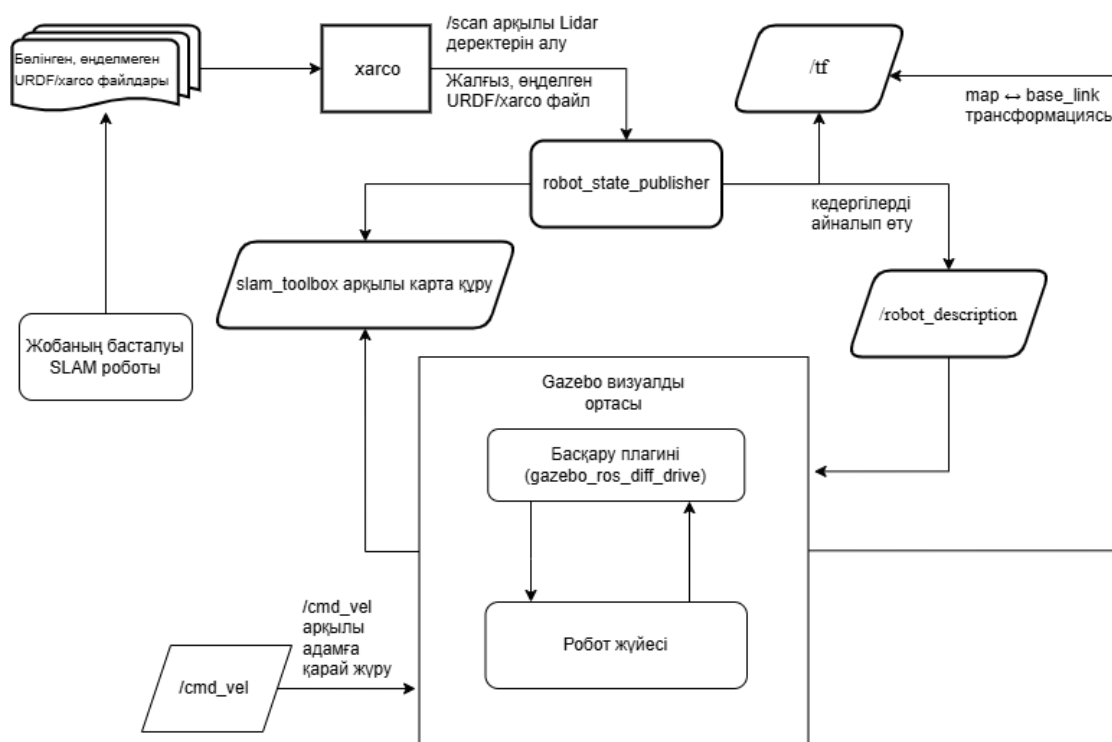


2.4-сурет - CoppeliaSim графикалық интерфейсі Pioneer P3-DX

Pioneer 3-DX бұл жоғарыда сипатталған жүйе үшін негіз болып табылады. Оны лидар, камера және сенсорлармен жабдықтау арқылы SLAM, адамды тану және қауіптілікті мониторингілеу міндеттерін орындауға болады. ROS 2 арқылы интеграциялау оны икемді және масштабтанатын етеді. Дифференциалды робот әдетте келесі негізгі компоненттерден тұрады: Шасси, дөңгелектер, қозғалтқыштар, сенсорлар Шасси роботтың барлық элементтері орнатылған платформа. Дөңгелектер роботтың екі жағында орналасқан екі жетекші дөңгелек. Көбінесе оларға тұрақтандыру үшін бір-екі қосымша тірек дөңгелек қосылады, олар қозғалмайды. Қозғалтқыштар дөңгелектерді айналдыратын приводтар. Сенсорлар кеңістікте бағдарлау үшін қолданылатын датчиктер, мысалы, ультрадыбыстық датчиктер, жарық сезгіштер, камералар және т.б.[10]

2.5-суретке сәйкес, дифференциалды роботтың жұмыс істеуінің негізгі принципі сол және оң дөңгелектердің жылдамдықтарының айырмашылығында жатыр, бұл оған маневр жасауға мүмкіндік береді. Негізгі қозғалыс сценарийлері:

Түзу сызықты қозғалыс роботтың түзу алға қарай қозғалуы үшін екі дөңгелек те бір бағытта бірдей жылдамдықпен айналуы керек. Егер екі дөңгелек те бірдей жылдамдықпен, бірақ артқа қарай айналса, робот артқа қарай қозғалады. Бұрылу егер бір дөңгелек екіншісінен жылдам айналса, робот баяу айналатын дөңгелектің бағытына қарай бұрылады. Роботтың бір жерде бұрылуы (немесе айналып кетуі) үшін бір дөңгелек алға, ал екіншісі артқа қарай бірдей жылдамдықпен айналуы керек. Токтау роботты тоқтату үшін екі дөңгелек те айналуды тоқтатады.

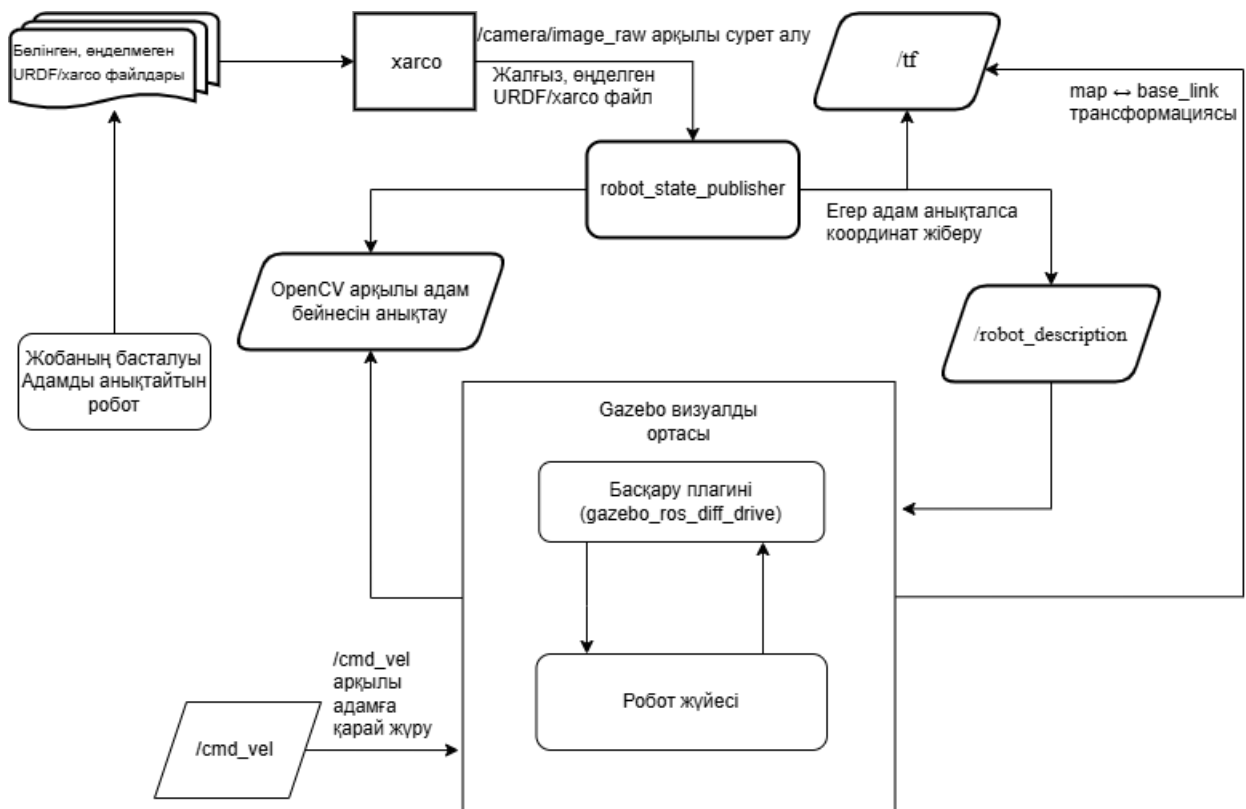


2.5-сурет - SLAM роботы

Дифференциалды роботты басқару әртүрлі әдістермен жүзеге асырылуы мүмкін, қашықтан басқару пульті арқылы қолмен басқарудан бастап, сенсорлар мен күрделі навигациялық алгоритмдерді пайдаланатын толық автономды қозғалысқа дейін. Автономды роботтар жиі SLAM (Бір мезгілде локализациялау және карта жасау) сияқты алгоритмдерді пайдаланады, бұл қоршаған ортаның картасын жасауға және сол картадағы өз орнын анықтауға мүмкіндік береді.

2.4 Навигация үшін LiDAR және SLAM пайдалану

LiDAR (Light Detection and Ranging) қоршаған ортадағы объектілерге дейінгі қашықтықты өлшеу үшін лазерлік импульстарды пайдаланатын технология. Содан кейін бұл деректер аймақтың нақты карталарын жасау немесе объектінің орнын анықтау үшін пайдаланылады. Мысалы, rplidar a1. RPLIDAR A1 бірнеше негізгі құрамдас бөліктерден тұратын ықшам лазерлік қашықтық өлшегіш. 2.6 суреттегідей, ол тандемде жұмыс істейтін лазерлік эмитент пен қабылдағышқа негізделген. Лазерлік модуль айналадағы нысандардан шағылысқан және қабылдағышқа қайтарылатын жарық импульстарын шығарады. Кірістірілген айналдыру механизмі 360 градустық көріністі қамтамасыз етеді, бұл лидарға айналасындағы кеңістікті сканерлеуге мүмкіндік береді.[5]



2.6-сурет - Адам танитын видеожүйесінің функционалдық сұлбасы

RPLIDAR A1 жұмыс принципі уақыт өлшеуге негізделген. Лазерлік импульс жіберіліп, объектіден шағылысқан кезде, лидар сигналдың қайтарылуына кететін уақытты өлшейді. Осы уақыт негізінде нысанға дейінгі қашықтық есептеледі.

RPLIDAR A1 кеңістікті сканерлеген кезде ол сканерлеу бұрышын (0° -тан 360° -қа дейін), нысанға дейінгі қашықтықты және өлшеудің сенімділігін көрсететін сигнал сапасын қоса деректерді жасайды. Бұл деректер компьютерге немесе контроллерге пакеттер түрінде беріледі.

ROS (Robot Operating System) контекстінде LiDAR деректері деректердің бір өлшемді сканерлеу (2d) немесе үш өлшемді нүкте бұлттары (3d) түрінде ұсынылуына байланысты әдетте `'sensor_msgs/LaserScan'` немесе `sensor_msgs/PointCloud2` түріндегі хабарлар ретінде өңделеді және жіберіледі.[11]

`sensor_msgs/LaserScan`: бұл хабар түрі әрқайсысы қашықтық пен бұрыш деректері бар сәулелер жиынтығы болып табылатын 2D сканерлеулерін жіберу үшін пайдаланылады. Бұл сканерлеулер ұшақта қоршаған ортаның «суретін» жасайды, содан кейін оны навигация және кедергілерді анықтау үшін пайдалануға болады.

`sensor_msgs/PointCloud2`: бұл хабар бұлттағы әрбір нүкте кеңістіктегі координаттарды (X, Y, Z) қамтитын 3D деректерін көрсету үшін пайдаланылады және сонымен қатар көрсетілген сигналдың қарқындылығы сияқты қосымша ақпаратты қамтуы мүмкін.

2.7 суретте көрсетілгендей, LiDAR деректері жарияланған тақырыптарды атауға болады, мысалы, `'LaserScan'` үшін `'/scan'` немесе `'PointCloud2'` үшін `'/point_cloud'`. Бұл деректерді SLAM немесе жолды жоспарлау тапсырмаларын орындау үшін пайдалана алатын басқа түйіндер өңдейді.



2.7-сурет – CorreliaSim графикалық интерфейсында камерамен робот көрінісі

SLAM (Simultaneous Localization and Mapping) роботты кеңістікте бір уақытта локализациялау және қоршаған ортаның картасын құру процесі. ROS жүйесі GMapping, Hector SLAM, Cartographer және SLAM Toolbox сияқты әртүрлі SLAM алгоритмдерін жүзеге асырады, олар LiDAR, камералар, инерциялық өлшем бірліктері (IMU) сияқты сенсорлардан алынған деректер негізінде жұмыс істейді.

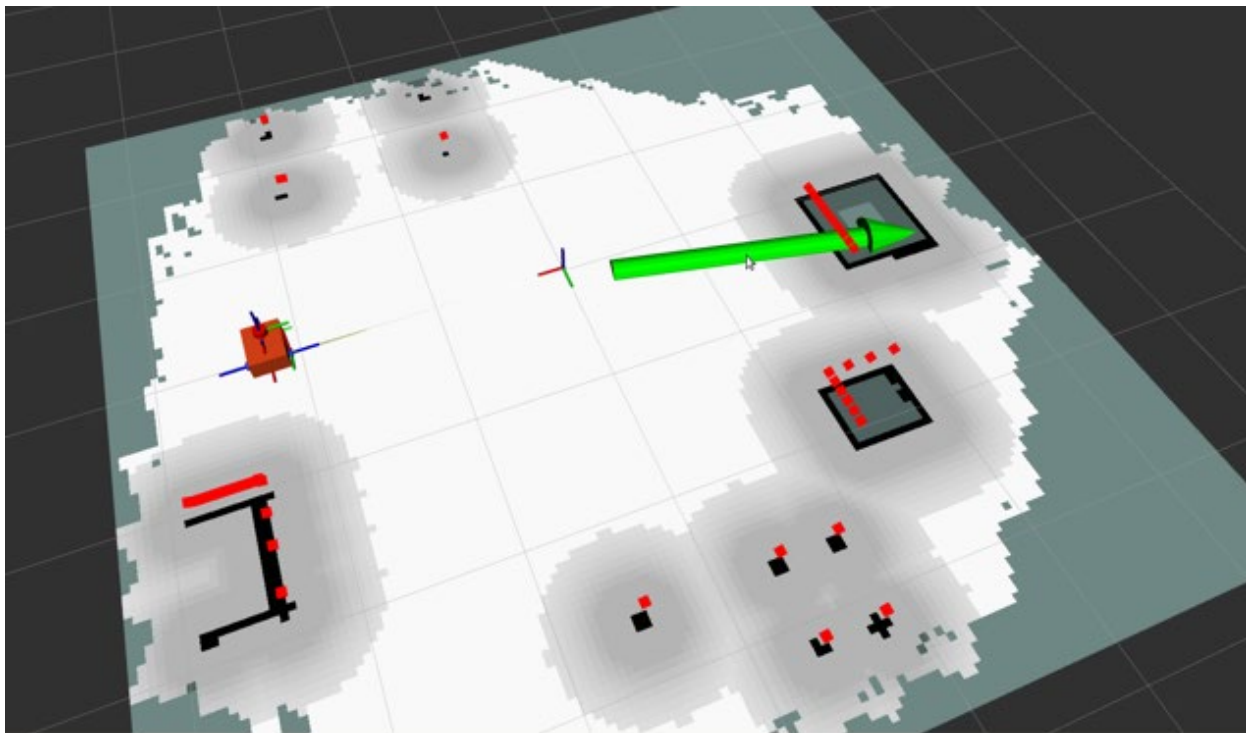
2.8 суретте жұмыс барысында SLAM келесі негізгі тақырыптар мен хабарламаларды пайдаланады:

Тақырып карта (`nav_msgs/OccupancyGrid`): бұл тақырып SLAM алгоритмі арқылы жасалған орта картасын жариялайды. Карта тор ретінде ұсынылған, мұнда әрбір мән ұяшық өтуге болатынын, кедергінің бар-жоғын немесе белгісіз екенін көрсетеді.

Тақырып odom (`nav_msgs/Odometry`): Бұл тақырып роботтың қозғалысын қадағалау үшін пайдаланылатын одометрия деректерін жібереді. Одометрия жүріп өткен қашықтық пен айналу бұрышы туралы ақпаратты береді, бұл роботтың салыстырмалы орнын анықтауға көмектеседі.

tf (Трансформация ағашы): tf робот негізі, LiDAR, камера және басқа сенсорлар сияқты әртүрлі робот координат жүйелері арасындағы түрлендірулерді байланыстырады. Бұл ортақ координаттар жүйесінде әртүрлі сенсорлардан алынған деректерді дұрыс пайдалану үшін қажет.

Тақырып сканерлеу (`sensor_msgs/LaserScan`): Бұл тақырып LiDAR деректерін алады және картаны құру және роботты локализациялау үшін SLAM алгоритмімен пайдаланылады.[8]



2.8-сурет – Робот жүйесін дайындау

SLAM алгоритмі қоршаған ортаның картасын біртіндеп құру үшін одометрия мен LiDAR деректерін пайдалану арқылы жұмыс істейді. Ол сондай-ақ роботтың орнын дәлірек анықтау үшін сенсор деректерін пайдалану арқылы одометрия қателерін түзетеді.[9]

LiDAR және SLAM көмегімен робот навигациясы бірнеше негізгі қадамдарды қамтиды:

SLAM алгоритмі нақты уақытта қоршаған орта картасын құру үшін LiDAR және одометрия деректерін пайдаланады. Бұл карта робот жаңа аумақтарды зерттеген сайын жаңартылады.

Құрылған карта мен ағымдағы сенсор деректерін пайдалана отырып, SLAM роботтың картадағы ағымдағы орнын анықтайды.

ROS 2 жүйесіндегі Nav2 сияқты навигациялық жүйе берілген межелі жерге оңтайлы бағытты есептеу үшін картаны және роботтың ағымдағы орнын пайдаланады. Робот қозғалған кезде жаңа кедергілер туралы LiDAR деректеріне негізделген өз жолын бейімдейді. Робот қозғалған кезде ол өз жолындағы кедергілерді анықтау және нақты уақытта өз бағытын реттеу үшін LiDAR деректерін пайдаланады.

2.5 Бір роботтың іздеу тиімділігін есептеу

Көпагентті робототехникалық жүйенің тиімділігін арттыру үшін бірқатар есептеулер жүргізілді. Есептеулер роботтың қозғалысы, энергия шығыны және сенсорлық жүйесінің тиімділігін бағалауға бағытталды.

Роботтың іздеу аймағындағы жұмысының тиімділігін есептеу

Робот біртекті $S_{\text{жалпы}}=120 \text{ м}^2$ іздеу аймағында жұмыс істейді. Робот жылдамдығы $v = 1.5 \text{ м/с}$ Сенсордың қамту радиусы $r=2 \text{ м}$. Роботтың іздеу тиімділігі η энергия шығынымен және іздеу уақытымен байланысты.

Робот сенсорының бір мезетте қамтитын аймағы (2.1) формулада S сенсор шеңбердің ауданына тең:

$$S_{\text{сенсор}} = \pi * r^2 \quad (2.1)$$

Есептейміз:

$$S_{\text{сенсор}} = \pi * r^2 = \pi * 2^2 = 4\pi \approx 12,57 \text{ м}^2$$

Жалпы іздеу уақытын есептеу

Роботтың жылдамдығы мен сенсордың қамту аймағын ескеріп, іздеу уақыты (2.2) формулада $T_{\text{жалпы}}$ есептеледі:

$$T_{\text{жалпы}} = \frac{S_{\text{жалпы}}}{S_{\text{сенсор}} * v} = \frac{120}{12,57 * 1,5} = 6,36 \text{ с} \quad (2.2)$$

Энергия шығынын есептеу

Роботтың іздеу кезінде тұтынатын жалпы энергиясы (2.3) формулада $E_{\text{жалпы}}$:

$$E_{\text{жалпы}} = (U_{\text{мотор}} * T_{\text{жалпы}}) + (U_{\text{сенсор}} * T_{\text{жалпы}}) \quad (2.3)$$

Берілген:

$$U_{\text{мотор}} = 20 \text{ ВТ}$$

$$U_{\text{сенсор}} = 5 \text{ ВТ}$$

Есептейміз:

$$E_{\text{жалпы}} = (20 + 5) * 6,36 = 25 * 6,36 \approx 159 \text{ Дж}$$

Іздеу тиімділігін есептеу

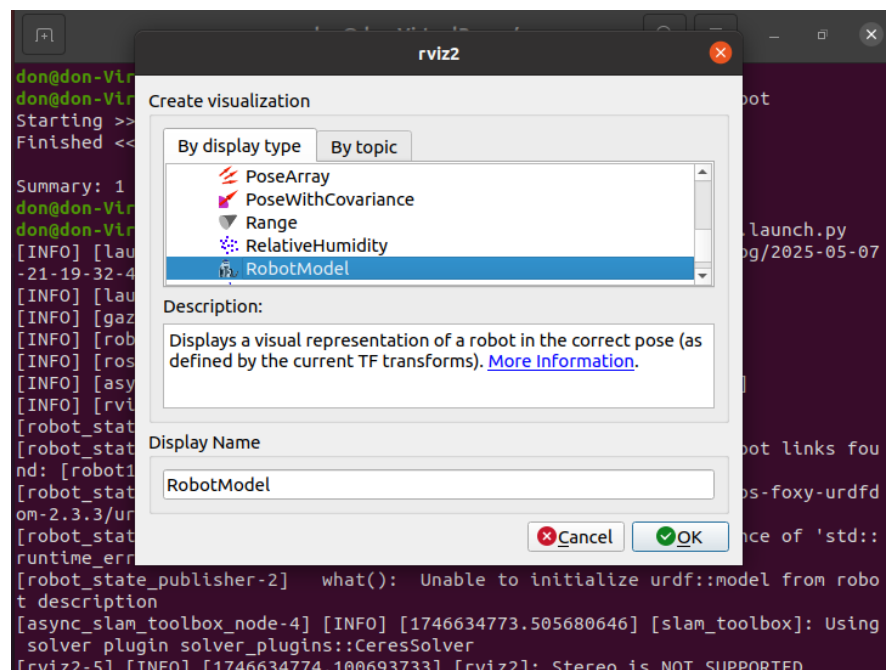
Іздеу тиімділігі η (2.4) формулада роботтың сенсор ауданы мен жалпы іздеу аймағының қатынасымен анықталады:

$$\eta = \frac{S_{\text{сенсор}}}{S_{\text{жалпы}}} * 100\% = \frac{12,57}{120} * 100\% = 10,48\% \quad (2.4)$$

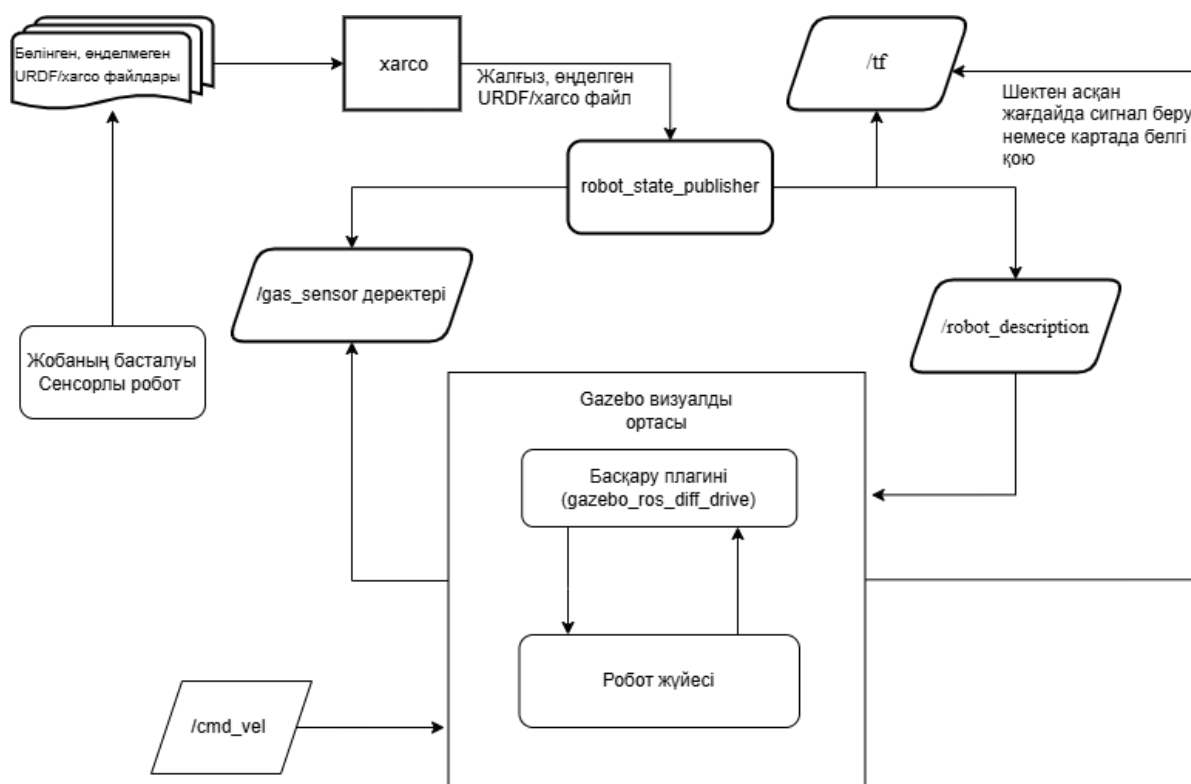
3. Жүйелік дизайн

3.1 Жүйе архитектурасы

2.9 суретте «launch_sim.launch.py» файлы Gazebo жүйесінде робот модельдеуін іске қосу үшін қажетті негізгі құрамдастарды біріктіру және үйлестіру үшін қызмет етеді. Бұл процесс қоршаған ортаны дайындау, робот үлгісін инициализациялау және модельдеу мен ROS арасындағы уақыт синхрондауды қамтамасыз ету сияқты бірнеше маңызды қадамдарды қамтиды.



2.9-сурет – Робот моделін таңдау



2.10-сурет – Сенсорлы робот жүйесінің функционалдық схемасы

2.10 суретке сәйкес, бірінші қадам URDF деректерін пайдаланып роботтың барлық қозғалатын бөліктерінің түрлендірулерін жариялаудың маңызды функциясын орындайтын «robot_state_publisher» қосу. use_sim_time='true' параметрі жүйеге симулятордың уақыт белгілерін пайдалануға мүмкіндік береді, бұл жүйенің әртүрлі құрамдас бөліктері

арасындағы дұрыс әрекеттесуді қамтамасыз ете отырып, Gazebo-дағы виртуалды уақытты ROS-пен синхрондау үшін қажет.

Келесі қадам «gazebo.launch.py» сценарийі арқылы Gazebo модельдеу ортасын іске қосу болып табылады. Бұл қадам роботты әртүрлі жағдайларда сынауға болатын виртуалды кеңістікті жасайды. Гравитация, үйкеліс және соқтығыстар сияқты физика модельдеулерін қосу роботтың әртүрлі сценарийлердегі әрекеті туралы сенімді деректерді алуға мүмкіндік береді.

Ортаны дайындаған соң, робот үлгісі «spawn_entity.py» сценарийі арқылы жасалады. URDF-те сипатталған және «robot_description» тақырыптарынан өткен робот үлгісі Gazebo ортасында орналастырылған. Бұл роботқа модельдеу ортасымен әрекеттесуге және тағайындалған тапсырмаларды орындауға мүмкіндік береді.

«rsp.launch.py» файлы арқылы конфигурациялау және іске қосу Xacro-дан жасалған URDF файлын өңдеуді қамтиды. Бұл қадам модельдеу үшін роботтың егжей-тегжейлі және дәл үлгісін жасау үшін қажет. Xacro файлы URDF форматына түрлендірілді, ол роботтың барлық аспектілерін, соның ішінде оның физикалық құрылымын, сенсорлары мен жетектерін сипаттауға арналған стандартты пішім болып табылады. Әрі қарай, бұл URDF файлы навигациялық жүйенің дұрыс жұмыс істеуі және роботтың қоршаған ортамен әрекеттесуі үшін маңызды болып табылатын бірлескен күйлер мен сәйкес түрлендірулердің жариялануын қамтамасыз ететін «robot_state_publisher» конфигурациялау үшін пайдаланылады.

Сонымен қатар, «gazebo_control.xacro» файлы роботтың дифференциалды дискісін басқаруға жауапты Gazebo плагинінің конфигурациясын анықтайды. «libgazebo_ros_diff_drive.so» плагині дөңгелектер арасындағы қашықтық, олардың диаметрі, максималды айналу моменті және үдеу сияқты параметрлерді қоса, робот дөңгелектерін басқару үшін конфигурацияланған. Бұл параметрлер роботтың қозғалыс әрекетін дәл модельдеу үшін өте маңызды.

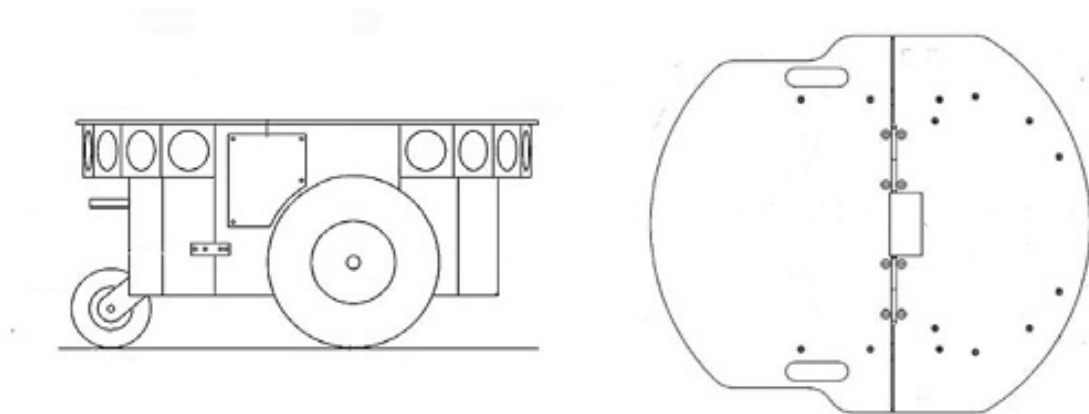
Сондай-ақ плагин роботтың орны мен қозғалысы туралы деректерді «odometry_frame» және «robot_base_frame» сияқты элементтер арқылы жариялайды, бұл навигация және бақылау жүйелеріне роботтың қозғалысын Gazebo модельдеу ортасында дәл бақылауға мүмкіндік береді.[6]

3.2 Роботтың дизайны

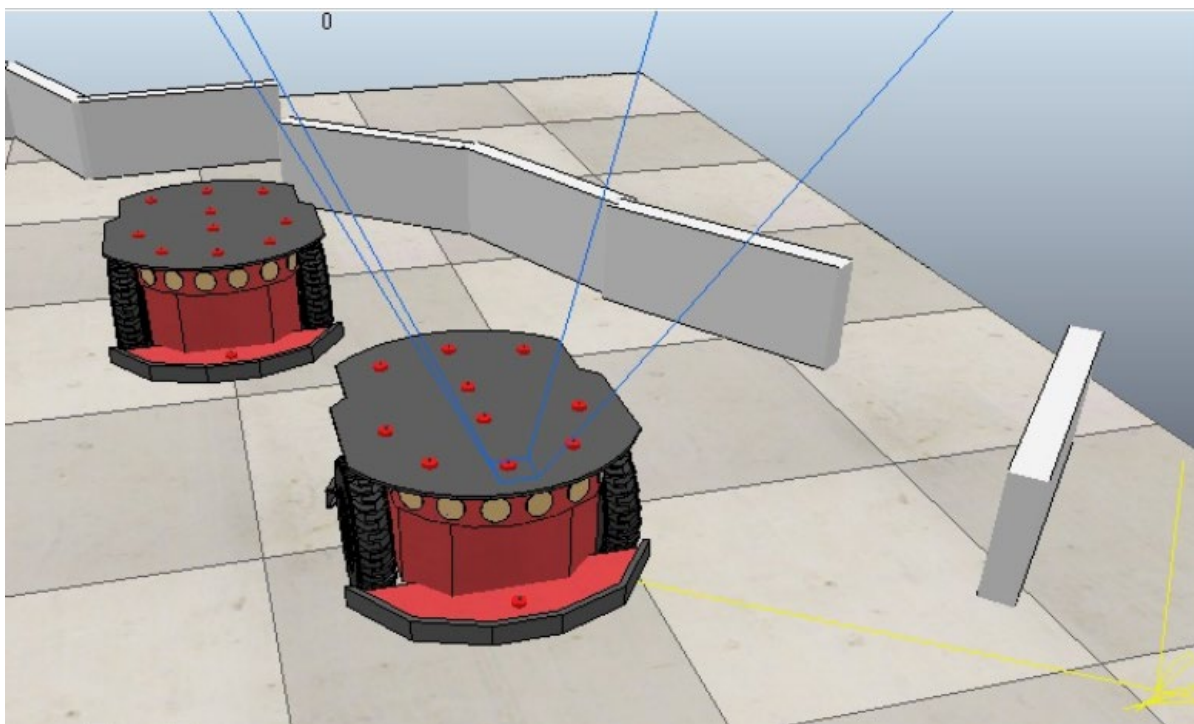
3.1 суретте «robot_core.xacro» файлында робот дизайны мен геометриясының негізгі анықтамалары бар. Ол шасси мен дөңгелектер сияқты негізгі компоненттердің өлшемдерін, салмағын және материалдарын көрсетеді. Бұл файл сонымен қатар шассидің өлшемдері мен салмағын, дөңгелектердің радиусы мен қалыңдығын және олардың негізгі корпусқа қатысты орналасуын сипаттайтын параметрлерді көрсетеді. Газебо және RViz симуляторларында роботты визуализациялау үшін түстер сияқты көрнекі атрибуттар қолданылады. Файлдың маңызды аспектісі буындар мен

буындардың сипаттамасы болып табылады роботтың әртүрлі бөліктері арасындағы қозғалмайтын және жылжымалы байланыстар.[3]

```
<link name="base_link">
</link>
<joint name="base_footprint_joint" type="fixed">
<parent link="base_link"/>
<child link="base_footprint"/>
<origin xyz="0 0 0" rpy="0 0 0"/>
</joint>
<link name="base_footprint">
</link>
<link name="chassis">
<visual>
<origin xyz="{chassis_length/2} 0 {chassis_height/2}"/>
<geometry>
<box size="{chassis_length} {chassis_width} {chassis_height}"/>
</geometry>
<material name="orange"/>
</visual>
</link>
```



3.1-сурет – Робот схемасы



3.2-сурет– CoppeliaSim графикалық интерфейсінде роботты модельдеу

3.2 суретте «inertial_macros.xacro» файлы физикалық дәл модельдеу үшін қажетті робот құрамдастарының инерциялық қасиеттерін анықтауға арналған макростарды қамтамасыз етеді. Макростар шарлар, цилиндрлер және тікбұрышты параллелепипедтер сияқты әртүрлі геометриялық фигуралар үшін инерция моменттерін есептеуге арналған формулаларды қамтиды. Бұл деректер «robot_core.xacro» негізгі файлында шассидің, дөңгелектердің және басқа компоненттердің инерциялық сипаттамаларын орнату үшін пайдаланылады, бұл роботтың физикалық қасиеттерінің Gazebo жүйесінде дұрыс үлгіленуін қамтамасыз етеді.

```
<xacro:macro name="inertial_sphere" params="mass radius *origin">
<inertial>
<xacro:insert_block name="origin"/>
<mass value="\${mass}" />
<inertia ixx="\${(2/5) * mass * (radius*radius)}" ixy="0.0" ixz="0.0"
iyy="\${(2/5) * mass * (radius*radius)}" iyz="0.0"
izz="\${(2/5) * mass * (radius*radius)}" />
</inertial>
</xacro:macro>
```

'lidar.xacro' файлы оның физикалық және сенсорлық сипаттамаларын қоса, роботқа lidar конфигурациясын және орнатуын сипаттайды. Бұл файл лидар робот шассііне қалай орнатылатынын, сондай-ақ оның орналасуы мен бағытын көрсетеді. Сонымен қатар, файл газебодағы дұрыс визуализация және соқтығысты өңдеу үшін lidar визуалды және соқтығыс модельдерін

анықтайды. Сондай-ақ, сканерлеу параметрлерін, жаңарту жылдамдығын және сәуленің параметрлерін анықтайтын Gazebo lidar сенсорының параметрлері кіреді. Бұған қоса, сканерлеу деректерін жариялау үшін сенсорды ROS тақырыптарына байланыстыратын ROS плагині конфигурацияланған.

```
<joint name="laser_joint" type="fixed">
<parent link="chassis"/>
<child link="laser_frame"/>
<origin xyz="0.1 0 0.175" rpy="0 0 0"/>
</joint>
<sensor name="laser" type="ray">
<visualize>true</visualize>
<update_rate>10</update_rate>
<ray>
<scan>
<horizontal>
<samples>360</samples>
<min_angle>-3.14</min_angle>
<max_angle>3.14</max_angle>
</horizontal>
</scan>
<range>
<min>0.3</min>
<max>12</max>
</range>
</ray>
<plugin name="laser_controller" filename="libgazebo_ros_ray_sensor.so">
<ros>
<argument>~/out:=scan</argument>
</ros>
<output_type>sensor_msgs/LaserScan</output_type>
<frame_name>laser_frame</frame_name>
</plugin>
</sensor>
```

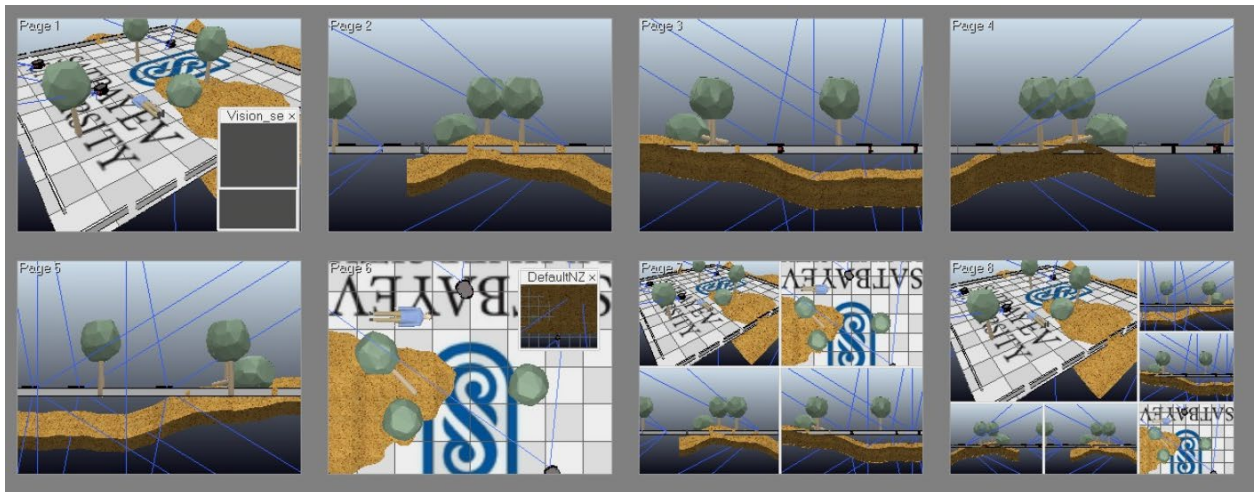
4. Жүйені енгізу

4.1 SLAM орнату және іске қосу

ROS2 жүйесінде SLAM Toolbox және 2D LiDAR көмегімен SLAM енгізу үшін бір уақытта локализация және карталау жүйесінің дұрыс жұмысын қамтамасыз ету үшін тақырыптарды, түйіндерді және параметрлерді дұрыс конфигурациялау қажет. Әрі қарай, бұл іс жүзінде қалай жүзеге асырылатынын қарастырайық.

SLAM құралдар жинағын 2D LiDAR көмегімен пайдалану кезіндегі негізгі компоненттер мен тақырыптар. /сканерлеу тақырыбы 2D LiDAR деректерін тасымалдау үшін пайдаланылады. Бұл қоршаған ортадағы кедергілер мен нысандар туралы ақпаратты көрсететін деректер орналастырылатын тақырып. /tf тақырыбы odom, base_link және map сияқты әртүрлі кадрлар арасындағы түрлендірулерді жариялауға жауапты. Бұл түрлендірулер LiDAR және басқа сенсор деректерін робот пен картаға қатысты дұрыс орналастыру үшін қажет. /odom тақырыбы дөңгалақ кодерлерінің деректері сияқты одометрия деректеріне негізделген роботтың уақыт бойынша қозғалысы туралы ақпаратты береді.

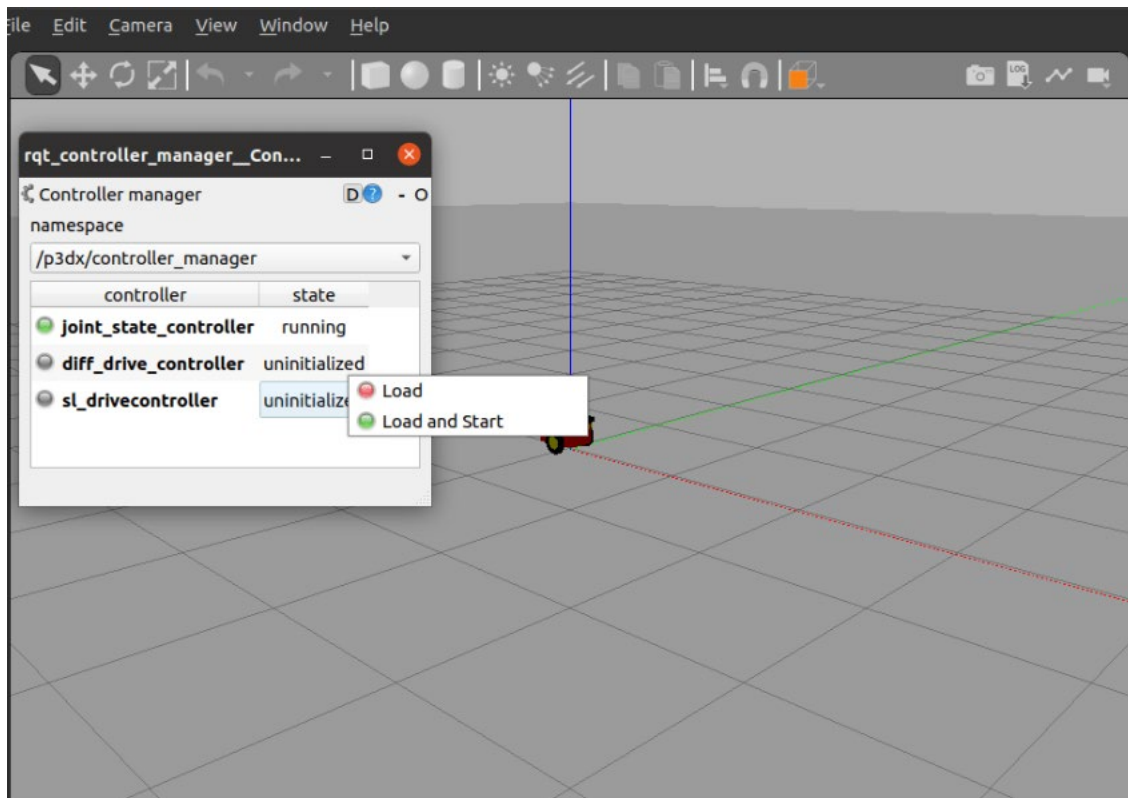
Карта жақтауы қоршаған орта картасы жасалатын ғаламдық координаттар жүйесі. LiDAR және басқа сенсорлардан келетін барлық деректер осы координаттар жүйесіне сілтеме жасайды. Одом кадры қысқа мерзімде роботтың қозғалысын қадағалау үшін жергілікті координаттар жүйесі ретінде пайдаланылады. base_link жақтауы робот негізімен байланысты және әдетте роботтың ортасында немесе оның дөңгелектерінің арасында орналасады. slam_toolbox түйіні SLAM алгоритмдерін жүзеге асырады. Асинхронды режимде бұл түйін нақты уақытта LiDAR деректерін өңдейді, карта жасайды және роботты ондағы локализациялайды. Robot_state_publisher түйіні картаға қатысты LiDAR деректерін дұрыс орналастыру үшін қажет әртүрлі робот кадрлары арасындағы түрлендірулерді жариялайды.[4]



4.1-сурет – CorreliaSim графикалық интерфейсі «Симуляция Алаңы»

4.1 суретте, SLAM құралдар жинағы картаны қалай салады және картадағы орынды анықтайды. SLAM Toolbox 2D LiDAR деректерін сканерлеу тақырыбы арқылы алады. Бұл деректер роботтың ортасындағы ең жақын кедергілерге дейінгі қашықтықты көрсететін сәулелер жиынтығы. odom тақырыбынан LiDAR және одометрия деректеріне негізделген SLAM Toolbox нақты уақытта картаны жаңартады. Әрбір жұмыс циклі сканерлеуді сәйкестендіру алгоритмін пайдалануды қамтиды, ол ағымдағы LiDAR

деректерін бұрыннан жасалған картамен сәйкестендіруге, роботтың орнын реттеуге және қателерді азайтуға көмектеседі.



4.2-сурет – Gazebo графикалық ортада роботты баптау

4.2 суретке сәйкес, локализация ағымдағы сканерлеуді бар картамен салыстыру арқылы жүзеге асырылады. Робот қозғалған кезде SLAM Toolbox LiDAR және одометрия деректері негізінде картаға қатысты орнындағы өзгерістерді анықтайды. Бұрыннан өтіп кеткен карта бөлімдеріне сәйкес келетін сәйкестіктер табылса, жинақталған қателерді азайту және карта мен роботтың траекториясын түзету үшін циклды жабу процедурасы қолданылады. Карта нақты уақыт режимінде жаңа деректер алынған кезде жаңартылады. Картаны жаңарту аралықтарын конфигурация файлындағы `map_update_interval` параметрі арқылы реттеуге болады. Карта торлы карта ретінде сақталады, мұнда әрбір ұяшық кедергі болуының белгілі бір ықтималдығы бар кеңістік аймағын көрсетеді(сурет 4.3).

SLAM Toolbox бағдарламасын 2D LiDAR көмегімен іске қосу үшін жоғарыда аталған барлық параметрлерді анықтайтын `mapper_params_online_async.yaml` конфигурация файлын жасау керек. Негізгі конфигурацияның мысалы келесі параметрлерді қамтуы мүмкін:

`mode: mapping`: SLAM Toolbox тек локализациялап қана қоймай, картаны нақты уақытта жаңартып отыратын жұмыс режимі.

`resolution: 0.05`: Карта ажыратымдылығы, мұндағы 0,05 мәні әрбір тор ұяшығы 5 сантиметрге сәйкес келетінін білдіреді.

map_update_interval: 2.0: Картаны жаңарту аралығы, әрбір 2 секунд сайын.

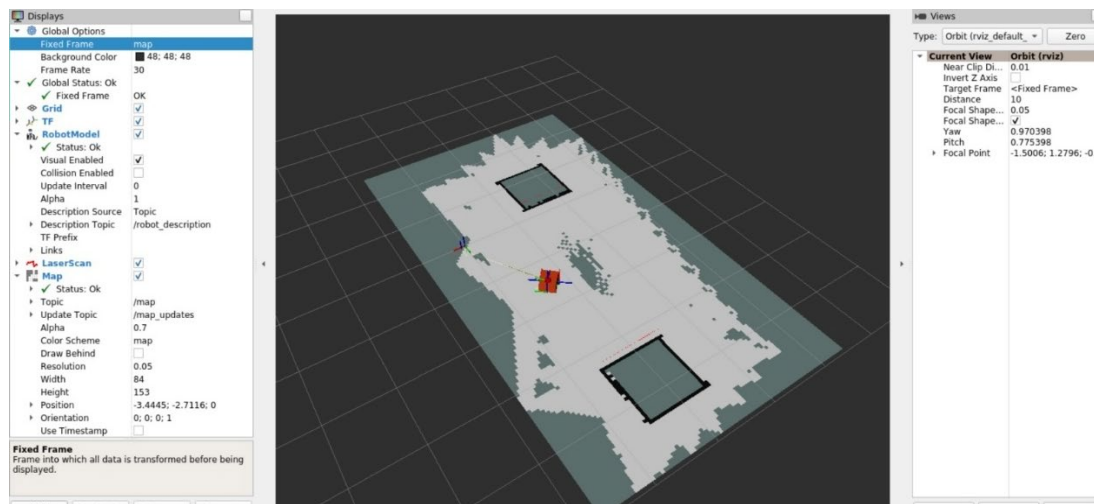
use_scan_matching: true: локализацияны жақсарту үшін сканерлеуге сәйкес алгоритмдерді қосады.

scan_topic: /scan: LiDAR деректері алынатын тақырып.

map_frame, odom_frame, base_frame: Бұл параметрлер сәйкесінше карта, одометрия және робот негізі үшін координат жүйелерін көрсетеді.

Осы файлмен SLAM Toolbox іске қосу пәрменімен орындалады:

```
ros2 iске қосу slam_toolbox online_async_launch.py
params_file:=./src/my_bot/config/mapper_params_online_async.yaml
use_sim_time:=true
```



4.3-сурет– Rviz интерфейсіндегі көрініс

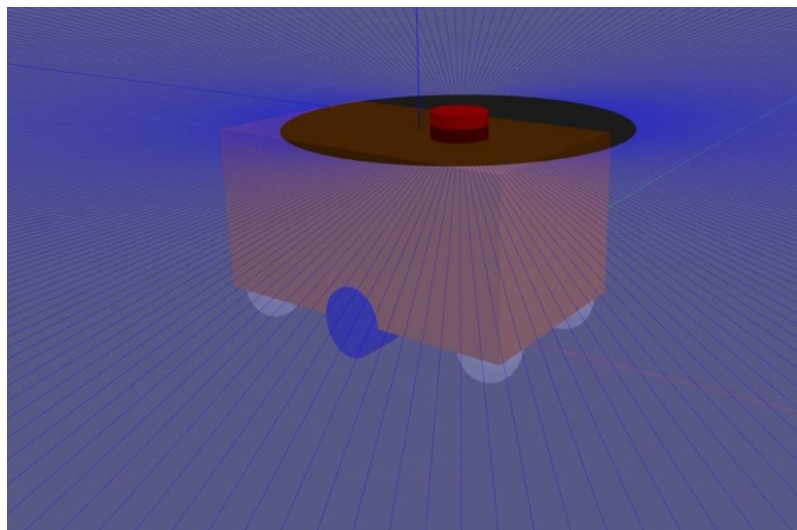
4.3 суретте көрсетілгендей, SLAM Toolbox-ты жоғарыда көрсетілген параметрлермен асинхронды режимде іске қосқаннан кейін жүйенің карталау және локализация тапсырмасын қалай орындайтынын тексеру қажет. RViz-тегі визуализация картаны жасау процесін нақты уақыт режимінде көруге, сондай-ақ роботтың қозғалысын қадағалауға және оның картадағы орнын жаңартуға мүмкіндік береді. RViz пайдалану жүйе өнімділігін тексеруге және реттеуге көмектесу үшін картаны, робот траекториясын, LiDAR деректерін және түрлендірулерді (TF) көрсетуді қамтиды.

4.2 Офлайн шарлау үшін Nav2-мен интеграция

Nav2 роботқа маршрутты жоспарлау, маршрутты орындау, кедергілерді болдырмау және қателерді қалпына келтіру сияқты функцияларды жүзеге асыра отырып, қоршаған ортада автономды шарлауға мүмкіндік береді. Бұл жүйе роботтардың әртүрлі түрлерімен және әртүрлі сценарийлерде жұмыс істеу үшін конфигурациялауға мүмкіндік беретін икемділікті қамтамасыз етеді. `navigation\_launch.py` файлы Nav2 бағдарламасының барлық қажетті құрамдастарын, соның ішінде маршруттауды, трафикті басқаруды және қателерді өңдеуді іске қосуды басқарады. Ол ROS2 Launch Python API арқылы

құрылымдалған және қажетті модульдерді импорттауды, параметрлерді жүктеуді және түйіннің өмірлік циклін басқаруды қамтиды.

«use_sim_time» параметрі Gazebo сияқты симуляторлармен синхрондау үшін модельдеу уақытын пайдалануды көрсетеді. Бұл виртуалды ортада навигациялық жүйенің дұрыс жұмыс істеуі үшін өте маңызды. «params_file» параметрі барлық Nav2 түйіндері үшін параметрлерді қамтитын YAML файлына жолды көрсетеді. Бұл файл навигациялық жүйенің әрекетін, соның ішінде маршрутты жоспарлаушы мен контроллер параметрлерін басқарады. «default_bt_xml_filename» параметрі навигация шешімдерін қабылдау үшін Nav2 пайдаланатын мінез-құлық ағашын сипаттайтын XML файлының жолын көрсетеді. «map_subscribe_transient_local» параметрі карта жазылымдарының «өтпелі жергілікті» болып табылатынын конфигурациялайды, бұл түйіндерге ең соңғы жарияланған карта деректерін тіпті жазылым алдында жарияланған болса да сақтауға мүмкіндік береді.[5]



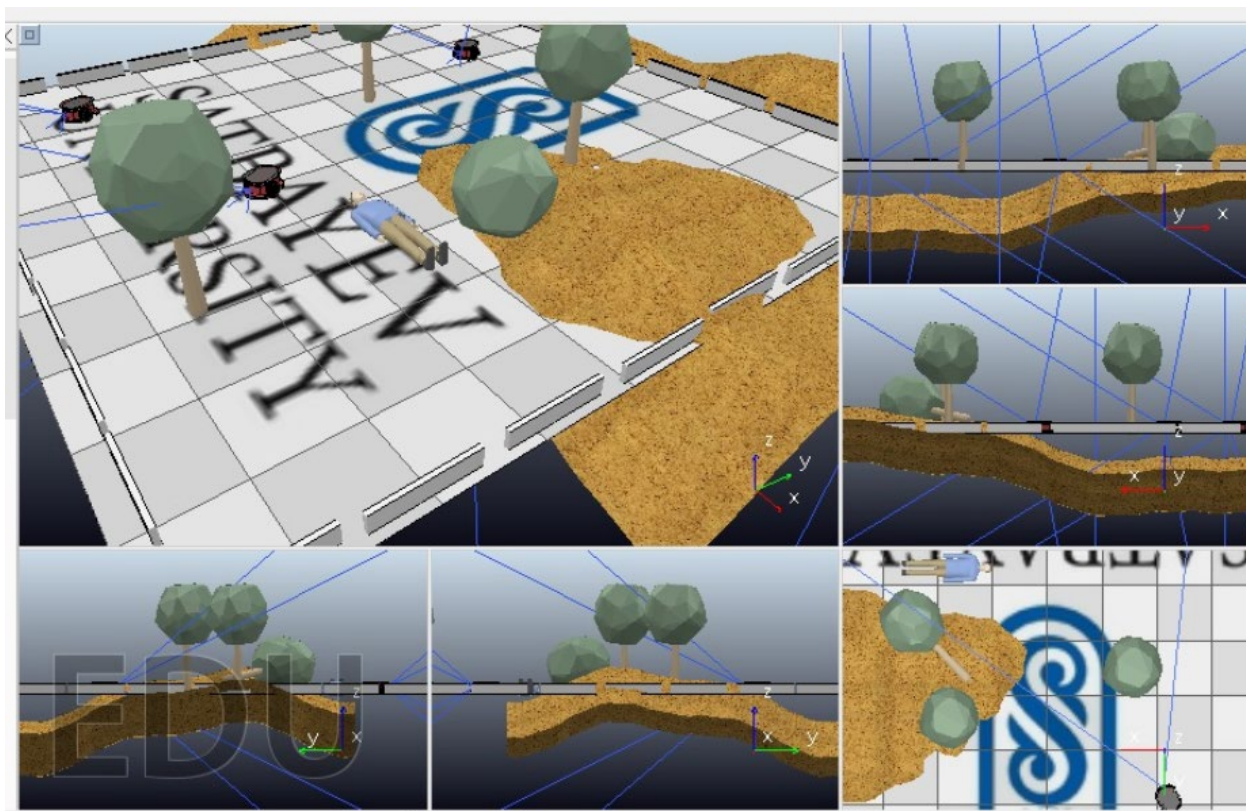
4.4-сурет – Gazebo графикалық ортада камерамен көрініс

«navigation_launch.py» файлы бірнеше негізгі түйіндерді іске қосады, олардың әрқайсысы шарлау процесінде белгілі бір рөл атқарады.

«nav2_controller» түйіні роботтың төменгі деңгейлі контроллеріне жіберілетін жылдамдық пәрмендерін жасау арқылы алдын ала жоспарланған маршрут бойынша роботтың қозғалысын басқарады. Бұл түйін маршруттың дәл жүруін қамтамасыз ету үшін роботтың ағымдағы орнын, бағдарын және сенсор деректерін ескереді. Мысалы, роботтың жолында кедергі пайда болса, «nav2_controller» соқтығысуды болдырмау үшін сенсор деректеріне негізделген жолды реттей алады.

4.4 суретте «nav2_planner» түйіні роботтың ағымдағы орнынан картадағы берілген мақсатқа дейінгі жаһандық бағытты жоспарлауға жауап береді. Бұл түйін картадағы белгілі кедергілерді болдырмай оңтайлы бағытты табу үшін A* немесе Dijkstra сияқты жол іздеу алгоритмдерін пайдаланады. Мысалы, роботты картаның белгілі бір нүктесіне жіберу қажет болса,

«nav2_planner» барлық белгілі статикалық кедергілерді ескере отырып, маршрутты есептейді.



4.5-сурет – CoppeliaSim графикалық ортада жүйелік көрініс

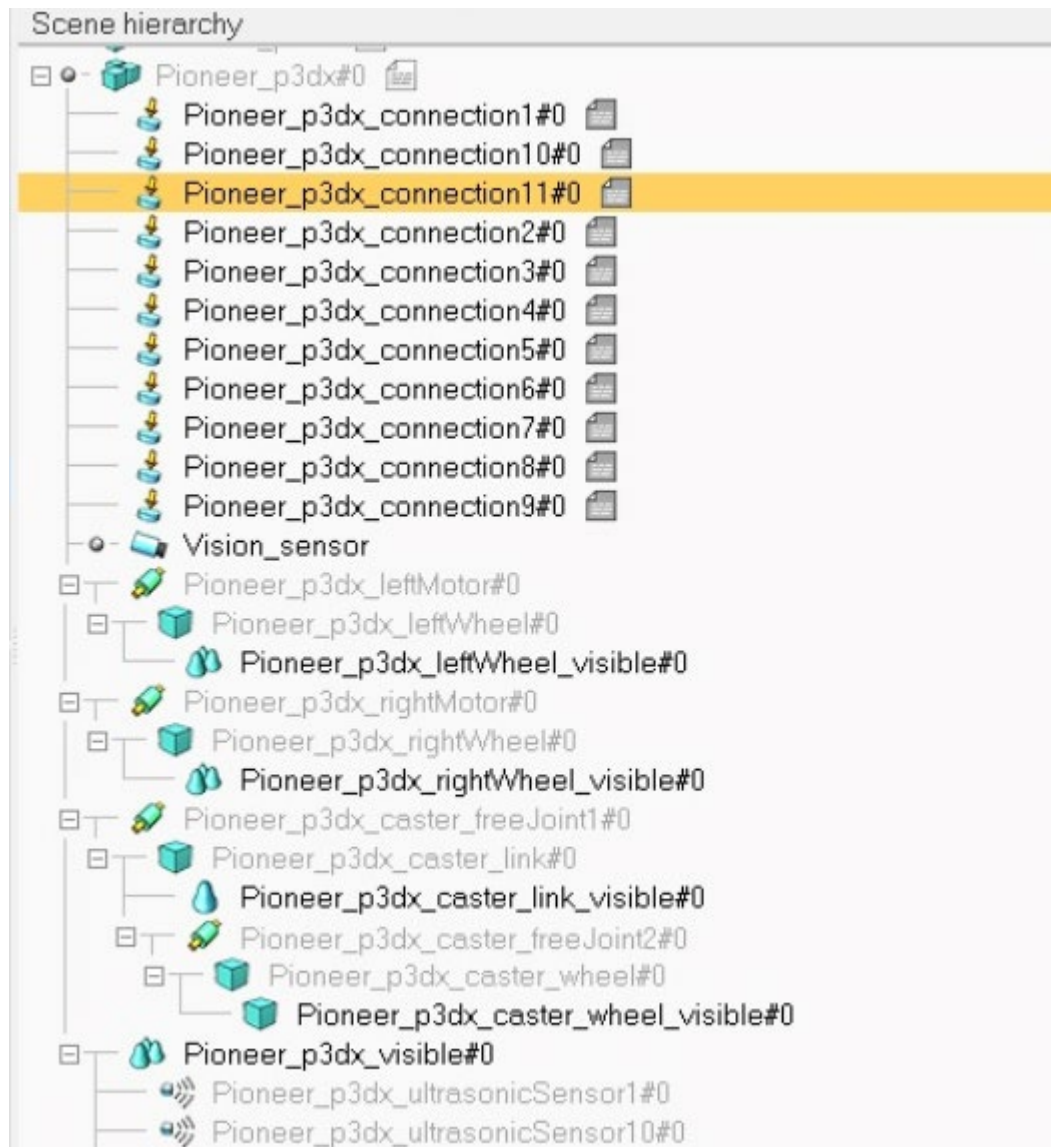
«nav2_recoveries» түйіні робот бұрышта тұрып қалу немесе қозғалуды жалғастыра алмау сияқты қиындықтарға тап болған жағдайда қалпына келтіру стратегияларын басқарады. Бұл түйін бұрылу, бұрын есептелген маршрутқа оралуға әрекет жасау немесе жаңа маршрут құру сияқты қалпына келтіру процедураларын бастай алады. Мысалы, робот еңсерілмейтін кедергіге тап болса, түйін баламалы жолды табу үшін бірқатар әрекеттерді бастайды.

4.5 суретте, «nav2_bt_navigator» түйіні робот әрекеттерінің ретін басқаратын мінез-құлық ағашын жүзеге асырады. Мінез-құлық ағашы роботтың маршрутты қашан және қалай жоспарлауы, оны орындауы және сәтсіздіктерден қалпына келтіруі керек екенін анықтайды. Мысалы, мінез-құлық ағашы алдымен маршрутты есептеу үшін жаһандық жоспарлаушыны іске қосады, содан кейін қозғалыс үшін басқаруды контроллерге тапсырады және кедергі анықталса, қалпына келтіру процедураларын бастайды.

«nav2_waypoint_follower» түйіні роботты алдын ала орнатылған нүктелер бойынша қадағалауға жауапты. Бұл түйін робот патрульдеу сияқты алдын ала белгіленген маршрутпен жүруі керек тапсырмалар үшін пайдалы. Мысалы, робот деректер жинау немесе кедергілерді болдырмау сияқты әрбір жол нүктесіне ретімен кіріп, әрқайсысында қажетті әрекеттерді орындайды.

«nav2_lifecycle_manager» түйіні барлық Nav2 түйіндерінің өмірлік циклін басқарады, олардың дұрыс инициализациялануын және өшірілуін

камтамасыз етеді. Бұл түйін конфигурация, белсендіру және өшіру сияқты түйін күйлері арасындағы ауысуларды үйлестіреді. Мысалы, барлық шарлау түйіндерін бастамас бұрын, «nav2_lifecycle_manager» барлық құрамдастардың іске қосылуға дайын екеніне және дұрыс инициализацияланғанына көз жеткізеді(сурет 4.5).[2]



4.6-сурет– CorreliaSim графикалық ортада роботтар деректері

Nav2 роботты жаһандық және жергілікті жоспарлау комбинациясы арқылы басқарады.

4.6 суретте, ғаламдық жоспарлау (nav2_planner түйіні): Картадағы нүкте сияқты мақсат көрсетілгенде, «nav2_planner» роботтың ағымдағы орнынан сол мақсатқа дейінгі бағытты есептейді. Көршілес картаны пайдалана отырып, түйін белгілі кедергілерді болдырмайтын жолды таңдайды және ең аз ұзындық немесе саяхат уақыты сияқты берілген критерийлерге сәйкес маршрутты оңтайландырады. Нәтиже робот үшін қозғалыс командаларын генерациялайтын контроллерге берілетін маршрут.[3]

Жергілікті басқару (nav2_controller түйіні): Бұл түйін ғаламдық бағытты қабылдайды және lidar немесе камера сияқты сенсорлардың ағымдағы деректерін ескере отырып, оны жылдамдық пәрмендеріне түрлендіреді. Контроллер жақын маңдағы кедергілерді бағалайды және соқтығысуды болдырмау және маршрутты дәл орындау үшін роботтың қозғалысын реттейді. Қозғалыс пәрмендері роботтың қозғалтқыштарын басқаратын «cmd_vel» тақырыбы арқылы роботтың төменгі деңгейлі контроллеріне жіберіледі.

Кедергілерді болдырмау және қалпына келтіру (nav2_recoveries түйіні): Егер робот қозғалу кезінде кедергіге тап болса, стандартты болдырмау әдістері сәтсіз болған жағдайда nav2_controller қалпына келтіру процедурасын орындау үшін сигнал жібере алады. «nav2_recoveries» түйіні маршрутқа қайта кіру әрекеті немесе маршрутты толығымен қайта есептеу сияқты процедураларды орындайды. Нәтижесінде робот не жаңа бағыт бойынша қозғалуды жалғастырады немесе бастапқыда жоспарланған жолды қайталауға тырысады.

Nav2 іске қосу үшін пәрменді пайдалану:

```
nav2_bringup navigation_launch.py use_sim_time:=true
```

Бұл пәрмен модельдеу уақытын пайдаланып шарлау жүйесін инициализациялайды және `navigation\_launch.py` ішінде көрсетілген файлдардан параметрлер мен параметрлерді жүктейді. Бұл Nav2 симулятормен дұрыс синхрондалғанын және барлық түйіндердің конфигурацияланғанын және дұрыс жұмыс істейтінін қамтамасыз етеді. Навигациялық жүйенің күйін бақылау және туындаған мәселелерді шешу үшін мыналарды пайдалану ұсынылады:

RViz: Бұл құрал картаны, траекторияны, сенсор деректерін және түрлендірулерді визуализациялауға көмектеседі, бұл Nav2 маршрутты қалай жоспарлайтынын және роботты нақты уақытта қалай басқаратынын түсінуге мүмкіндік береді.

Журналдар мен тақырыптар: «/cmd_vel», «/tf», «/odom» және «/scan» сияқты тақырыптарды бақылау проблемаларды диагностикалауға және Nav2 алгоритмдерінің жұмысын тексеруге көмектеседі.

ROS2 CLI: Түйін күйлерін басқару және жүйе параметрлерін шығарып алу үшін «ros2 өмірлік циклі», «ros2 тақырыбы жаңғырығы» және «ros2 param» пәрмендерін пайдалану.[9]

4.3 Басқару коды және деректерді ұйымдастыру

«launch_sim.launch.py» файлы Gazebo симуляторы арқылы ROS2 жүйесінде робот модельдеуін инициализациялау үшін пайдаланылады. Ол робот күйлерін жариялау, қозғалысты басқару және модельдеумен өзара әрекеттесуді орнату сияқты негізгі басқару түйіндерін қосады. Бұл файл ROS2

Control және Gazebo интеграциясын қолдайтын роботты сынау және жөндеу үшін виртуалды ортаны құру процесінің маңызды бөлігі болып табылады.

"robot_state_publisher" функциясы роботтың URDF үлгісіне негізделген робот буындарының күйі (мысалы, дөңгелектің айналуы немесе қолдың орналасуы) туралы деректерді жариялайды. Бұл роботты модельдеу және визуализация үшін маңызды, себебі ол оның ағымдағы күйін RViz және Gazebo сияқты құралдарда көрсетуге мүмкіндік береді.[9]

Параметрлер:

«use_sim_time»: модельдеу уақытын қосу («true») барлық процестерді Gazebo симуляторымен синхрондауға мүмкіндік береді.

«use_ros2_control»: роботты ROS2 басқару арқылы басқаруға мүмкіндік береді, ол қозғалыс пәрмендерін орындау үшін қажет.

```
rsp = IncludeLaunchDescription(  
  PythonLaunchDescriptionSource([os.path.join(  
    get_package_share_directory(package_name), 'launch', 'rsp.launch.py'  
  ])),  
  launch_arguments={ 'use_sim_time': 'true', 'use_ros2_control': 'true' }.items()  
)
```

«Twist_mux» функциясы әртүрлі көздерден келетін қозғалыс пәрмендерін біріктіру үшін қолданылады (мысалы, автономды навигация жүйесі немесе джойстик) және оларды роботтың басқару жүйесіне жіберу. Бұл роботтың жұмысын тоқтатпай әртүрлі басқару режимдері арасында икемді ауысуға мүмкіндік береді.

«twist_mux.yaml» бұл конфигурация файлы әртүрлі құрылғылардан сигналдар бір уақытта келгенде жүйенің әрекетін басқаратын пәрмен басымдықтарын қоса алғанда, мультиплекстеу ережелерін анықтайды.

```
twist_mux_params = os.path.join(get_package_share_directory(package_name),  
  'config', 'twist_mux.yaml')  
twist_mux = Node(  
  package="twist_mux",  
  executable="twist_mux",  
  parameters=[twist_mux_params, {'use_sim_time': True}],  
  remappings=[('/cmd_vel_out', '/diff_cont/cmd_vel_unstamped')]  
)
```

«twist_mux» арқылы өткен қозғалыс пәрмендері робот қозғалтқышы деңгейінде пәрмендерді орындау үшін дифференциалды контроллерге бағытталады.

Gazebo роботты және оның қоршаған ортасын визуалды және физикалық түрде дәл модельдеуді қамтамасыз етеді. «launch_sim.launch.py» конфигурация файлы модельдеу мәліметтерін және оның ROS2

интеграциясын басқаратын параметрлер жинағы бар Gazebo модельдеу ортасын іске қосады.

Қосымша опциялар:

«gazebo_params.yaml» бұл файлда физика модельдеуін және сенсорларды орнатуға арналған параметрлер бар.

«extra_gazebo_args»: Gazebo бағдарламасын ROS2 параметрлерімен іске қосу үшін қосымша аргументтер.

```
gazebo_params_file = os.path.join(get_package_share_directory(package_name),
'config', 'gazebo_params.yaml')
gazebo = IncludeLaunchDescription(
PythonLaunchDescriptionSource([os.path.join(
get_package_share_directory('gazebo_ros'), 'launch', 'gazebo.launch.py')]),
launch_arguments={'extra_gazebo_args': '--ros-args --params-file ' +
gazebo_params_file}.items()
)
```

Робот модельдеу ортасында пайда болуы үшін оның үлгісі «spawn_entity.py» түйінін пайдаланып Gazebo-ға орналастырылуы керек. Бұл түйін әдетте роботтың URDF файлынан жасалған "robot_description" тақырыбынан робот сипаттамасын жүктейді және робот үлгісін имитацияланған ортаға орналастырады.

```
spawn_entity = Node(
package='gazebo_ros',
executable='spawn_entity.py',
arguments=['-topic', 'robot_description', '-entity', 'my_bot'],
output='screen'
)
```

Бұл процесс робот моделінің дұрыс физика қасиеттері мен параметрлері бар модельдеуге орналастырылуын қамтамасыз етеді.

Gazebo робот қозғалысын басқару үшін ROS2 Control жүйесімен біріктірілген ROS2 контроллерлерін пайдаланады. Контроллерлер қозғалыс пәрмендерін алады және оларды дөңгелектер сияқты роботтың жетектеріне жібереді. «launch_sim.launch.py» файлында екі негізгі контроллер іске қосылады:

«diff_drive_spawner» жылдамдық командалары негізінде робот қозғалысын басқару үшін дифференциалды контроллерді (diff_cont) іске қосатын түйін.

```
diff_drive_spawner = Node(
package="controller_manager",
executable="spawner.py",
arguments=["diff_cont"]
)
```

joint_broad_spawner контроллерге барлық робот буындарының күйін жариялауға мүмкіндік беретін түйін, олардың орнын нақты уақыт режимінде көрсетуге мүмкіндік береді.

```
joint_broad_spawner = Node(  
    package="controller_manager",  
    executable="spawner.py",  
    arguments=["joint_broad"]  
)
```

Бұл түйіндер Газебодағы нақты робот қозғалысын модельдеуге мүмкіндік беретін сәйкес контроллердің іске қосылуын басқарады.

4.4 Тестілеу ортасын ұйымдастыру

Жүйені тексеру және модельдеу мақсатында CorreliaSim және Gazebo симуляциялық орталарында арнайы тәжірибелік алаң жасалды. CorreliaSim (бұрынғы атауы V-REP) робототехникалық жүйелерді модельдеуге және сынауға арналған әмбебап 3D симуляциялық платформа. Бұл орта нақты физикалық модельдеуге, түрлі сенсорлар мен басқару алгоритмдерін тестілеуге, сондай-ақ көп агентті жүйелерді құруға мүмкіндік береді.

4.7 суретте, CorreliaSim ортасы роботтардың қозғалысын, сенсорлық ақпаратты өңдеуді және ортадағы объектілермен әрекеттесуін дәл модельдейді. Ол бір уақытта бірнеше роботтың өзара әрекеттесуін зерттеуге және координация алгоритмдерін сынауға қолайлы.



4.7-сурет – CorreliaSim графикалық ортада арнайы алаң

Зерттеу аясында роботтардың қауіпті ортада әрекет ету қабілетін тексеру үшін CoppeliaSim симуляторында арнайы апаттық жағдайды модельдейтін тест алаңы жасалды. Бұл ортаның негізгі мақсаты – көп агентті жүйенің өрт, түтін және бөгде кедергілерге (мысалы, қирандылар немесе адамдар) шынайы уақыт режимінде әрекет етуін бағалау. Алаң шекарамен қоршалған және бірқатар виртуалды қауіп элементтерімен жабдықталған.

4.8 суретте, сценарий құрамында бірнеше қызыл сфералық объектілер бар, олар тірі адамды немесе жылу көзі ретінде шартты түрде қабылданады. Сонымен қатар, түтін мен жалын анимациялары өрт шыққан аймақтарды белгілейді. Құлаған темір құрылымдар қирандылардың болуын білдіреді, ал кейбір аумақтарда қозғалысты шектейтін бөгеттер орналасқан. Мұндай орта нақты апат жағдайының моделін құра отырып, роботтардың навигациялық және сенсорлық функцияларын сынауға мүмкіндік береді.



4.8-сурет – CoppeliaSim графикалық ортада қауіп қатер бейнесі

4.9 суретте, CoppeliaSim симуляторының ең үлкен артықшылығы әмбебаптығы мен икемділігі. Бұл ортада кез келген типтегі роботтарды, манипуляторлардан бастап автономды ұшатын аппараттарға дейін, шынайы физика заңдылықтарына негізделген модельдеу арқылы сынауға болады. Бағдарлама әртүрлі сенсорларды (LiDAR, камера, инфрақызыл, дыбыс) оңай біріктіріп, олардан алынған деректерді өңдеу алгоритмдерін сынауға мүмкіндік береді. Сонымен қатар, әр агенттің жеке логикасын орнатуға болады, бұл көп агентті жүйелерді зерттеуге өте қолайлы.

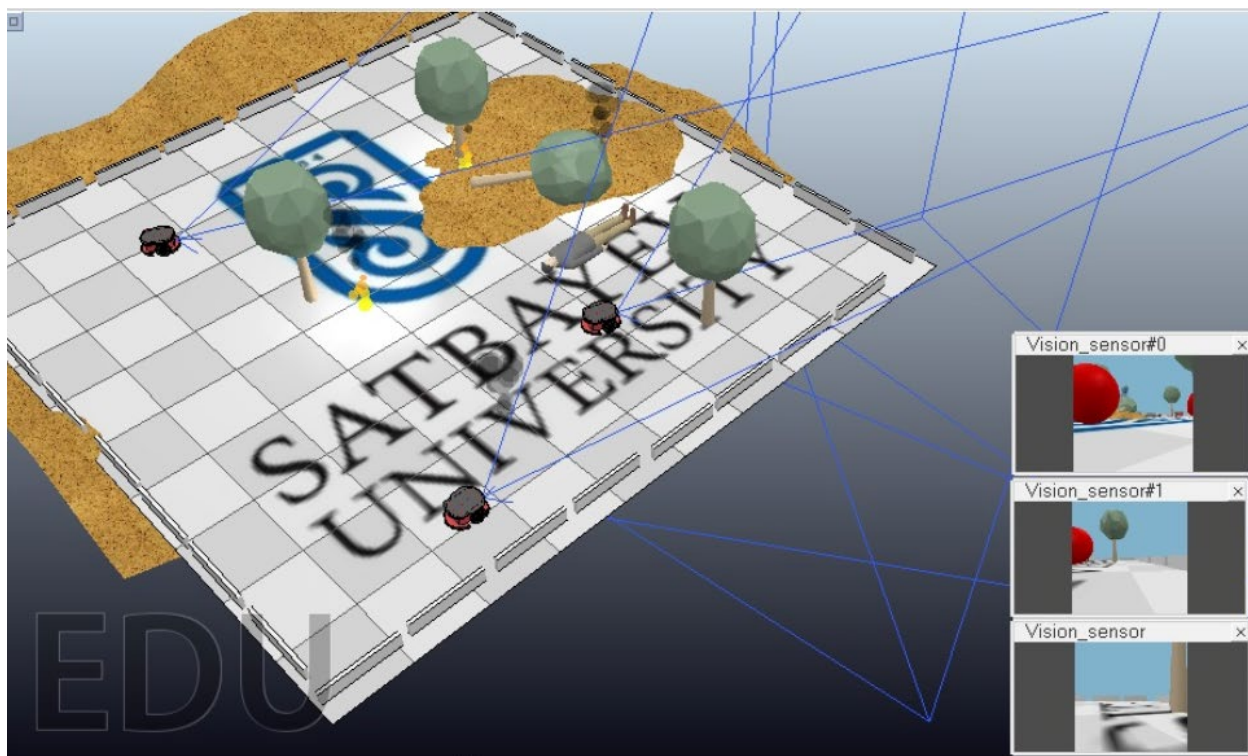
Екінші маңызды артықшылық көп тілді қолдауы мен ашық интерфейстері. CorreliaSim ішкі басқару үшін Lua тілін пайдаланады, ал сыртқы жүйелермен байланыс орнату үшін Python, C++, MATLAB және ROS интерфейстерін ұсынады. Бұл мүмкіндіктер зерттеушіге модельденген ортаны нақты кодпен басқаруға және сыртқы алгоритмдерді оңай интеграциялауға жол ашады. Атап айтқанда, ROS2-мен тікелей байланыс орнату мүмкіндігі роботтарды нақты жүйелерге көшіру кезінде уақыт пен күш үнемдейді.

Үшінші артықшылық интерактивті визуализация және шынайы уақыттағы бақылау мүмкіндігі. CorreliaSim роботтардың сенсорлардан не «көріп» тұрғанын, қалай қозғалатынын нақты уақытта көрсетеді. Мұндай визуалды кері байланыс зерттеудің сапасын арттырып, жүйе логикасындағы ақауларды жылдам анықтауға көмектеседі. Сонымен қатар, көп сценарийлі симуляциялар құру арқылы роботтардың түрлі ортада қалай әрекет ететінін алдын ала сынауға болады, бұл нақты прототип жасамай-ақ тиімді жүйелер әзірлеуге мүмкіндік береді.



4.9-сурет – CorreliaSim графикалық ортада симуляция

4.10 суретте көрсетілгендей, оң жақтағы камера терезелерінен (Vision_sensor#0, #1, #2) қызыл нысандар мен апатты элементтерді роботтардың сенсорлары нақты уақыт режимінде тіркеп тұрғанын байқауға болады. Бұл камералы роботтың өрт немесе қауіпті аймақты қашықтықтан бақылап, өңделген сурет негізінде шешім қабылдау алгоритмін іске қосқанын білдіреді. Сонымен қатар, ағаштардың арасындағы кедергілер мен құм төбелер SLAM және навигациялық жүйелер үшін нақты сынақтық элементтер қызметін атқарады.



4.10-сурет - CoppeliaSim графикалық ортада камералық жағдай

Көп агентті роботтар осы ортада бір-біріне кедергі келтірмей, карта жасап, адамды іздеп, қауіпті аймақтарды белгілеп әрекет етеді. Алынған камералық бейнелер ішкі жүйеге өңдеуге беріледі, ал бұл өз кезегінде нақты уақыттағы адаптивті маршруттауды қамтамасыз етеді. Бұл сцена жүйенің шынайы құтқару жағдайында қолданылу тиімділігін бағалауға мүмкіндік берді.

Қорытынды

Жүргізілген зерттеу нәтижесінде жер сілкінісі жағдайында іздестіру-құтқару операцияларын орындауға арналған көп агентті роботтық жүйе әзірленіп, оның тиімді жұмыс істейтіні көрсетілді. Зерттеу барысында бірнеше автономды роботтардың өзара әрекеттесуіне негізделген үйлесімді жұмыс алгоритмдері жүзеге асырылды. Роботтар белгісіз ортада бір мезгілде өзіндік орналасуын анықтап, карта құрастыру (SLAM) және сол картаны пайдаланып автономды навигация жасау мүмкіндіктері іске асты. Нәтижесінде, бірнеше роботтан тұратын топ ортақ карта түзіп, кедергілерді айналып өту арқылы берілген мақсаттарға дербес жете алды. Бір робот үшін мүмкін болмайтын кең аумақты бірнеше робот қатар зерттей алады; сонымен бірге, әр роботтың резервтік рөлі жалпы жүйенің сенімділігін арттыратыны белгілі. Осылайша, ұсынылған тәсіл автономды роботтардың бірлесіп әрекеттесуі арқылы іздестіру-құтқару тиімділігін едәуір жоғарылататынын көрсетті.

Жүйені сынақтан өткізу және модельдеу Gazebo және CoppeliaSim платформаларында жүргізілді. Бұл виртуалды орталар роботтардың қозғалысы мен сенсор әсерлерін шынайы физикалық жағдайларға жақын етіп қайта шығаруға мүмкіндік берді. Модельдеу барысындағы навигация және карта құру процестерін визуализациялау үшін RViz құралы пайдаланылды, ол әрбір роботтың күйін, траекториясын және құрылған карталарды нақты уақыт режимінде қадағалауға жағдай жасады. Жүйенің бағдарламалық жүзеге асырылуы Robot Operating System 2 (ROS2) негізінде орындалып, маршрут жоспарлау мен қозғалысты басқару үшін Navigation2 (Nav2) пакеті қолданылды. Заманауи ROS2 және Gazebo сияқты құралдарды пайдалану зерттеу алгоритмдерін шынайы сценарийлерде тиімді тексеруге мүмкіндік берді.

Көп агентті роботтық жүйенің тиімділігі бірнеше көрсеткіш бойынша бағаланды. Атап айтқанда, роботтар арасында тапсырмаларды бөлу стратегиясының жұмыс істеуі, карта құру дәлдігі және навигация траекториясының нақтылығы нақты уақыт режимінде өлшенді. Сынақ нәтижелері тапсырмаларды бөлудің тиімді жүзеге асқанын көрсетті: әрбір робот өзіне бекітілген аумақты дербес зерттеп, қайталанатын жұмыстарды болдырмады. Құрастырылған карталардың дәлдігі жоғары деңгейде болып, модельденген ортадағы кедергілер мен объектілер картада дұрыс бейнеленді. Сол сияқты, навигацияның дәлдігі де қанағаттанарлық деңгейде екені анықталды – роботтар берілген межелі нүктелерге кедергісіз және дәл жетті.

Зерттеу барысында бірқатар техникалық қиындықтар да анықталды. Солардың бірі сенсор деректерінің уақытша кідірісі мәселесі. Кей жағдайларда роботтардың лидар, камера сынды сенсорларынан алынатын ақпаратта аздаған кешігулер пайда болды, бұл деректерді біріктіру және дәл карта құру үдерістеріне әсер етуі мүмкін еді. Алайда аталған уақытша кідіріс жүйенің жалпы жұмыс істеу қабілеттілігіне айтарлықтай кедергі келтірмеді. Роботтар

арасындағы үйлестіру мен навигация тұрақты сақталып, сенсор деректеріндегі кідірістер нақты уақыттағы жұмыстың нәтижелеріне елеулі ықпал еткен жоқ.

Қорыта келе, бұл диссертациялық зерттеу жұмысы қойылған ғылыми мақсат-міндеттерді толық орындап, жер сілкінісінен құтқаруға бағытталған көп агентті роботтық жүйенің тиімділігін модельдік сынақтар арқылы дәлелдеді. Алынған нәтижелерді алдағы уақытта практикалық тұрғыда дамыту мүмкіндігі бар. Атап айтқанда, әзірленген жүйені жер сілкінісі мен құрылыс апаттары жағдайында алғашқы іздестіру-құтқару кезеңінде нақты қолдануға бейімдеуге болады. Ол үшін роботтардың нақты аппараттық платформаларда жүзеге асырылуын қамтамасыз ету, дала жағдайындағы сынақтарды өткізу және сенсор деректерінің кідірістеріне төзімділігін арттыру бойынша қосымша жұмыстар атқарылуы тиіс. Жалпы, ұсынылған көп агентті роботтық жүйе төтенше жағдайлардағы іздестіру-құтқару жұмыстарына жеделдік пен тиімділік қосып, құтқарушылардың жұмысын жеңілдетуге және адам өмірін сақтап қалуға маңызды үлес қоса алатынын көрсетті.

Пайдаланылған әдебиеттер тізімі

- 1 Назарбаева, С.М., Сурашев, Н.Т. Көтеру-тасымалдау жүйелері және робот техникасы: Оқулық / С.М. Назарбаева, Н.Т. Сурашев. – Алматы: Дәуір, 2011. – 256 б.
- 2 Шыңғысов, Б.Т. Робот техникасы негіздері: Оқулық / Б.Т. Шыңғысов. – Алматы: LantarBooks, 2020. – 312 б.
- 3 Макаров, И.М. Робототехника. История и перспективы / И.М. Макаров. – М.: Наука, 2015. – 200 с.
- 4 Байғұншеков, Ж.Ж. Структурно-кинематический синтез механизмов и манипуляционных устройств высоких классов / Ж.Ж. Байғұншеков. – Алматы: Ғылым, 1993. – 220 с.
- 5 Молдабеков, М.М. Основы аналитической теории манипуляционных устройств высоких классов / М.М. Молдабеков. – Алматы: Білім, 1989. – 180 с.
- 6 Жолдасбеков, Ө.А. Теория механизмов и машин / Ө.А. Жолдасбеков. – Алматы: Мектеп, 1978. – 320 с.
- 7 Stewart, J., Church, M. ROS 2 Robot With SLAM / J. Stewart, M. Church // ResearchGate. – 2024. – DOI: 10.13140/RG.2.2.15510.56643
- 8 Mansur, I.B. SLAM and Nav2 for Custom Robots in ROS2 / I.B. Mansur // *Medium*. – 2025. – <https://ibrahimmansur4.medium.com/slam-and-nav2-for-custom-robots-in-ros2-905400437ae1>.
- 9 Parker L. E. ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation // IEEE Transactions on Robotics and Automation. – 1998. – Vol. 14, № 2. – P. 220–240
- 10 Balch T., Parker L. E. Distributed Behavior-Based Control for Multi-Robot Teams // Autonomous Robots. – 2002. – Vol. 7, № 3. – P. 217–232.
- 11 Murphy R. R. Trial by Fire [Rescue Robots] // IEEE Robotics & Automation Magazine. – 2004. – Vol. 11, № 3. – P. 50–61.
- 12 Casper J., Murphy R. R. Human–Robot Interactions During the Robot-Assisted Urban Search and Rescue Response at the World Trade Center // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). – 2003. – Vol. 33, № 3. – P. 367–385.
- 13 Amigoni F. Coordinating Multiple Robots for Emergency Response // Expert Systems with Applications. – 2008. – Vol. 34, № 2. – P. 1284–1297.
- 14 Yim M., Shen W.-M., Salemi B., Rus D., Moll M., Lipson H., Klavins E., Chirikjian G. Modular Self-Reconfigurable Robot Systems: State of the Art and Grand Challenges // IEEE Robotics & Automation Magazine. – 2007. – Vol. 14, № 1. – P. 43
- 15 Quaritsch M., Kriechbaum M., Bernsteiner D., Rinner B. Deployable Multi-Agent Robotics for Disaster Management // Journal of Field Robotics. – 2011. – Vol. 28, № 6. – P. 914–934.
- 16 Thrun S., Burgard W., Fox D. Probabilistic Robotics. – Cambridge, MA: MIT Press, 2005. – 345 p.

- 17 Murphy R. R. Disaster Robotics. – Cambridge, MA: MIT Press, 2014.
– 264 p.
- 18 Zeng Z., Yin Z. Multi-Agent Systems in Earthquake Disaster Response:
A Review // International Journal of Disaster Risk Reduction. – 2020. – Vol.45. – P.
101461.
- 19 Lee K., Lee J., Kim H. Adaptive Multi-Robot Task Allocation in
Uncertain Environments // Robotics and Autonomous Systems. – 2019. – Vol. 115.
–P. 45–61.

А Қосымша

```
<?xml version="1.0"?>
<robot name="multi_bot" xmlns:xacro="http://www.ros.org/wiki/xacro">

  <!-- Бірінші SLAM роботы -->
  <xacro:include filename="robot_core.xacro"/>
  <xacro:robot_core prefix="r1_" x="0" y="0" z="0"/>

  <!-- Екінші адамды танитын робот -->
  <xacro:robot_core prefix="r2_" x="1" y="0" z="0"/>

  <!-- Үшінші қауіп-қатерді бақылайтын робот -->
  <xacro:robot_core prefix="r3_" x="2" y="0" z="0"/>

</robot>

<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">

  <xacro:property name="prefix" value=""/>
  <xacro:property name="x" value="0"/>
  <xacro:property name="y" value="0"/>
  <xacro:property name="z" value="0"/>

  <link name="${prefix}base_link">
    <visual>
      <geometry><cylinder length="0.3" radius="0.2"/></geometry>
      <material name="black"><color rgba="0.2 0.2 0.2 1"/></material>
    </visual>
  </link>

  <joint name="${prefix}base_joint" type="fixed">
    <parent link="world"/>
    <child link="${prefix}base_link"/>
    <origin xyz="${x} ${y} ${z}" rpy="0 0 0"/>
  </joint>

</robot>

from launch import LaunchDescription
from launch_ros.actions import Node
from launch.actions import ExecuteProcess

def generate_launch_description():
    return LaunchDescription([
```

А Қосымша жалғасы

```
ExecuteProcess(
    cmd=['gazebo', '--verbose', '-s', 'libgazebo_ros_factory.so'],
    output='screen'
),
Node(
    package='robot_state_publisher',
    executable='robot_state_publisher',
    output='screen',
    parameters=[{'robot_description':
open('/home/don/rescue_ws/src/my_bot/urdf/multi_bot.xacro').read()}],
),
Node(
    package='slam_toolbox',
    executable='async_slam_toolbox_node',
    name='slam_toolbox',

parameters=['/home/don/rescue_ws/src/my_bot/config/mapper_params_online_async.yaml'],
    output='screen'
),
Node(
    package='rviz2',
    executable='rviz2',
    name='rviz2',
    output='screen'
),
])
slam_toolbox:
ros__parameters:
  use_sim_time: true
  mode: "mapping"
  map_update_interval: 2.0
  resolution: 0.05
  max_laser_range: 10.0

function sysCall_init()
  usensors={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1}
  for i=1,16,1 do
    usensors[i]=sim.getObjectHandle("Pioneer_p3dx_ultrasonicSensor"..i)
  end
  motorLeft=sim.getObjectHandle("Pioneer_p3dx_leftMotor")
  motorRight=sim.getObjectHandle("Pioneer_p3dx_rightMotor")
```

А Қосымша жалғасы

```
noDetectionDist=0.5
maxDetectionDist=0.2
detect={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
braitenbergL={-0.2,-0.4,-0.6,-0.8,-1,-1.2,-1.4,-1.6,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
braitenbergR={-1.6,-1.4,-1.2,-1,-0.8,-0.6,-0.4,-0.2,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}

vBaseLeft = 1.5
vBaseRight = 2.0
end

function sysCall_cleanup()
end

function sysCall_actuation()
    local obstacleDetected = false

    for i=1,16,1 do
        local res,dist = sim.readProximitySensor(usensors[i])
        if (res>0) and (dist<noDetectionDist) then
            obstacleDetected = true
            break
        end
    end

    if obstacleDetected then
        sim.setJointTargetVelocity(motorLeft, 0)
        sim.setJointTargetVelocity(motorRight, 0)
    else
        for i=1,16,1 do
            local res,dist = sim.readProximitySensor(usensors[i])
            if (res>0) and (dist<noDetectionDist) then
                if (dist<maxDetectionDist) then
                    dist=maxDetectionDist
                end
                detect[i]=1-((dist-maxDetectionDist)/(noDetectionDist-
maxDetectionDist))
            else
                detect[i]=0
            end
        end
    end
end
```

А Қосымша жалғасы

```
vLeft = vBaseLeft
vRight = vBaseRight

for i=1,16,1 do
    vLeft = vLeft + braitenbergL[i]*detect[i]
    vRight = vRight + braitenbergR[i]*detect[i]
end

sim.setJointTargetVelocity(motorLeft,vLeft)
sim.setJointTargetVelocity(motorRight,vRight)
end
end

function sysCall_init()
    usensors={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1}
    for i=1,16,1 do
        usensors[i]=sim.getObjectHandle("Pioneer_p3dx_ultrasonicSensor"..i)
    end
    motorLeft=sim.getObjectHandle("Pioneer_p3dx_leftMotor")
    motorRight=sim.getObjectHandle("Pioneer_p3dx_rightMotor")
    noDetectionDist=0.5
    maxDetectionDist=0.2
    detect={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
    braitenbergL={-0.2,-0.4,-0.6,-0.8,-1,-1.2,-1.4,-1.6,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
    braitenbergR={-1.6,-1.4,-1.2,-1,-0.8,-0.6,-0.4,-0.2,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}

    vBaseLeft = 1.5
    vBaseRight = 2.0
end

function sysCall_cleanup()
end

function sysCall_actuation()
    for i=1,16,1 do
        res,dist=sim.readProximitySensor(usensors[i])
        if (res>0) and (dist<noDetectionDist) then
            if (dist<maxDetectionDist) then
                dist=maxDetectionDist
            end
        end
    end
end
```

А Қосымша жалғасы

```
        detect[i]=1-((dist-maxDetectionDist)/(noDetectionDist-maxDetectionDist))
    else
        detect[i]=0
    end
end

vLeft=vBaseLeft
vRight=vBaseRight

for i=1,16,1 do
    vLeft=vLeft+braitenbergL[i]*detect[i]
    vRight=vRight+braitenbergR[i]*detect[i]
end
sim.setJointTargetVelocity(motorLeft,vLeft)
sim.setJointTargetVelocity(motorRight,vRight)
end

function sysCall_init()
    usensors={}
    for i=1,16 do
        usensors[i]=sim.getObjectHandle("Pioneer_p3dx_ultrasonicSensor"..i)
    end
    motorLeft=sim.getObjectHandle("Pioneer_p3dx_leftMotor")
    motorRight=sim.getObjectHandle("Pioneer_p3dx_rightMotor")

    noDetectionDist=0.5
    maxDetectionDist=0.2
    detect={}

    vBaseLeft = 1.5
    vBaseRight = 2.0
    reverseTime = 5.0
    turnTime = 1.5

    state = "forward"
    stateStart = sim.getSimulationTime()
end

function sysCall_actuation()
    local timeNow = sim.getSimulationTime()

    local obstacle = false
```

А Қосымша жалғасы

```
for i=1,16 do
    local res, dist = sim.readProximitySensor(usensors[i])
    if (res>0) and (dist<noDetectionDist) then
        obstacle = true
        break
    end
end

if state == "forward" and obstacle then
    state = "reverse"
    stateStart = timeNow
elseif state == "reverse" and timeNow - stateStart >= reverseTime then
    state = "turn"
    stateStart = timeNow
elseif state == "turn" and timeNow - stateStart >= turnTime then
    state = "forward"
    stateStart = timeNow
end

if state == "forward" then
    local vLeft = vBaseLeft
    local vRight = vBaseRight

    for i=1,16 do
        local res, dist = sim.readProximitySensor(usensors[i])
        if (res>0) and (dist<noDetectionDist) then
            if (dist<maxDetectionDist) then
                dist=maxDetectionDist
            end
            detect[i]=1-((dist-maxDetectionDist)/(noDetectionDist-
maxDetectionDist))
        else
            detect[i]=0
        end
    end

    braitenbergL={-0.2,-0.4,-0.6,-0.8,-1,-1.2,-1.4,-1.6,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
    braitenbergR={-1.6,-1.4,-1.2,-1,-0.8,-0.6,-0.4,-0.2,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
    for i=1,16 do
        vLeft = vLeft + braitenbergL[i]*detect[i]
        vRight = vRight + braitenbergR[i]*detect[i]
    end
end
```

А Қосымша жалғасы

```
end

sim.setJointTargetVelocity(motorLeft, vLeft)
sim.setJointTargetVelocity(motorRight, vRight)

elseif state == "reverse" then
    sim.setJointTargetVelocity(motorLeft, -2.0)
    sim.setJointTargetVelocity(motorRight, -2.0)

elseif state == "turn" then
    sim.setJointTargetVelocity(motorLeft, 2.0)
    sim.setJointTargetVelocity(motorRight, -2.0)
end
end
```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРҒЫ БІЛІМ МИНИСТРЛІГІ

«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық емес акционерлік қоғамы

7M07107 – «Робототехника және мехатроника» мамандығының студенті

Игибай Даниярдың
Магистрлік диссертациясы

Тақырыбы: «Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі»

Әзірленген:

а) графикалық бөлімі _____ парақ

б) түсіндірме жазбасы _____ бетте

ЖҰМЫСҚА ЕСКЕРТУ ЖАСАУ

Игибай Даниярдың магистрлік диссертациясы «Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі» атты тақырыпқа арналған. Диссертациялық жұмыстың негізгі мақсаты жер сілкінісі кезінде іздеу-құтқару жұмыстарын тиімді жүргізуге арналған үш түрлі функциясы бар роботтардан тұратын көп агентті жүйені модельдеу және олардың өзара әрекеттесуін ұйымдастыру болып табылады. Жұмыста SLAM арқылы карта жасайтын лидермен жабдықталған робот, адамдарды анықтайтын камерамен жабдықталған робот және қауіптілікті (газ, от, т.б.) анықтайтын сенсорлы робот қарастырылады.

Жұмыстың ғылыми жаңалығы әртүрлі функционалдық міндеттер атқаратын бірнеше мобильді роботтардың өзара байланысын, деректермен алмасуын және ортақ мақсатқа бағытталған әрекетін бірлескен түрде ұйымдастыру. Жүйе ROS2 ортасында құрылған, ал модельдеу Gazebo және CoppeliaSim бағдарламалық құралдары арқылы жүзеге асырылған. Роботтардың URDF және XACRO файлдары құрастырылып, RViz визуализациясы орындалған.

Жұмыс барысында Python, C++ тілдері, ROS2, Gazebo, CoppeliaSim құралдары қолданылған. Үш роботтың әрекеті XACRO және URDF форматтарында сипатталып, RViz визуализациясы жүзеге асырылған. Жұмысқа қатысты ғылыми мақала «World of Conference» ғылыми журналына жарияланған және «Фараби әлемі» халықаралық ғылыми конференциясына жарыққа шықты. Жұмыстың тақырыбын мүмкіндігінше нақтылау қажет, тақырып атауы басқаша қабылдануы мүмкін.

Бірінші бөлімде жер сілкінісінен құтқару роботтарының түрлері мен халықаралық тәжірибелер қарастырылған. Екінші бөлімде көп агентті жүйенің құрылымы, архитектурасы мен деректер алмасу механизмдері сипатталған. Үшінші бөлімде роботтардың ортада жұмыс істеуін модельдеу, тапсырмаларды бөлу және нәтижелерді бағалау берілген. Жұмыс құрылымы мен мазмұны жалпы ғылыми талаптарға сәйкес келеді. Графикалық материалдар мен сипаттамалар толық және түсінікті түрде ұсынылған.

ЖҰМЫС БАҒАСЫ

Магистрант Игибай Даниярдың «Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі» атты диссертациялық жұмысы өзекті, жоғары деңгейде орындалған, қойылған ғылыми-техникалық мәселелер нақты шешімін тапқан.

Магистрлік диссертацияны 90 балға бағалап, магистр академиялық дәрежесіне лайық деп есептеймін.

Сын пікір беруші

эл-Фараби атындағы ҚазҰУ,
Плазма физикасы, нанотехнология
және компьютерлік физика
кафедрасының қауымдастырылған профессоры, PhD докторы
« _____ » _____ 2025 ж

Ф ҚазҰТЗУ 704-22 Рецензия





Отчет подобия

Метаданные

Название организации

Satbayev University

Название

Жер сілкінісінен құтқарудың көп агентті роботтық жүйесі

Автор

Научный руководитель / Эксперт

Игибай Данияр Галымбекулы Мурат Бактыбаев

Подразделение

ИАиИТ

Объем найденных подобиий

КП-ия определяют, какой процент текста по отношению к общему объему текста был найден в различных источниках.. Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



КП1

25

Длина фразы для коэффициента подобия 2



КП2

11149

Количество слов



КЦ

89033

Количество символов

Тревога

В этом разделе вы найдете информацию, касающуюся текстовых искажений. Эти искажения в тексте могут говорить о ВОЗМОЖНЫХ манипуляциях в тексте. Искажения в тексте могут носить преднамеренный характер, но чаще, характер технических ошибок при конвертации документа и его сохранении, поэтому мы рекомендуем вам подходить к анализу этого модуля со всей долей ответственности. В случае возникновения вопросов, просим обращаться в нашу службу поддержки.

Замена букв		0
Интервалы		0
Микропробелы		5
Белые знаки		0
Парафразы (SmartMarks)		6

Подобия по списку источников

Ниже представлен список источников. В этом списке представлены источники из различных баз данных. Цвет текста означает в каком источнике он был найден. Эти источники и значения Коэффициента Подобия не отражают прямого плагиата. Необходимо открыть каждый источник и проанализировать содержание и правильность оформления источника.

10 самых длинных фраз

Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	https://with-rl.tistory.com/entry/Gazebo%E2%84%B5-%E2%9D%B4%E2%9A%A9%E2%95%B4-2D-LiDAR-%E2%8B%9C%E2%AE%AC%E2%A0%88%E2%9D%B4%E2%85%98-%E2%95%98%E2%B8%B0	21 0.19 %
2	https://with-rl.tistory.com/entry/Gazebo%E2%84%B5-%E2%9D%B4%E2%9A%A9%E2%95%B4-2D-LiDAR-%E2%8B%9C%E2%AE%AC%E2%A0%88%E2%9D%B4%E2%85%98-%E2%95%98%E2%B8%B0	17 0.15 %

3	https://with-rl.tistory.com/entry/Gazebo%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%B4-2D-LiDAR-%EC%8B%9C%EB%AE%AC%EB%A0%88%EC%9D%B4%EC%85%98-%ED%95%98%EA%B8%B0	17 0.15 %
4	https://ni.www.techfak.uni-bielefeld.de/files/URDF-XACRO.pdf	11 0.10 %
5	LT_Karl_Kink.pdf 5/16/2024 Tallinna Tehnikakõrgkool (Tallinna Tehnikakõrgkool)	10 0.09 %
6	https://ni.www.techfak.uni-bielefeld.de/files/URDF-XACRO.pdf	6 0.05 %
7	https://with-rl.tistory.com/entry/Gazebo%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%B4-2D-LiDAR-%EC%8B%9C%EB%AE%AC%EB%A0%88%EC%9D%B4%EC%85%98-%ED%95%98%EA%B8%B0	6 0.05 %
8	LT_Karl_Kink.pdf 5/16/2024 Tallinna Tehnikakõrgkool (Tallinna Tehnikakõrgkool)	5 0.04 %

из базы данных RefBooks (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из домашней базы данных (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из программы обмена базами данных (0.13 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	LT_Karl_Kink.pdf 5/16/2024 Tallinna Tehnikakõrgkool (Tallinna Tehnikakõrgkool)	15 (2) 0.13 %

из интернета (0.70 %)

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	https://with-rl.tistory.com/entry/Gazebo%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%B4-2D-LiDAR-%EC%8B%9C%EB%AE%AC%EB%A0%88%EC%9D%B4%EC%85%98-%ED%95%98%EA%B8%B0	61 (4) 0.55 %
2	https://ni.www.techfak.uni-bielefeld.de/files/URDF-XACRO.pdf	17 (2) 0.15 %

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	------------	---

CERTIFICATE

Igibay Daniyar Galymbekuly

for participation in

XX international scientific conference. Dortmund. Germany. 01-02.05.2025

«Development of science in the XXI century»

with scientific work

MULTI-AGENT ROBOTIC SYSTEM FOR EARTHQUAKE RESCUE



DOI <https://doi.org/10.5281/zenodo.15367077>




Editor-in-Chief Tarmo Vesik

07.05.2025