

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

ЖУРЕНКО Алла Миколаївна

**Математичне та програмне забезпечення модуля
навігації мобільного робота/ Mathematical tool
and software for navigation module of mobile robot**

спеціальність: 8.05010302 - Інженерія програмного забезпечення
магістерська програма - Інженерія програмного забезпечення

Магістерська робота

Виконала студентка групи
ІПЗм-21
А. М. Журенко (Акокхія)

Науковий керівник:
к.т.н., доцент ШПІНТАЛЬ М.Я.

Магістерську роботу допущено
до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2016

ВСТУП

Роботи - автоматичні системи, призначені для відтворення рухових та інтелектуальних функцій людини. Від традиційних автоматів відрізняються більшою універсальністю і здатністю адаптації на виконання різних завдань, у тому числі в мінливих обстановці.

В даний час робототехніка перетворилася на розвинену галузь промисловості: тисячі роботів працюють на різних підприємствах світу, підводні маніпулятори стали неодмінною приналежністю підводних дослідницьких та рятувальних апаратів, вивчення космосу спирається на широке використання роботів з різним рівнем інтелекту. Особлива увага приділяється автоматизації важких, шкідливих, утомливих і монотонних робіт в різних галузях з допомогою роботів-маніпуляторів.

Проте сьогодні у фахівців в галузі робототехніки виникають приблизно ті ж труднощі, що й 30 років тому у розробників ЕОМ. Через відсутність загальних стандартів і платформ творцям роботів доводиться починати розробку кожного нового творіння практично з нуля.

Все ж, незважаючи на всі складності, ті, хто зайнятий у сфері робототехніки, від професорів до підприємців і студентів, повні ентузіазму, що нагадує про пору створення Microsoft, коли творці шукали шляхи розвитку нових технологій і мріяли, щоб комп'ютери були доступні кожному. І сьогодні, аналізуючи тенденції розвитку робототехніки, можна уявити майбутнє, де роботи стануть незамінними помічниками людей в їх повсякденному житті. Можливо, людство знаходиться на порозі нової ери, коли персональні комп'ютери зійдуть зі столів і дозволять нам бачити, чути, відчувати і, можливо навіть, маніпулювати предметами на відстані.

Зараз розробники систем з використанням штучного інтелекту можуть оснастити свої творіння системою навігації GPS, відеокамерами і безліччю

додаткових детекторів, в результаті чого можливості сучасних роботів збільшуються.

Зв'язок домашніх роботів і персональних комп'ютерів полегшить життя людині. Наприклад, офісний службовець стежить за охороною свого будинку, прибиранням, розкладанням випраної білизни, контролюючи роботу домашніх роботів на екрані свого ПК. Крім того, роботи зможуть обмінюватися інформацією між собою і домашнім комп'ютером [1].

Метою даної роботи є визначення завдань та розробка структурної схеми системи навігації мобільного робота.

Для успішної навігації в просторі система робота повинна вміти будувати маршрут, управляти параметрами руху (задавати кут повороту коліс і швидкість їх обертання), правильно інтерпретувати відомості про навколишній світ, одержувані від датчиків, і постійно відстежувати власні координати.

Комп'ютерні системи побудови маршруту розроблені досить добре. Спочатку вони створювалися для найпростіших віртуальних середовищ, і програма, що моделює дії робота, швидко знаходила оптимальний шлях до мети в двовимірних лабіринтах і кімнатах, наповнених простими перешкодами. Коли з'явилися швидкі процесори, стало можливим формувати траєкторію руху вже на складних тривимірних картах, причому в реальному часі [2].

Сучасна робототехніка виникла в 60-ті - 70-ті роки минулого сторіччя як відповідь на запити комплексної автоматизації, коли в результаті з'єднання керованих людиною маніпуляторів з системами ЧПУ верстатів та іншого технологічного обладнання з'явилися автоматичні машини принципово нового типу. Це були роботи з програмним управлінням - роботи першого покоління.

Успіхи застосування перших роботів викликали швидке зростання потреб у них і відповідно вимог до їх можливостей. Стали розвиватися роботи з комбінованим керуванням, в яких програмне управління доповнюється

управлінням від людини-оператора - роботи проміжного 1,5-го покоління з супервизорного, а потім інтерактивним управлінням.

У ті роки тільки перші кроки почала робити теорія адаптивного управління. І одними з перших машин з таким управлінням стали адаптивні роботи. Це роботи другого покоління, оснащені сенсорикой.

За міру розвитку систем адаптивного управління в них стали застосовуватися методи штучного інтелекту. Коли ці технології зайняли визначальне становище в алгоритмічній забезпеченні систем управління, сформувалося нове, третє покоління роботів - інтелектуальні роботи [3].

I. АНАЛІЗ І БУДОВА РОБОТІВ

1.1 Аналіз та будова роботів

Сучасні роботи функціують на основі принципів зворотного зв'язку, підлеглого управління і ієрархічності системи управління роботом.

Ієрархія системи управління роботом має на увазі ділення системи управління на горизонтальні шари, що управляють загальною поведінкою робота, розрахунком необхідної траєкторії руху маніпулятора, поведінкою окремих його приводів, і шари, що безпосередньо здійснюють управління двигунами приводів[2].

Концепція підлеглого управління служить для побудови системи управління приводом[1]. Сенс концепції такий: Хай необхідно побудувати систему управління приводом по положенню (наприклад, по куту повороту ланки маніпулятора). Система замикається зворотним зв'язком по положенню, а усередині системи управління по положенню функціонує система управління за швидкістю з своїм зворотним зв'язком за швидкістю, усередині якої існує контур управління по струму з своїм зворотним зв'язком.

Сучасний робот оснащений не тільки зворотними зв'язками по положенню, швидкості і прискоренням ланок[12]. При захопленні деталей робот повинен знати, чи вдало він захопив деталь. Якщо деталь крихка або її поверхня має високий ступінь чистоти, будуються складні системи із зворотним зв'язком по зусиллю, що дозволяють роботів схоплювати деталь, не ушкоджуючи її поверхню і не руйнуючи її.

Управління роботом може здійснюватися як людиною-оператором, так і системою управління промисловим підприємством (ERP-системою), що погоджують дії робота з готовністю заготовок і верстатів з ЧПК до виконання технологічних операцій[7].

Серед найпоширеніших дій, що здійснюються промисловими роботами можна назвати такі:

- переміщення деталей і заготовок від верстата до верстата або від верстата до систем змінних палет;
- шовне та точкове зварювання;
- фарбування;
- виконання операцій різання з рухом інструменту по складній траєкторії.

Промисловий робот є пристроєм, що проводить якісь маніпулятивні функції, схожі з функціями руки людини[10].

«Робот» взагалі — поняття невизначене, і тому до класу роботів можна віднести багато автоматичних пристроїв.

Промислові роботи є важливими компонентами автоматизованих гнучких виробничих систем, які дозволяють збільшити продуктивність праці. На жаль, повну і економічно виправдану автоматизацію виробництв ще не досягнуто[5].

Роботи, що працюють в спеціальних умовах (висока радіація, тиск, температура, підводний світ, космос) дозволяють, не ризикуючи життям людей, здійснювати різні операції як дослідницького, так і рятувального і антитерористичного характеру[21].

У складі робота є механічна частина і система управління цією механічною частиною, яка у свою чергу отримує сигнали від сенсорної частини. Механічна частина робота ділиться на маніпуляційну систему і систему пересування[9].



Рис.1.1- Навчальний робот-маніпулятор SCORBOT-ER 4u

Маніпулятори включають рухомі ланки двох типів:

- ланки, що забезпечують поступальний рух;
- ланки, що забезпечують кутові (повертальні) переміщення.

Поєднання і взаємне розташування ланок визначає кількість ступеней вільності, а також область дії маніпуляційної системи робота[19].

Для забезпечення руху в ланках можуть використовуватися електричний, гідравлічний або пневматичний привод[11].

Частиною маніпуляторів (хоч і необов'язковою) є захоплювальні пристрої. Найуніверсальнішими такими пристроями є пристрої-аналоги руки людини, у яких захоплення здійснюється за допомогою механічних «пальців». Для захоплення плоских предметів використовуються захоплювачі з пневматичним присмоктуванням[18]. Для захоплення ж множини однотипних деталей (що зазвичай і відбувається при застосуванні роботів в промисловості) застосовують спеціалізовані конструкції.

Замість захоплювальних пристроїв маніпулятор може бути оснащений робочим інструментом. Це може бути пульверизатор, зварювальна головка, викрутка тощо.

Система пересування роботів

Для пересування по відкритій місцевості найчастіше використовують колісну або гусеничну, рідше — крокуючу систему пересування роботів. Це найуніверсальні види систем переміщення[4].



Рис.1.2 - Робот на гусеничному ході

Для нерівних поверхонь створюються гібридні конструкції, що поєднують колісний або гусеничний хід з складною кінематикою руху коліс. Така конструкція була застосована в місяцеході.

Усередині приміщень, на промислових об'єктах використовуються пересування уздовж монорельсів, по підлоговій колії тощо.

Для переміщення по похилих, вертикальних площинах використовуються системи аналогічні «крокуючим» конструкціям, але з пневматичними присосками[26].

Система керування роботами

Системи керування робототехнічними пристроями будуються на тому самому технічному базисі, що і всі інші автоматичні пристрої[22]. Алгоритми систем керування роботами вивчаються у курсах теорії автоматичного керування, теоретичної механіки. Звичайно це лише найзагальніші курси, для глибшого вивчення рекомендується теорія навігаційних систем, наближена теорія гіроскопів, електротехніка, цифрова та аналогова схемотехніка та ін[41].

У сучасному світі, коли домінуючою течією є конвергенція технологій та наук, у керуванні роботами цікавих результатів досягають застосуванням знань людства з біології та моделей поведінки тварин.

Загально визнаним є поділ методів керування на: програмне керування, адаптивне керування, інтелектуальне керування, напівавтоматичне або телекерування — тобто за участі людини.

Програмне керування

Програмне керування — найпростіший тип системи керування, використовується для управління маніпуляторами на промислових об'єктах. У таких роботах відсутня сенсорна частина, всі дії жорстко фіксовані і регулярно повторюються[24].

Адаптивне керування

Роботи з адаптивною системою керування оснащені сенсорною частиною. Сигнали, що передаються сенсорами, аналізуються і залежно від результатів ухвалюється рішення про подальші дії, перехід до наступної стадії дій тощо[22].

Інтелектуальне керування

Інтелектуальний спосіб керування заснований на методах штучного інтелекту. Серед яких можна виділити, як найпоширеніші — нечітку логіку та нейронні мережі.

Управління за участі людини

Управління роботом людиною за допомогою дистанційних систем керування. Прикладом такого робота є апарат для розмінування з дистанційним керуванням[2].

- Робот-гуманоїд

Робот-гуманоїд (людиноподібний робот)— це машина, шасі (ходова частина) якої виконано у вигляді людиноподібного тіла. Гуманоїдний дизайн робота зумовлений певною метою: функціональністю — для використання людських інструментів чи середовищ життя людини; з експериментальною метою — для вивчення прямоходіння; з медичною метою — вивчення впливу на організм тих чи інших навантажень тощо. Загалом, людиноподібні роботи мають тулуб, голову, дві руки і дві ноги; хоча деякі види людиноподібних роботів можуть моделювати тільки частину тіла, наприклад, від голови — до пояса. Деякі людиноподібні роботи можуть мати голову, призначену для реплікації людських рис обличчя (таких, як очі і рот тощо). Прикладом може бути HRP-4C.

HRP-4C — японський людиноподібний робот, що є втілення середньостатистичної молодої жінки 19-29 років. Створений в Японському національному інституті наук і технологій (AIST) спільно з Kawada Industries та Kokoro Co. Ltd. [1], перша його презентація відбулась у 2009 році. Важить HRP-4C (разом з батареями) 43 кг, а її зріст становить 158 см. Була протестована як модель на подіумі, та як поп-співачка^[2]. Робота оснащено великою кількістю моторів, які

забезпечують ходу і рух руками. Окрім того, вона розуміє людську мову і сама підтримує розмову, а завдяки вживленим під «шкіру» обличчя і шиї мікромоторчикам HRP-4C вміє демонструвати різні вирази людських почуттів[22].



Рис.1.3 - людиноподібний робот HRP-4C

- Індустріальний робот

Промислові (індустріальні) роботи в останні десятиріччя майже повністю замінили людську працю в різноманітних галузях, особливо в технологічних процесах, де потрібна прецизійна точність, швидкість та одноманітність, повторюваність операцій — в машинобудуванні та обробці матеріалів, у виробництві мікропроцесорів і навіть в таких технологіях як виробництво або складання паперово-картонної тари[22].

- Сільськогосподарський робот

Сільськогосподарський робот або агроробот — робот, який використовується у сільськогосподарських цілях.

Основна область застосування роботів у сільському господарстві — процес збирання врожаю[10]. Роботи, що збирають овочі або фрукти; трактор-розпилювач, що працює автономно; робот, який стриже овець, призначені для заміни людської праці у цій галузі. Індустрія сільського господарства відстає у використанні роботів від інших галузей, так як види робіт, пов'язані з сільським господарством, не є однозначними, і навіть часто повторювані дії кожного разу не завжди збігаються. У більшості випадків певна множина факторів (наприклад, розмір і колір плодів) повинна аналізуватись до початку виконання завдання[12].

Роботи можуть бути використані при вирішенні таких рослинницьких задач, як обрізання, прополювання, оранка, поливання, моніторинг перебігу процесу вирощування тощо. У тваринництві роботи знайшли застосування для доїння, миття, кастрації тощо[22].



Рис.1.4 - «BoniRob» — прототип автономного польового робота компанії «Amazonen-Werke»

- Побутовий робот

Побутовий робот — робот, призначений для допомоги людині в повсякденному житті. Наразі поширення побутових роботів є невеликим, проте футурологи передбачають широке їх використання у найближчому майбутньому[24].

Відомі такі комерційні моделі побутових роботів:

- роботи-іграшки;
- соціальні роботи, які автономному чи напівавтономному режимі можуть взаємодіяти та спілкуватись з людьми;
- роботи-помічники, наприклад:
 - роботи-прибиральники (робот-пилосос, робот для миття підлоги тощо)
 - роботизовані газонокосарки;
 - роботи для чищення басейнів, каналізаційних труб тощо.

З початку 2000-х років багато компаній розпочали виробництво «роботизованих порохотягів», такі як Electrolux Trilobite^[en], Roomba^[en], Robomaxx, Samsung Navibot, FloorBot та ін. Пристрій найчастіше являє собою диск діаметром 28-35 см і висотою 9-13 см. Переважно великий

контактний сенсор встановлений в передній частині пристрою, з інфрачервоним датчиком по центру у верхній передній частині. Залежно від моделі, пристрої можуть бути оснащені інфрачервоними датчиками «віртуальна стіна». Перше і друге покоління моделей потребувало ручного вказування площі кімнати, у моделях третього покоління такої необхідності немає. Для роботи робот-порохотяг використовує внутрішні акумулятори, тому потребує регулярної підзарядки[27]. Нові моделі мають можливість самостійно знайти зарядний пристрій, зв'язуючись з ним через інфрачервоний датчик. Тривалість зарядки пристроїв першого і другого покоління — близько 12 годин, третього покоління — близько 3 годин. Роботи-порохотяги, як правило, мають невелику висоту, досить низьку, щоб пройти під ліжком або іншими меблями. Коли робот-порохотяг розуміє, що він застряг, він перестає рухатися і починає подавати звукові сигнали, які допомагають власнику його виявити.



Рис.1.5 - Робот-пилосос iRobot Roomba (2005)

- Військовий робот

Бойовий робот «Swords» («мечі») — скор. від англ. Special Weapons Observation Reconnaissance Detection Systems (спеціальна бойова система спостереження та розвідки).

SWORDS (скорочення від Special Weapons Observation Reconnaissance Detection Systems) - спеціальна бойова система спостереження і розвідки.



Рис.1.6 - Бойовий робот «Swords»

Створено компанією Фостер-Міллер TALON Робот (надалі стала підрозділом QinetiQ). За твердженням виробника, робот призначений для дій в місті, здатний долати пісок, воду і сніг до 1 фути (це становить 0,3048 м [1]) глибини[2], а також здійснювати підйом по сходах.

Розрахований на 8,5 годин роботи від батарей в нормальному експлуатаційному режимі - очікування до 7 діб. Контролюється оператором на відстані до 1000 метрів. Він важить близько 100 фунтів (45 кг) або 60 фунтів (27 кг) у версії для розвідки.

Є цілий ряд різних видів зброї, які можуть бути розміщені на SWORDS: гвинтівки M16, 5,56-мм SAW M249, 7,62 мм кулемет M240, гвинтівки Barrett M82, шестиствольний 40-мм гранатомет або чотиріхствольний 66 мм M202A1 FLASH.

Застосовувався в Афганістані і Іраку.

Дослідний Центр Військово-морського флоту США опублікував концепцію використання бойових роботів^[9] На відміну від попередніх документів такого роду, що готуються з 1960-х років, у цьому вперше запропоновано кардинально новий принцип: машини повинні знищувати тільки інші машини[26].

У документі, що називається Концепція Операцій Збройних Автономних Систем підкреслюється, що за нинішніх умов головним завданням армій є не виграш війни, а виграш миру. У цих умовах украй небезпечно використовувати бойову техніку, що апріорі небезпечна для мирного населення. Знищення мирних жителів у ході легітимних воєнних операцій є практично неминучим. Небезпека убивства безневинних зростає у разі, якщо в бій йдуть бойові роботи, не здатні відрізнити, наприклад, бойовика з вибуховим пристроєм у руці від жінки з відром[12].

Виходячи з цього, автори концепції пропонують від самого початку налаштовувати роботів на боротьбу не з носіями зброї, а із самою зброєю. Наприклад, з появою супротивника, озброєного автоматом, робот повинен цілитися не в людину, а в автомат. Якщо при цьому буде убито або поранено ворожого солдата чи повстанця, подібні огріхи можна буде розцінювати, як нестрашну помилку.

1.2 Аналіз навігації мобільного робота

Одним з актуальних завдань робототехніки є завдання навігації робота в просторі, тобто аналіз ситуації і вибір маршруту самим роботом без участі людини[2].

До теперішнього часу успішно вирішені завдання глобальної навігації, але її використання стає проблематичним, коли треба щоб робот орієнтувався усередині приміщень. Взагалі робота усередині закритих приміщень характеризується безліччю найрізноманітніших перешкод, починаючи від нерівномірності освітлення і закінчуючи проблемами відображення радіосигналів або, навпаки, утрудненнями при реалізації каналів зв'язку. У такому разі середовище вважається погано обумовленим, з ненадійними каналами зв'язку, з принциповою неточністю і невизначеністю. Тоді очевидною стає необхідність дослідження способів орієнтування, здатних працювати в сильно зашумленій середовищі[1].

Система навігації складається власне з мобільного робота і комплексу маяків. За кожним маяком жорстко закріплений його номер (сигнатура з п'яти символів).

Щоб підвищити завадостійкість переданого сигналу від маяків до ІК-приймача, використовується система кодування каналу. Для збільшення ймовірності безпомилкового розпізнавання сигналу на приймальній стороні в умовах зашумленою середовища використовується метод переходу до широкосмугового сигналу, додаючи надмірність у вихідний сигнал. Для цього замість одного переданого інформаційного біта треба передавати певний код (послідовність так званих чіпів) [2].

Від вибору кодової послідовності залежить завадостійкість системи. При однакових довжинах, властивості послідовностей можуть кардинально відрізнятися.

Після дослідження властивостей послідовностей була обрана кодування сигналу з допомогою M-послідовності довжиною 31 символ, оскільки з її

допомогою можна передавати зі стовідсотковою ймовірністю сигнал з 20% зашумлених даних[3].

Конструктивно система вирішена так, що приймач ІЧ – сигналу розташовується на роботі, на осі крокового двигуна, що обертає приймач навколо своєї осі. На рис. 1.7 а показана система навігації мобільного робота. Обертовий приймач дозволяє роботів визначити кути α_1 , α_2 , α_3 на маяки щодо своєї осі. Враховуючи, що координати маяків відомі роботу заздалегідь, отримана інформація дозволяє розрахувати координати робота і обчислити кут α – кут між напрямком робота і системою координат. На рис. 1.7 б показана схема системи навігації мобільного робота[11].

Обчислити координати робота можна за допомогою різних геометричних методів. В роботі використовується т. зв. метод кіл. Подібні методи дозволяють визначити положення робота при точно заданих кутах орієнтації на маяки α_1 , α_2 , α_3 . Але так як ці кути визначаються з деякими похибками Δ_1 , Δ_2 , Δ_3 відповідно, кути орієнтації являють собою інтервали $[\alpha_1 - \Delta_1, \alpha_1 + \Delta_1]$, $[\alpha_2 - \Delta_2, \alpha_2 + \Delta_2]$, $[\alpha_3 - \Delta_3, \alpha_3 + \Delta_3]$. Використання інтервальних обчислень в даному методі призводить до того, що область становища робота визначається з надмірно великою похибкою.

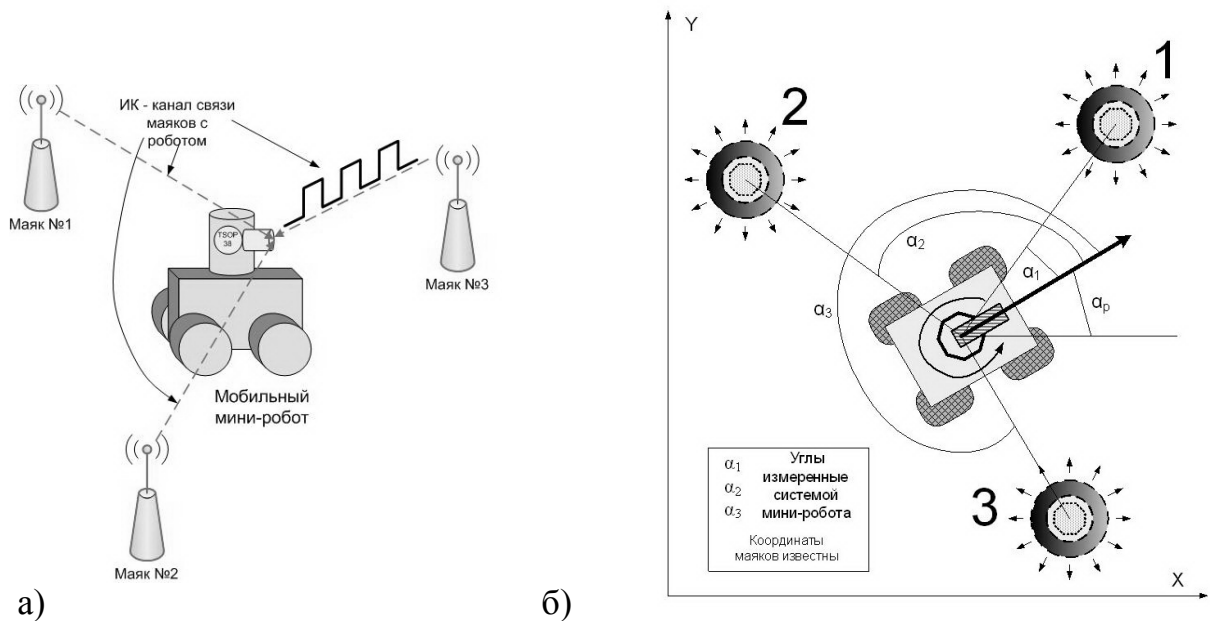


Рис.1.7 - (а) Загальний вигляд мобільного робототехнічної системи
 (б) Схема системи навігації мобільного робота

Для звуження області визначення робота виявилось доцільним застосувати один з ітераційних обчислювальних методів - метод недовизначених обчислень [9]. Метод недовизначених обчислень (моделей) був запропонований А. С. Нариньяні ще на початку 80-х років минулого століття і є досить ефективною технологією вирішення завдань задоволення обмежень у самій загальній постановці. Параметри реальних завдань завжди мають початкові оцінки їх значень. Так як кути орієнтації на маяки представлені інтервалами, це дозволяє задати сектор, що включає справжнє становище робота з кожного маяка[2].

Отже, кожен символ сигнатури маяка кодується 31 символами М-послідовності, 1 – її прямим кодом, 0 – інверсним. Далі, для можливості прийому ІК-сигналу елементом TSOP1738, кожна одиниця замінюється пачкою з п'ятнадцяти імпульсів з частотою 38 кГц. На Рис 1.8 - представлений вид кодування сигнатури.

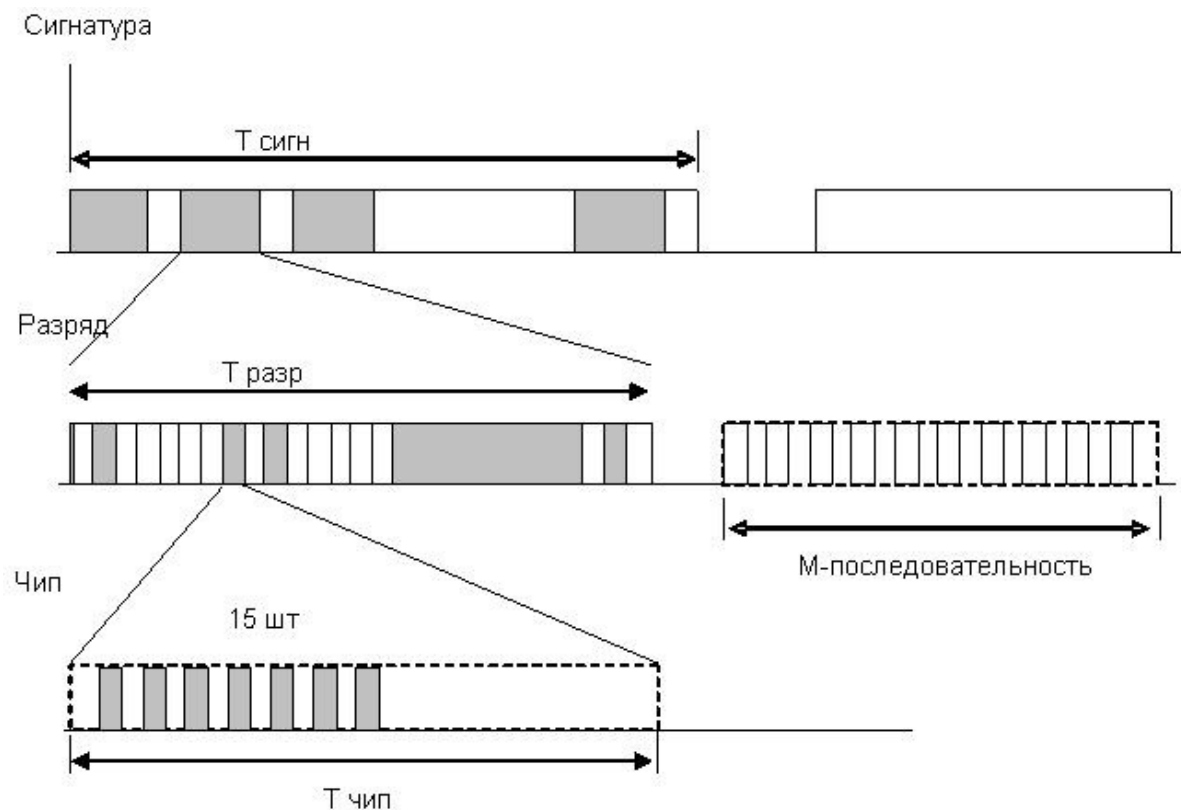


Рис.1.8 - Вид кодування сигнатури на передачу

Для виключення расфазирования сигнал від маяків приймається з подвоєною частотою.

Прийняті чіпи зберігаються в 32-х розрядному буфері (4 регістра). Також є буфер, що зберігає в собі М-послідовність, що кодує нуль (еталон нуля).

Лічені або поміщаються в буфер, при кожному новому прийомі символу, вміст буфера зсувається на один розряд, фактично це черга довжиною 31, через яку проходять всі прийняті символи. При кожному новому прийомі символу вміст буфера складається по модулю 2 (XOR) з буфером, що зберігають нульову М-послідовність.

Довжина послідовності , тобто вона поміщається цілком у чотирьох регістрах. Після додавання в регістрах залишаються одиниці на тих позиціях, на яких буфер не збігся з еталоном нуля. Тобто кількість символів, які співпали з еталоном одиниці[25].

Дуже важливим є швидкість підрахунку цих одиниць. Якщо йти в циклі по 8 розрядів у 4-х регістрах (з допомогою маски вирізаючи по одному біту) і скласти одиниці, виходить дуже довго. У результаті переривання не встигає завершитися.

Для більш точного визначення кута, приймач серії TSOP повинен приймати сигнал від маяків через вузьку щілину. Була розроблена система спостереження, яка дозволяє крокового двигуна постійно обертатися навколо своєї осі, а приймача, пов'язаний з мікроконтролером через дріт, перебувати на місці. Також система повністю захищена від попадання сигналу не через щілину прийому. На Рис.1.9 - представлена схема її конструкції і зовнішній вигляд[32].

Велика частина конструкції закріплена на кроковому двигуні й обертається разом із ним. Через щілину всередину проходить сигнал від маяка. У середині квадрата він переломлюється від дзеркала, закріпленого під кутом 45° , і надходить нагору на елемент серії TSOP. Так як, TSOP пов'язаний з мікроконтролером проводами, він не може обертатися разом з кроковим двигуном. Він закріплений усередині циліндра, вставленого у обертову конструкцію.

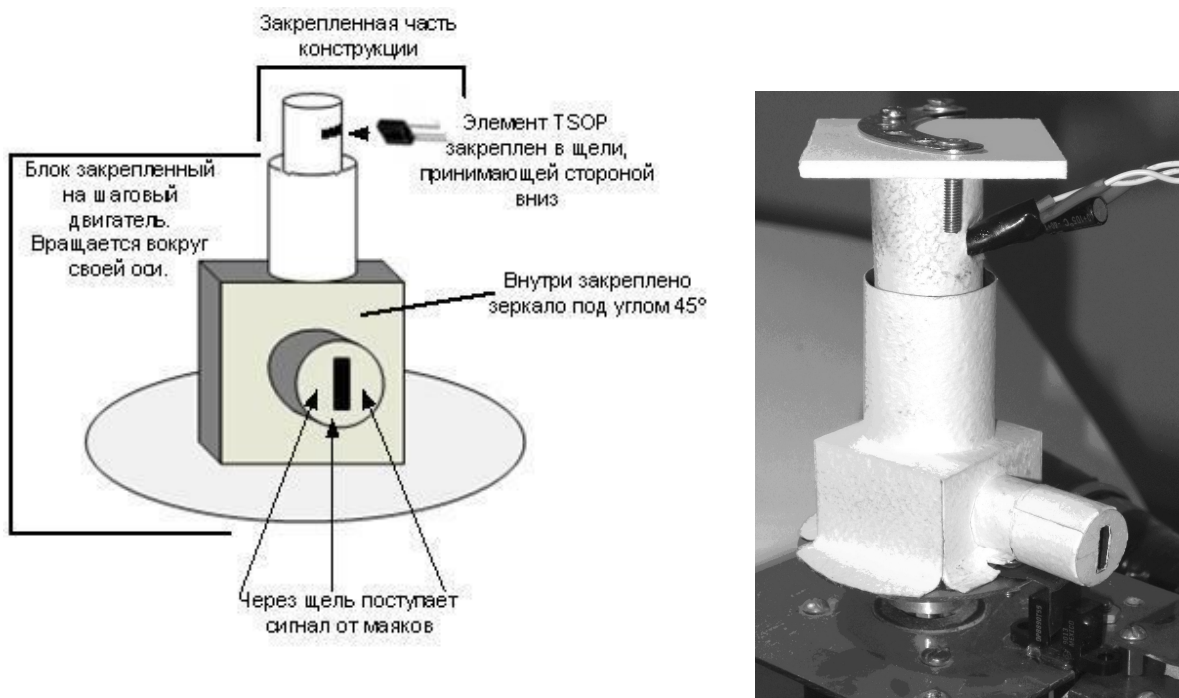


Рис.1.9 - Оптична частина адміністратора системи

Конструкція на кроковому двигуні не може забезпечити дуже вузький отвір прийому сигналу від маяка, щоб приймати одну сигнатуру за період обігу, всі прийняті кути від маяків записуються в масив. Після обороту крокового двигуна цей масив треба усереднити, але якщо в масив вкрався випадково прийнятий сигнал, він може кардинально змінити середнє число. Для цього треба виключити аномальні результати із загальної маси типових результатів, застосувавши статистичну оцінку[21].

Після визначення кутів, і знаючи координати маяків, обчислюються координати робота. Система побудована на базі мікроконтролерів AtMega8 і AtMega162. Маяки побудовані на базі мікроконтролерів AtMega8. Для написання програм для контролерів використовувався мова високого рівня – Сі. Така система навігації є мобільною, вона розгортається на місцевості шляхом встановлення трьох маяків, також ця система є перешкодозахищеною.

1.3 Завдання проекту

Зважаючи на тему моєї дипломної роботи основне завдання є зробити програмний модуль для навігації робота, що уникає перешкоди, для цього необхідно розглянути систему управління і апаратну систему, яка робить навігацію мобільного робота можливою в своєму середовищі.

Апаратна система, що складається з цифрових і аналогових датчиків, які використовуються для виявлення перешкоди і взаємодії з навколишнім середовищем впродовж його навігації, силовий привід використовує інформацію, наданого йому контролером, мікроконтролери, які запрограмовані інструкцією управління та організовує введення і виведення дій мобільного робота[14].

У мобільній робототехніці однією з основних областей, що викликають заклопотаність в тому, як управляти рухами робота, що заставляє робота рухатися, не зачепивши ніяких перешкод. Алгоритм прийняття цих рішень називається алгоритмом обходу перешкод.

Мета моєї дипломної роботи - це знайти спосіб поліпшення алгоритму обходу перешкод, який використовується в системі з рухомим роботом. Дивлячись на динаміку рухомого робота, новий спосіб управління роботами було запропоновано в даній роботі[19].

Для досягнення мети дипломної роботи будуть виникати деякі проблеми.такі як:

- можливість мобільного робота взаємодіяти з навколишнім середовищем. А також знати свою позицію в навколишньому середовищі [12];
- можливість мобільного робота досягти своєї цільової точки, виявлення перешкод на цьому шляху навігації, уникнути зіткнення з перешкодами на шляху навігації в даному середовищі [13];
- здатність робота ефективно досягти своєї цільової точки;

Беручи до уваги пункти важливих аспектів контролю і технічної реалізації та дизайн для мобільного робота. Основне завдання, яке необхідно вирішити в моїй дипломній роботі є розробка і впровадження програмного модуля для навігації мобільного робота, уникаючи зіткнень. Досягнення мети забезпечують наступні завдання:

- розгляд вже відомих рішень і забезпечення управління роботизованої системи;
- аналіз мобільного робота навігації
- пояснювати існуючі алгоритми для виявлення перешкод і запобігання зіткнення мобільного робота;;
- створення сценарію (реалізації), щоб пояснити алгоритм обходу перешкод;
- пояснювати вибір розробки програмного забезпечення;
- експериментування і моделювання.

II. АЛГОРИТМИ І МЕТОД НАВІГАЦІЇ

2.1 Відображення існуючого алгоритму

Уникнення перешкод є ключовим компонентом будь-якої мобільної роботизованої системи. Ці перешкоди можуть бути або статичними, як, наприклад, стіл або динамічними, як, наприклад, люди або інші роботи. Що ми говоримо про уникання перешкод - це є здатність переходити від одного місця до іншого, не стикаючись з речами на шляху[9]. Це означає, що ухилення від перешкод тісно пов'язане з плануванням шляху, це не зовсім те ж саме, але вони залежать один від одного настільки, що часто не відрізняють. Загалом, є два види методів планування шляху щоб уникнути перешкоди, глобальний і локальний. Вони засновані на тих же ідеях, структури, згаданих раніше, глобальний алгоритм запобігання перешкод вимагає відносно повного знання про світ і може обчислювати найкращий шлях від початку до кінцевої цілі в автономному режимі. Оскільки це практично неможливо в динамічному світі, інша частина цього розділу буде зосереджена тільки на методах, які мають відношення до локальної парадигми. Ці методи розглядають тільки те, що вони "бачать" і діють відповідним чином[4].

2.1.1 Потенційні поля

Багато з методів ухилення від перешкод засновані на ідеї про потенційні полі. Потенціал поля є спосіб подачі навігаційного простору у вигляді подразників, які одночасно впливають на роботу. Цей метод досягнення цілі, як притягувати так і відштовхуватись від перешкод, які надають сила тяжіння або відштовхування від роботи[50]. Для мети, сила вирівнюється по всьому навігаційному просторі, незалежно від того, де робот зупинився, він може "відчувати тяжіння" до цілі. Для перешкоди, сила присутня тільки в його середовищі. Класична функція, яка використовується для розрахунку цієї сили є

закон всесвітнього тяжіння; сила тяжіння зменшується зі збільшенням відстані між роботом і об'єктом. Для кожного об'єкта в сенсорному діапазоні робота, окреме поле обчислюється з використанням потенційної функції. Ці поля потім об'єднуються з утворенням єдиного глобального поля. Це показано на малюнку 2.1.

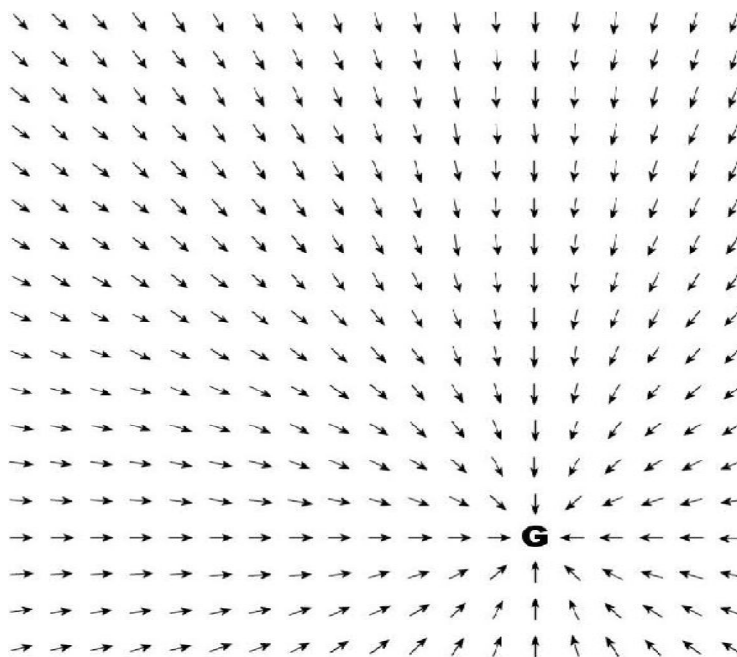


Рис.2.1 - Потенційна функція цілі тут показана на всій карті

Зазвичай вона обчислюється тільки для місця положення, в якому робот знаходиться. Проблема з потенційними полями є кількість часу, необхідного для обчислення всієї сили поля[48]. Щоб уникнути цього, необхідні обчислення сили, які впливають на стан, в якому робот знаходиться на даний час. Це призводить до того, що ніякого глобального планування шляху не проводиться для всіх, робот просто реагує на об'єкти, що знаходяться поблизу, що призводить до дуже швидкого методу обходу перешкод. На жаль, є й інші проблеми, пов'язані з потенційними полями; Корен і Боренштайн [17], що визначили деякі проблеми, які притаманні потенційним полям, тобто вони існують у всіх реалізаціях методу. Проблеми: пастка ситуації, через обмежене місце зчитування інформації.

Пастка ситуації може статися, коли робот потрапив в U-подібну перешкоду. Сила спрямована до мети буде в якийсь момент всередині перешкоди дорівнювати силі, спрямованої в сторону від перешкоди. Так як існують перешкоди з боку робота, а робот не може просто повернути біля перешкоди і, ймовірно, буде в пастці. Пастка ситуації може бути вирішена шляхом глобального відновлення, тобто робот повинен розпізнати ситуацію і відновити попередні відстеження свого пройденого шляху, поки він не зможе знайти інший спосіб.

Або неможливість проходу між близько розташованими перешкодами. Це завдання аналогічне попередньому, якщо дві перешкоди розташовані близько одна до одної, як дверного отвору, сили відштовхування від кожного перешкоди об'єднані в одну. Сила відштовхування, яка вказує в сторону від отвору між перешкодами. Робот буде відвертатися від відкриття, навіть, якщо мета знаходиться на іншій стороні. Вібрація при наявності перешкод і в вузьких проходах

Боренштайн і Корен також показали, що робот іноді буде гойдатися в наявності перешкод. Коли робот рухається уздовж осьової лінії між двома стінами, рух стійкий. Якщо, проте, робот буде наближатись ближче до однієї зі стін, він рушити назад до середньої лінії за допомогою сили відштовхування від найближчої стінки. Робот тепер буде відчувати силу відштовхування від іншої стіни і бути відкинутий до осьової лінії. Така поведінка буде потім продовжуватись і робот буде гойдатись між двома стінками.

Реалізація потенційного підходу поля є Віртуальний метод силового поля [5]. На жаль, він страждав від проблем, перерахованих вище і, незабаром, був замінений метод гістограм векторних полів, описаної в наступному розділі.

2.1.2. Метод гістограм векторних полів

Після виявлення проблем з методами потенційного поля, (Корен і Боренштайн [17]) продовжили свою роботу, запропонувавши метод, який має

справу з недоліками потенційних полів. В своїй роботі [6] вони пропонують швидкий спосіб уникнення перешкод, і називають метод гістограм векторного полів (VFH). Метод, який лише коротко обговорюється тут, так як він не прямо пов'язаний з алгоритмом вивченого в даній роботі. Це, однак, важливий метод для уникнення перешкод і, отже, варто згадати.

Метод VFH використовує сітку гістограми, описану вище, щоб представити навколишнє середовище, в якій робот переміщається. Це робить карту досить точно, навіть коли робот оснащений гідролокатором. Вміст кожного осередку в сітці гістограми розглядаються в якості вектора перешкоди; напрямок вектора є направлення від поточного положення робота до перешкоди. Величина вектора визначається за формулою, яка враховує значення визначеності осередку і відстань до осередка (зверніть увагу, що зі значенням визначеності нуля величина вектора перешкод також буде дорівнює нулю) [43].

Далі, область навколо робота ділиться на ряд рівних секторів. У кожному секторі на перешкоди, вектори цього сектора підсумовуються, щоб дати провідну щільність перешкоди (POD).

Експериментальні результати цього методу показали, що більшу частину часу робот пройдене на максимальній швидкості (0,78 м / с), не зупиняючись для того, щоб обчислити безпечний шлях. Швидкість була знижена тільки тоді, коли перешкода підійшла дуже близько. В ході цих експериментів, 20-МГц 80386 комп'ютер побіг алгоритм VFH [6]. Є, однак, деякі серйозні проблеми з цим методом.

Динамічне обмеження

Річ у тому, що всі раніше описані системи мають спільне, що всі вони не беруть до уваги динамічні обмеження, що накладаються на що-небудь з вагою. Робот, який рухається на повній швидкості, просто не може зробити 90-градусний поворот. Навіть якщо він не впаде, це, безумовно, не буде залишатися на розрахунковій кривизни. Він також не може зупинитися відразу, якщо перешкода

раптово з'явилася перед ним, за законами динаміки робот буде рухатися навіть якщо колеса перестануть обертатися. В роботі [21] методу VFH доповнюється беручи до уваги динаміку. Динамічні обмеження, що накладаються на робота будуть блокувати деякі з секторів, які були відзначені як вільні для поїздки в оригінальному методі.

Установка параметрів

Іншою серйозною проблемою є налаштування параметрів, це проблема, що більшість алгоритмів з якою доводиться мати справу. У методі VFH поріг, коли сектор є економним на поїздки, визначає поведінку робота. Занадто низький поріг зробить робота не в змозі рухатися за наявності перешкод, в той час як занадто великий поріг змусить робота врзатися в перешкоду. Це робить налаштування цього параметра важким, є відсутність інтуїтивної інтерпретації.

У відповідь на проблеми, згадані вище, Бауер та ін [4] уявити, що вони називають підхід Steer Angle Field (керування кута поля). Пізніше на двох подібних методів були запропоновані, метод викривлення швидкості (Curvature-Velocity Method) [20] і динамічного вікна (Dynamic Window).

2.1.3 Підхід Global Dynamic Window

Оскільки DWA не має ніякої інформації про зв'язності вільного простору воно, точно так само, як і потенційні методи в польових умовах, сприйнятливий до локальних мінімумів. Брок і Хатіб [9] запропонували метод обходу перешкод, Global Dynamic Window підхід GDWA, який розширює DWA і Holonomic DWA за допомогою алгоритму простого руху планування, тобто функція, яка знаходить траєкторію руху без зіткнень від стартової конфігурації до конфігурації мети. Алгоритм планування руху досить швидкий, щоб бути обчислений в кожному повторенні управління петлі, що дозволяє роботу реагувати в режимі реального часу, при цьому використовуючи глобальне планування[39].

Вільний простір підключення

Для того, щоб спланувати шлях у навколишньому середовищі, він повинен бути змодельований, якимось чином так, що робот може витягувати інформацію про зв'язок вільного простору від моделі.

GDWA не використовує апріорне знання навколишнього середовища, але його зразки, використовуючи датчики робота, і запобіжну інформацію датчика в зайнятості сітки. Кожне читання від датчиків, розглядається як точка в R^2 , де міститься перешкода. Дані переведені в конфігураційному просторі шляхом розширення перешкоди з радіусом робота. Це означає, що робот може бути представлений точкою, і це робить шлях планування набагато простіше, якщо є незайнята клітина, то робот буде вписуватися в неї. Такий підхід передбачає, що робот має круглу форму, в іншому випадку необхідне використання іншого методу[37].

Функція навігації

Ідея навігаційної функції, щоб знайти кращий шлях від робота до поставленої цілі. Кращий шлях тут визначається як найкоротший шлях, який не обов'язково збігається з найшвидшим шляхом. GDWA використовує глобальні, локальні мінімуми вільного плавання функції NF1 [18] для пошуку простіру конфігурації найкоротшого шляху. Це робиться шляхом обчислення відстані Манхеттена, від мети клітини до робота але має бути найкоротший шлях; почніть з початку підбити всі клітинки до великого значення і дати наміченій клітинці значення 0, віддавати кожному 1 - сусіду від наміченого сектора, який знаходиться у вільному просторі значення 1. Треба продовжити, даючи кожній клітині у вільному просторі своїх батьків значення плюс один поки не залишилося вільних клітин. Все C-космічні перешкоди тепер матимуть значення, що всі клітини були початі в той час як всі клітини в вільному просторі матимуть своє значення L1 до намеченої. Прикладом цього можна побачити на малюнку 2 і малюнку 3. Проблема, пов'язана з NF1 є те, що він зачіпає C-перешкоди. Цього можна

уникнути, в GDWA лише вибираючи команди руху, які дають роботу мінімальний зазор до перешкод[5].

Оскільки орендована сітка перелічується в кожному циклі керування петлею, додані нові перешкоди, а старі видалені; NF1 повинен бути перерахований. Це було б дуже дорого, якщо NF1 була обчислена для всієї зайнятої сітки, але GDWA використовує інший підхід. NF1 обчислюється тільки в відповідному прямокутнику з метою курсу. Якщо алгоритм виявляє, що існує перешкода, блокується його шлях, ширина прямокутника збільшується до тих пір, доки робот не досягне необхідний фронт.

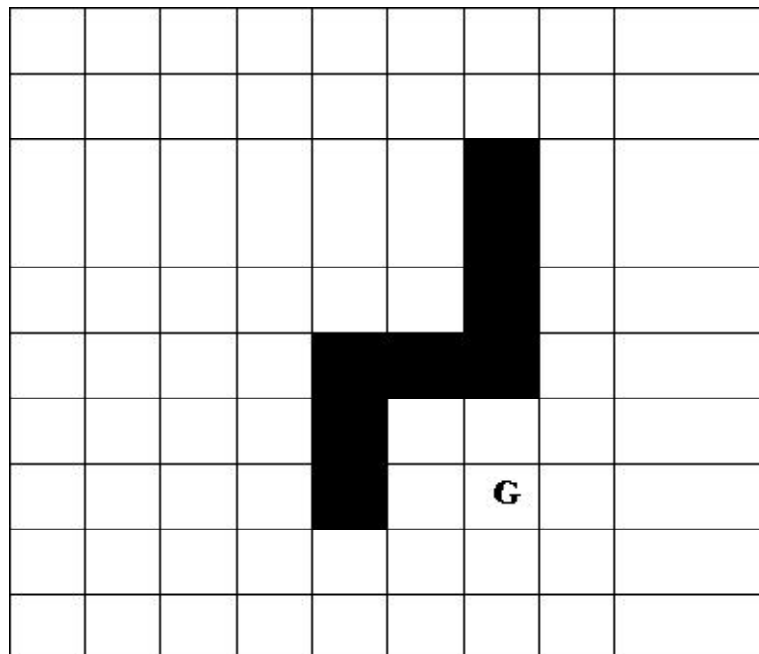


Рис.2.2 -. Ця цифра показує невелику дискретну сітку-карту 10x10 клітин

Чорна клітина на карті є перешкодами і мета осередок позначається "G"

15	14	13	12	11	10	9	8	9	10
14	13	12	11	10	9	8	7	8	9
13	12	11	10	11	10	999	6	7	8
12	11	10	9	10	11	999	5	6	7
11	10	9	8	9	10	999	4	5	6
10	9	8	7	999	999	999	3	4	5
9	8	7	6	999	2	1	2	3	4
8	7	6	5	999	1	0	1	2	3
7	6	5	4	3	2	1	2	3	4
8	7	6	5	4	3	2	3	4	5

Рис.2.3 - Ця цифра показує Манхеттен відстань від мети до будь-якої іншої клітини в карті

Мета, очевидно, має нульову відстань для себе, і він позначений "0", а перешкоди не розраховані таким чином, вони мають велике значення, яке гарантовано не буде досягнуто в якійсь комірці у вільному просторі. Шлях з будь-якої клітини до мети розраховується шляхом пройденого відрізка до секції з найменшим значенням, яке знаходиться поруч з нинішнім. При використанні NF1 як алгоритм шляху планування, важливо розглянути питання про те, що шлях буде зрізати кути перешкод.

Цільова функція

Як і в динамічному вікні наближення до GDWA використовує цільову функцію вибору між усіма можливими рухами, які він може зробити. Це в значній частині так само, як і для DWA, але з важливою відмінністю, є додавання глобальної навігаційної функції.

Присутні суворі вимоги в алгоритмі уникнення перешкод.. Звичайно, не доводиться стикатися з жодним алгоритмом уникнення перешкод. Це ставить високі вимоги до точності і швидкості виявлення нових перешкод. Алгоритм не повинен дозволити роботу стояти на місці, якщо є шлях подорожувати. Тільки якщо кожен можливий шлях заблокований, тільки тоді робот повинен зупинитись.

Це також вимагає, щоб карта була точною і сприяє безліч спрощень. Якщо робот має овальну форму вона не повинна бути змодельована як коло, так як ці додаткові кілька сантиметрів можуть дозволити йому проходити вузький шлях. Робот повинен також мати можливість подорожувати як можна швидше. Швидкість може бути обмежена з кількох причин. Це ставить вимоги на вибір алгоритму, а також вимагає довіри до карти, щоб бути точним.

З обговорень в цьому розділі має бути ясно, що GDWA є найкращим способом з описаних. Вона також задовольняє вимогам, описаним вище. За допомогою гістограмної сітки для відображення є швидким і надійним у виявленні динамічних перешкод[27]. Справедливо реалізований алгоритм запобігає зіткнення, оскільки ці швидкості повинні бути позначені як неприпустиме. Він не стоїть на місці, якщо є якийсь шлях, який не є заблокований, так як висока швидкість є одним із завдань функції.

2.1.4 Алгоритм зірки

Алгоритм зірки базується на пошуку методу процесу, в якому оцінюється, що є кращим центром переплетіння для пошуку наступного. Функція використовується для вибору наступного вузла можна побачити в рівнянні 2.1.

$$f(n) = g(n) + h(n) \quad (2.1)$$

$f(n)$ загальна вартість шляху від початкового вузла до вузла мети через вузол n , $g(n)$ є вартість переходу від вихідного вузла до вузла n і $h(n)$ є наближеним

вартості переходу від вузла n до мети вузла. При цьому вузли з меншою кількістю загальної довжини шляху є кращими, щоб вивчити, який подальший вузол показано на малюнку 4.

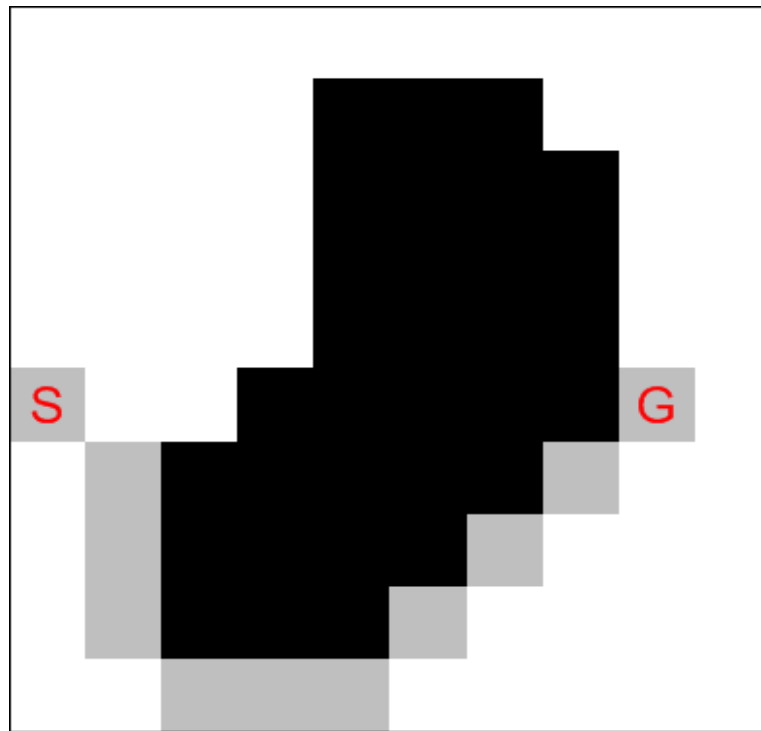


Рис.2.4 - Приклад шляху, отриманий з використанням алгоритму зірки

2.2 Алгоритм використання псевдокоду

Ця частина розділу буде використовуватися для пояснення алгоритму, як він працює в середовищі з перешкодами за допомогою пошуку шляху. Основним моментом роботи є керування мобільного робота з однієї точки в іншу. Нижче зображений алгоритм з використанням псевдо коду на Рис.- 5 і Рис.2.6 ілюструє крок за кроком процесу алгоритму псевдо коду.

Псевдокод є системою позначень і правил, призначеною для одноманітного запису алгоритмів.

Псевдокод займає проміжне місце між природною і формальною мовами. З одного боку, він близький до звичайної природної мови, тому алгоритми можуть на нім

записуватися і читатися як звичайний текст. З іншого боку, в псевдокодi використовуються деякі формальні конструкції і математична символіка, що наближає запис алгоритму до загальноприйнятого математичного запису[21].

У псевдокодi не прийняті строгі синтаксичні правила для запису команд, властиві формальним мовам, що полегшує запис алгоритму на стадії його проектування і дає можливість використовувати ширший набір команд, розрахований на абстрактного виконавця.

Проте в псевдокодi зазвичай є деякі конструкції, властиві формальним мовам, що полегшує перехід від запису на псевдокодi до запису алгоритму на формальній мові. Зокрема, в псевдокодi, так само, як і у формальних мовах, є службові слова, сенс яких визначений раз і назавжди. Вони виділяються в друкарському тексті жирним шрифтом, а в рукописному тексті підкреслюються[39].

Єдиного або формального визначення псевдокоду не існує, тому можливі різні псевдокоди, що відрізняються набором службових слів і основних (базових) конструкцій.

Для запису алгоритмів за допомогою мови псевдокодів використовуються службові слова та спеціальні правила запису окремих дій.

У мові псевдокодів прийняті певні синтаксичні правила для запису команд, що полегшує запис алгоритму на стадії його проектування і дає можливість використання більш широкого набору команд, розрахованого на абстрактного виконавця[42].

Наприклад, у мові псевдокодів використовуються спеціальні службові слова та правила запису для вибору напрямку подальшого виконання алгоритму в залежності від виконання чи невиконання сформульованої умови **ЯКЩО-ТО-ІНАКШЕ**, повторення визначеної групи дій певну кількість разів **ПОКИ-ПОЧАТОК-КІНЕЦЬ** тощо. Разом з тим мова псевдокодів дозволяє використовувати дещо довільну форму представлення математичних записів, умов тощо. Розглянемо як приклад алгоритм Евкліда:

У наведеному алгоритмі зовсім неважко розібратися. Мова псевдокодіє максимально наближена до звичайної розмовної мови. Усі службові слова, що використовуються для запису алгоритму, виділені жирним шрифтом, їх необхідно записувати тільки таким чином, як це передбачено правилами мови. Окремі блоки алгоритму виділені вертикальними лініями і означають певні дії. Внутрішній блок визначає вибір тієї чи іншої дії в залежності від відношення між значеннями величин p та t [29]. Середній блок визначає дію повторення внутрішніх дій до того часу, поки значення величин p та t не стануть рівними. Зовнішній блок визначає повністю весь алгоритм, в якому передбачені також дії введення та виведення відповідно вхідної та результуючої інформації.

```
OPEN //the set of nodes to be evaluated
CLOSED //the set of nodes already evaluated
add the start node to OPEN

loop
  current = node in OPEN with the lowest f_cost
  remove current from OPEN
  add current to CLOSED

  if current is the target node //path has been found
    return

  foreach neighbour of the current node
    if neighbour is not traversable or neighbour is in CLOSED
      skip to the next neighbour

  if new path to neighbour is shorter OR neighbour is not in OPEN
    set f_cost of neighbour
    set parent of neighbour to current
    if neighbour is not in OPEN
      add neighbour to OPEN
```

Рис.2.5 - Псевдокод для алгоритму



Рис.2.6 - Відкриті кола в вузлах належать до відкритого списку, червоний / зелений закриті список.

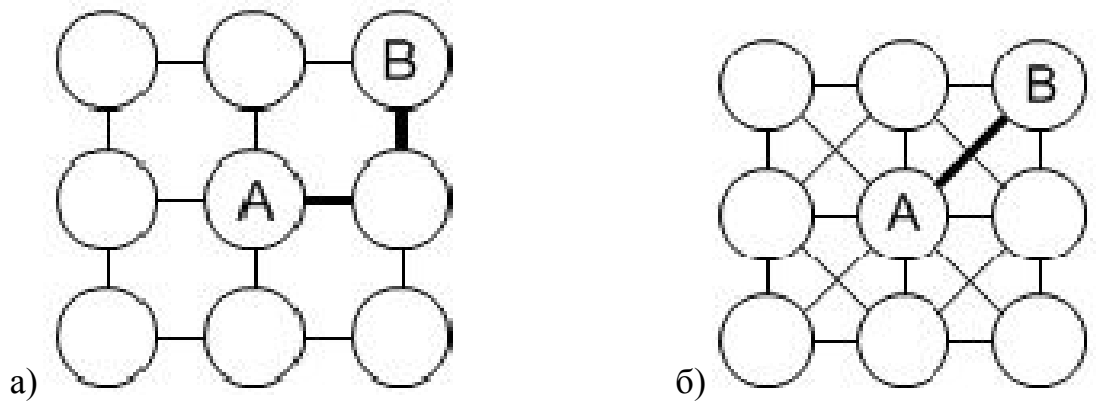


Рис.2.7 - Шлях, отриманий з використанням 4-підключення або 8-зв'язність в графі, щоб перейти від вузла А до вузла В більш плавний шлях виходить, якщо 8-зв'язність використовується (б) замість використання 4-підключення

2.3 Дизайн конструкції робота без зіткнень

2.3.1 Коридор

Перший сценарій коридору був обраний, тому що він є найпоширенішим, головним чином тому, що він доступний в будь-якому науково-дослідному відділі. Тут головна мета робота, щоб подалати певну відстань по коридору, вказану користувачем, уникаючи перешкоди, щоб досягнути мети.

Конструкція синьої лінії вказує на те місце, з якого повинен починати рух робот, маючи на увазі, що він може почати рухатися з будь-якої точки на лінії. Червона лінія є кінцевим пунктом призначення робота. На малюнку 2.8 представлені приклади різних можливих конфігурацій сценарію. Коридор заповнюється різними перешкодами. Робот повинен почати рух паралельно стіни і стінок, розміщених на правій частині робота, що повинні бути видимі для робота (тобто, немає ніяких відкритих дверей) на самому початку. Таким чином, робот здатний локалізувати себе по відношенню до загальних характеристик. Намічена лінія повинна бути уздовж порожньої області. При визначенні початку і пункту призначення лінією, а не фіксованою точкою, доведено, що робот не обмежується, щоб почати і закінчити в тому ж місці на рівній формі перешкод.

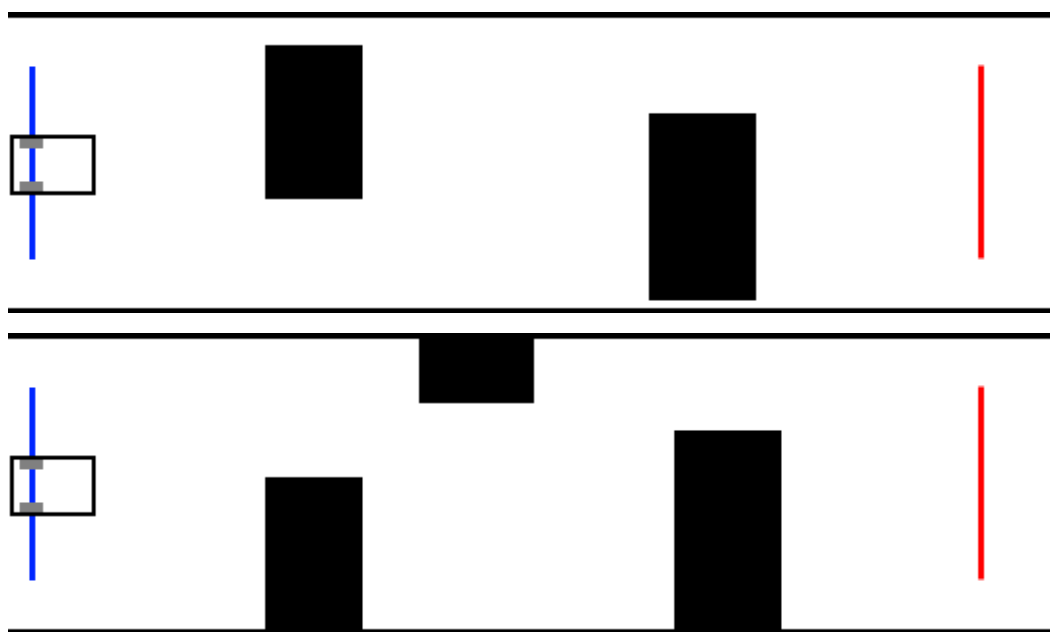


Рис.2.8- Дві різні конфігурації коридору. Перешкоди представлені у вигляді чорних блоків

Мета для робота - пройти по коридору, рішення задачі просто і легко зрозуміти, подивившись на Рис.-8. Робот починає рухатись, дивлячись на його праву сторону, вимірюючи відстань до стіни і, таким чином, робот може бути

розташований на мапі. Потім, попереду робота вимірюється відстань до наміченої точки, яка задається користувачем.

Коли мету вже встановлено, основний контур управління виконується до тих пір, поки намічена точка не буде досягнута. Потім лазерний сканер використовується для вимірювання відстані до видимих перешкод, ці отримані дані від цього процесу використовується для побудови карти.

Щойно збудована карта може бути розширена для використання алгоритму планування шляху, якщо жоден шлях не запланований, або якщо є будь-які перешкоди вздовж вже спланованого шляху.

2.3.2 Вихід з кімнати

Другий сценарій являє собою приміщення з виходом. Цей сценарій вводить новий виклик автономії робота, де робот починає всередині приміщення і повинен знайти дверний проріз і пройти через нього автономно.

На Рис.2.9- показано структуру, використовуваної для сценарію "вийти з кімнати", де робот починає рух в будь-якій точці всередині приміщення, навіть не будучи паралельно одній зі стін, що робить задачу більш складнішою. Робот повинен закінчити приблизно в червоній лінії таким чином, що кімната залишилося позаду нього.

Наявність перешкод обмежена в процесі пошуку дверей. Якщо є перешкода між роботом і виходом, робота може вона збити з пантелику і може сприйматися в якості одного з країв дверей.

Розміри приміщення обмежені також, безпосередньо в залежності від максимальної відстані вимірювання лазерного сканера. Ці обмеження обумовлені стратегією, використовуваної, щоб знайти вихід, як це видно на Рис.2.10-.

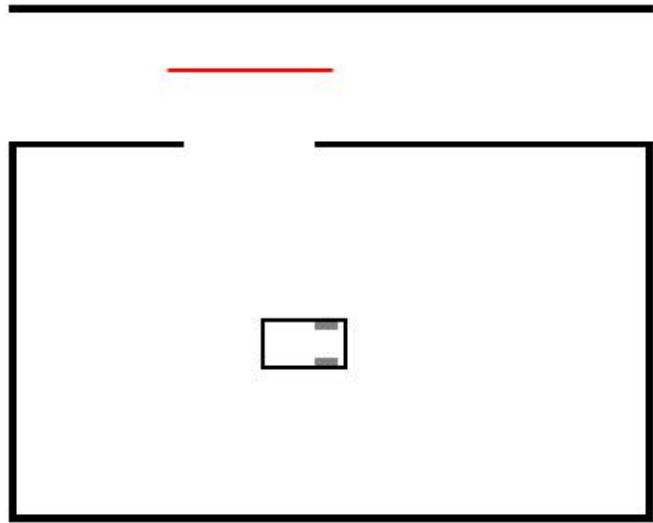


Рис.2.9- Можлива конфігурація сценарію "вийти з кімнати". Робот починає в будь-якій точці всередині приміщення і повинен закінчити біля червоної лінії.

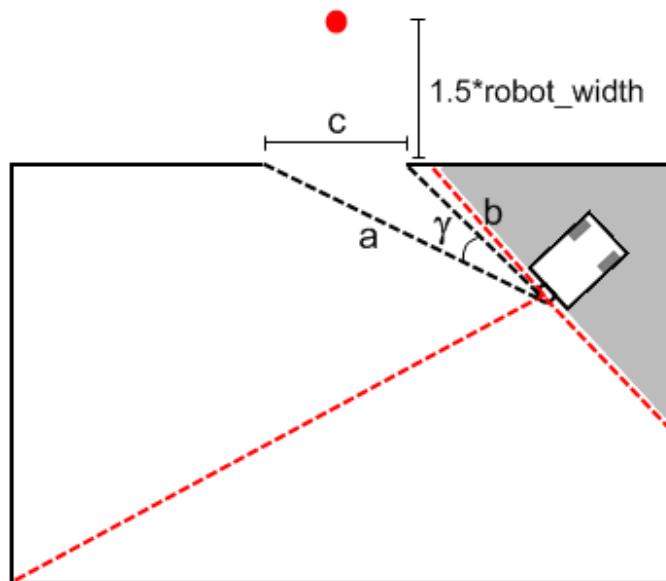


Рис.2.10 - знайти вихід

Закон косинусів використовується для розрахунку ширини дверей, яка повинна бути досить великою, щоб розглядатися. Якщо двері досить великі намічена точка (червона крапка) встановлюється в 1,5 рази більше ширини робота від дверей. Видима область для робота позначається пунктирними червоними лініями. Якщо

лазерний сканер не бачить всю область перед роботом, намічена точка може бути встановлений в неправильному місці.

$$c^2 = a^2 + b^2 - 2ab \cos(\gamma) \quad (2,2)$$

Для того, щоб знайти вихід, використовується проста стратегія. Ідея полягає в тому, щоб знайти достатньо великий зазор між стінами, використовуючи закон косинусів (див рівняння 2.1), а потім встановити точку, в 1,5 рази більше ширини виходу робота з кімнати.

Головною новинкою щодо реалізації сценарію коридору є фаза, яка здійснює процес пошуку двері, так що тільки ця фаза описана тут. Після того, як карта вже побудована, робот шукає дверний отвір, використовуючи дані, що надходять від лазерного сканера. Якщо двері досить широкі, вони будуть знайдені, кінцева точка встановлюється поза приміщенням i , таким чином, на етапі планування шляху робот готовий до виконання руху. Якщо це не так, то набір тимчасових рухів зроблені таким чином, що робот стикається з іншою перешкодою навколишньому середовища.

2.3.3 Лабіринт

Останній сценарій – це обраний лабіринт. Лабіринти є, по суті, складною умовою для всіх, так що це гарна ідея, щоб реалізувати загальну стратегію, щоб вирішити її за допомогою робота. Основна мета для робота в цьому сценарії – дістатися до виходу з лабіринту, уникаючи існуючі перешкод

На Рис.2.11 - показаний приклад лабіринту, щоб завдання було вирішене за допомогою робота. Робот повинен почати біля синьої лінії і закінчити на червоній лінії, не повертаючись на своєму шляху. Лабіринтом може бути бути одноходовою лабіринт, такий, що робот повинен рухатися по шляху. Не існує обмежень щодо розмірів лабіринту, беручи до уваги, що робот повинен мати можливість

проходити крізь стіни.

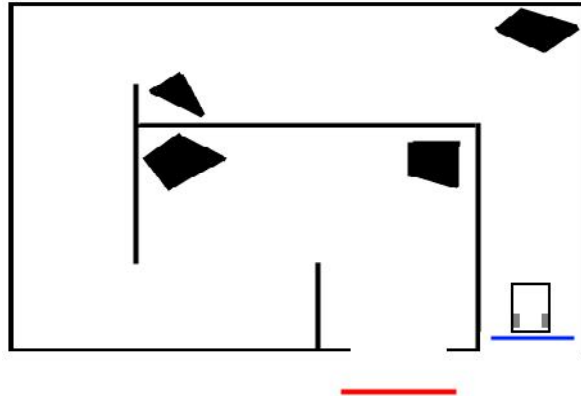


Рис.2.11 - Можлива конфігурація сценарію "лабіринт"

Робот починає від синьої лінії, поверненої всередину лабіринту. Червона лінія є кінцевим пунктом призначення робота.

Лабіринт пройдений шляхом встановлення тимчасових точок всіх цілей, як це видно на малюнку 2.12. Коли намічена точка поставлена, то вона є далекою точкою, видно з лазерного сканера в той момент, відкидаючи моменти, які могли б зробити ситуацію, щоб робот врізався в стіну. Вихід вважається досягнутим, коли більшість ліній сканування показують значення більше ніж попередньо визначену відстань.

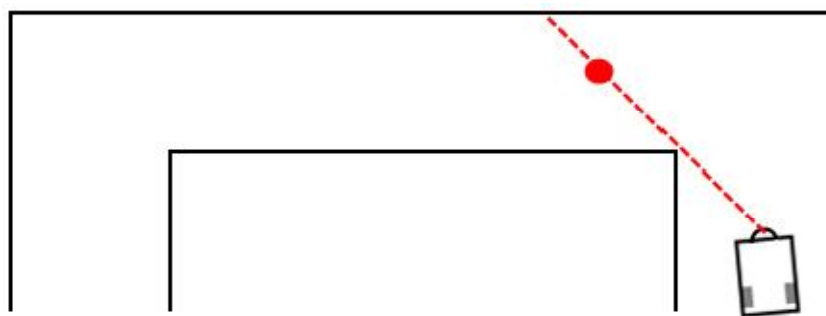


Рис.2.12 - Установка попередньої наміченої точки, щоб вийти з лабіринту

Намічена точка представлена у вигляді червоної крапки і найбільш віддаленої досяжні точки вздовж лінії сканування, де найбільша відстань. Для цього, розширена карта використовується, щоб уникнути удару стіни

Реалізований вихід в лабіринті показаний на рис.2.11. Після локалізації та сканування фаз, перевіряється, якщо вихід був досягнутий. В цьому випадку робот завершує завдання. Якщо вихід не був досягнутий, після будівництва карти, попередня тимчасова точка була досягнута або тимчасова намічена точка натикається на перешкоду. Після того, як нова попередня мета поставлена, потрібно планувати новий шлях. Якщо жодна попередня мета не встановлена протягом поточного циклу і існує перешкода, що блокує поточний запланований шлях, це планується знову. І, нарешті, крок планованого шляху виконується на етапі виконання шляху продовження з наступного циклу.

Під час цієї глави, три різні сценарії були запропоновані і різноманітні стратегії були розроблені і впроваджені для вирішення простих завдань в цих сценаріях. Стратегії намагаються бути максимально автономними, наскільки це можливо, таким чином, що робот здатний рухатися протягом певного періоду часу, не отримуючи прямих замовлень від людини. Більш того, певні обмеження були чітко вказано для кожного сценарію. Тепер все готово для проведення випробувань, щоб продемонструвати доцільність запропонованих стратегій.

III. РОЗРОБКА ПРОГРАМИ

3.1 Програма для робота

Програма для робота називається SMR (Small Mobile Robot) доступний і розроблений в DTU Electrical Engineering і використовується для перевірки рішень, запропонованих в ході цього проекту. Ця програма вже використовується в багатьох науково-дослідних проектах і курсах в області робототехніки.

SMR – загальний диференціальний робот, який працює на Linux, за умови, якщо він укомплектований різними датчиками: датчика лінії, датчики близькості ІК, кодери колеса, камери і лазерного сканера.

На малюнку 3.1 показаний SMR, забезпечений HOKUYO URG-04LX лазерного сканера. Цей лазерний сканер вимірює площу до 4 м, і 240 ° з дозволом приблизно 0.36°.

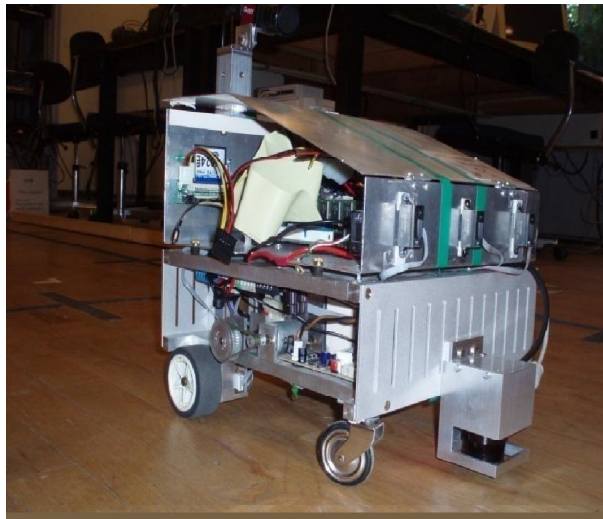


Рис.3.1 - SMR робот

Два інтерфейси програмного забезпечення використовуються для проведення випробувань: smrdemo і лазерного серверу. smrdemo реалізує SMR-CL

(SMR-Control Language) [SMR-CL], який забезпечує набір команд високого рівня для переміщення робота, а також команди для читання одометра. Лазерний сервер надає команди, щоб отримати свідчення від пристрою лазерного сканера і встановлює ширину сканування.

Будова, використовувана для робота SMR з MATLAB зображений на малюнку 3.2. Два сполуки гнізда, встановленого між MATLAB і відповідними серверами. smrdemo отримує необхідні команди для переміщення робота і повертає поточні дані одометра, коли це потрібно. Лазерний сервер пропонує нове читання сканера при необхідності і дозволу, в числі інших варіантів, щоб змінити ширину сканера.

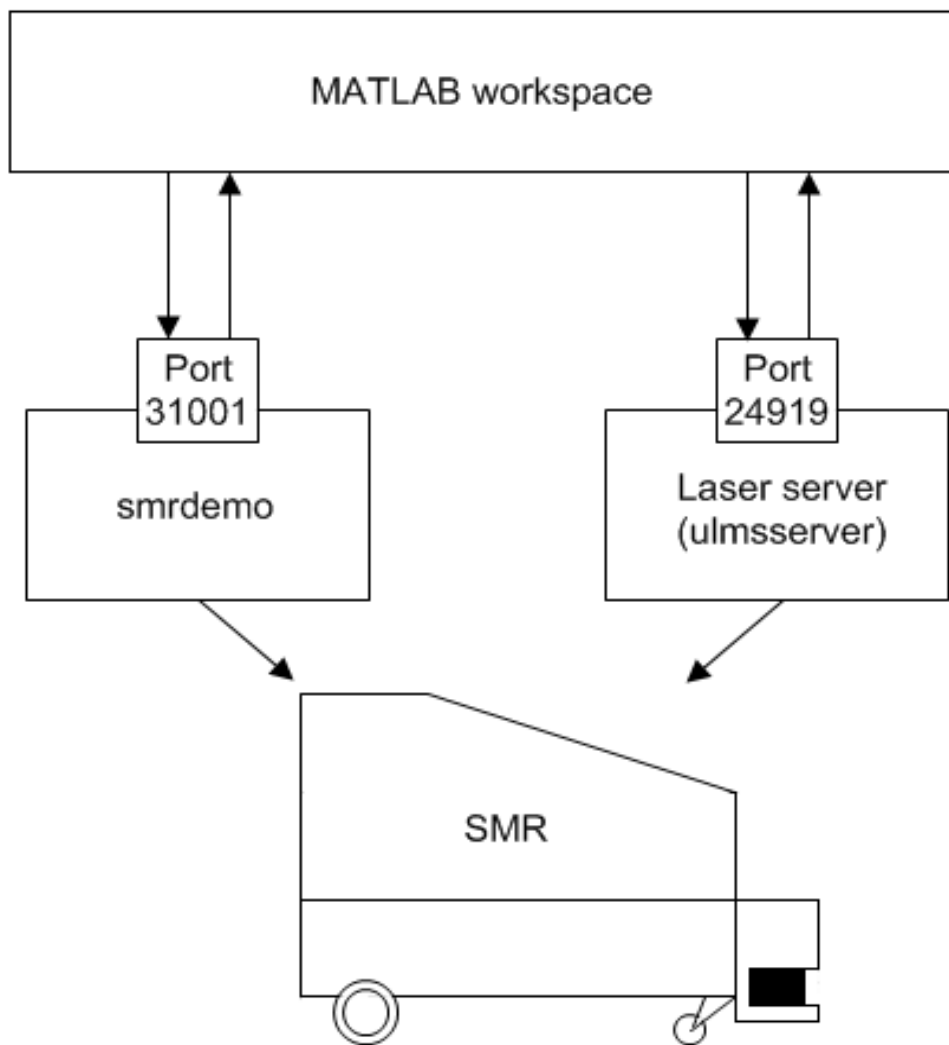


Рис.3.2 - Структура, використона для роботи з SMR

3.2 Matlab

MATLAB — пакет прикладних програм для числового аналізу, а також мова програмування, що використовується в даному пакеті. Система створена компанією The MathWorks і є зручним засобом для роботи з математичними матрицями, малювання функцій, роботи з алгоритмами, створення робочих оболонок (user interfaces) з програмами в інших мовах програмування. Хоча цей продукт спеціалізується на чисельному обчисленні, спеціальні інструментальні засоби працюють з програмним забезпеченням Maple, що робить його повноцінною системою для роботи з алгеброю.

MATLAB має більше, ніж мільйон користувачів на виробництвах і науковців. Ціна базової комерційної версії без інструментів близько 2000 дол. США і лише 100 дол. США для навчальних закладів з мінімальним набором інструментів.

MATLAB надає користувачеві велику кількість функцій для аналізу даних, які покривають майже всі області математики, зокрема:

- Матриці та лінійна алгебра — алгебра матриць, лінійні рівняння, власні значення і вектори, сингулярності, факторизація матриць та інше.
- Многочлени та інтерполяція — корені многочленів, операції над многочленами та їх диференціювання, інтерполяція та екстраполяція кривих...
- Математична статистика та аналіз даних — статистичні функції, статистична регресія, цифрова фільтрація, швидке перетворення Фур'є та інші.
- Обробка даних — набір спеціальних функцій, включаючи побудову графіків, оптимізацію, пошук нулів, чисельне інтегрування та інше.
- Диференційні рівняння — вирішення диференційних і диференційно-алгебраїчних рівнянь, диференційних рівнянь із запізнюванням, рівнянь з обмеженнями, рівнянь в часткових похідних та інше.

- Розріджені матриці — спеціальний клас даних пакету MATLAB, що використовується у спеціалізованих додатках.
- Цілочисельна арифметика — виконання операцій цілочисельної арифметики в середовищі MATLAB.

MATLAB — одна з найстарших, ретельно пророблених й перевірених часом систем автоматизації математичних розрахунків, яка побудована на розширеному представленні й застосуванні матричних операцій. Це знайшло відображення в назві системи — MATrix LABoratory — матрична лабораторія. Однак синтаксис мови програмування системи продуманий настільки ретельно, що ця орієнтація майже не відчувається тими користувачами, яких не цікавлять безпосередньо матричні обчислення.

Матриці широко застосовуються в складних математичних розрахунках, наприклад, при рішенні задач лінійної алгебри і математичного моделювання статичних і динамічних систем і об'єктів. Вони є основою автоматичного складання й рішення рівнянь стану динамічних об'єктів і систем. Прикладом може служити розширення MATLAB — Simulink. Це істотно підвищує інтерес до системи MATLAB, що увібрала в себе кращі досягнення в області швидкого рішення матричних задач.

Однак сьогодні MATLAB далеко вийшла за межі спеціалізованої матричної системи і стала однією з найбільш могутніх універсальних інтегрованих обчислювальних систем. Слово «інтегрована» указує на те, що в цій системі об'єднані зручна оболонка, редактор виражень і текстових коментарів, обчислювач і графічний програмний процесор. У новій версії використовуються такі могутні типи даних, як багатомірні масиви, масиви осередків, масиви структур, масиви Java і розріджені матриці, що відкриває можливості застосування системи при створенні й налагодженні нових алгоритмів матричних і заснованих на них рівнобіжних обчислень і великих баз даних.

У цілому MATLAB — це унікальна колекція реалізацій сучасних чисельних методів комп'ютерної математики, створених за останні три десятки років. Вона увібрала в себе і досвід, правила і методи математичних обчислень, які були накопичені за тисячі років розвитку математики. Це сполучається з могутніми засобами графічної візуалізації і навіть анімаційної графіки. Систему з прикладеної до неї великою документацією цілком можна розглядати як фундаментальний багатотомний електронний довідник по математичному забезпеченню ЕОМ — від масових персональних комп'ютерів до супер-ЕОМ. На жаль, поки представлений цілком лише англійською і частково японською мовами.

Система MATLAB була розроблена Молером (С. В. Moler) і з кінця 70-х рр. широко використовувалася на великих ЕОМ. На початку 80-х рр. Джон Литл (John Little) з фірми MathWorks, Inc. розробив версії системи PC MATLAB для комп'ютерів класу IBM PC, VAX і Macintosh. Надалі були створені версії для робочих станцій Sun, комп'ютерів з операційною системою UNIX і багатьох інших типів великих і малих ЕОМ. Зараз понад десяток популярних комп'ютерних платформ можуть працювати із системою MATLAB. До розширення системи були притягнуті найбільші наукові школи світу в області математики, програмування й природознавства. І от тепер з'явилася новітня версія цієї системи — MATLAB 6. Однією з основних задач системи було надання користувачам могутньої мови програмування, орієнтованого на математичні розрахунки і здатного перевершити можливості традиційних мов програмування, що багато років використовувалися для реалізації чисельних методів. При цьому особлива увага приділялася як підвищенню швидкості обчислень, так і адаптації системи до рішення найрізноманітніших задач користувачів.

Можливості MATLAB дуже великі, а по швидкості виконання задач система нерідко перевершує своїх конкурентів. Вона застосовна для розрахунків практично в будь-якій області науки й техніки. Наприклад, дуже широко

використовується при математичному моделюванні механічних пристроїв і систем, зокрема, в динаміці, гідродинаміці, аеродинаміці, акустиці, енергетиці і т.д. Цьому сприяє не тільки розширений набір матричних і інших операцій і функцій, але і наявність пакета розширення (toolbox) Simulink, спеціально призначеного для рішення задач блокового моделювання динамічних систем і пристроїв, а також десятків інших пакетів розширень.

У великому й постійно поповнюваному комплексі команд, функцій і прикладних програм системи MATLAB містяться спеціальні засоби для електротехнічних і радіотехнічних розрахунків (операції з комплексними числами, матрицями, векторами й поліномами, обробка даних, аналіз сигналів і цифрова фільтрація), обробки зображень, реалізації нейронних мереж, а також засоби, що відносяться до інших нових напрямків науки й техніки. Вони ілюструються безліччю практично корисних прикладів.

До розробок розширень для системи MATLAB притягнуто багато наукових шкіл світу, якими керують великі вчені й педагоги університетів. Важливими достоїнствами системи є її відкритість і розширюваність. Більшість команд і функцій системи реалізовані у виді текстових m-файлів (з розширенням .m) і файлів мовою Си, крім того усі файли доступні для модифікації. Користувачеві надана можливість створювати не тільки окремі файли, але і бібліотеки файлів для реалізації специфічних задач.

Разюча легкість модифікації системи і можливість її адаптації до рішення специфічних задач науки і техніки привели до створення десятків пакетів прикладних програм (toolbox), які набагато розширили сфери застосування системи.

Однією з проблем сучасної науки є розробка і впровадження в практику методів дослідження функціонування складних систем. До класу складних систем відносять технологічні, виробничі, енергетичні комплекси, системи автоматизації керування й інших об'єктів. Моделювання є одним із найбільш могутніх засобів

дослідження подібних систем на сьогоднішній день. Моделювання - один із найбільш розповсюджених засобів вивчення різних процесів і явищ. Моделлю вихідного об'єкта називається представлення об'єкта в деякій формі, відмінної від форми його реального існування [1]. В інженерній практиці модель, звичайно, створюється для:

1. проведення на моделі експериментів, які неможливо або складно провести на реальному об'єкті (що надає можливість одержання нових знань про об'єкт);
2. прискорення, здешевлення, спрощення і будь-якого іншого удосконалення процесу проектування, що досягається за рахунок роботи з більш простим об'єктом, чим вихідний, тобто з моделлю.

Сьогодні відомі і широко використовуються в наукових дослідженнях і інженерній практиці різні типи моделей і численні методи моделювання. Якщо взяти за основу ступінь абстрактності (ступінь відмінності від реального об'єкта), то можна визначити наступні типи моделей:

1. фізичні (натурні) моделі (відтворюють досліджуваний процес із збереженням його фізичної природи і є інструментом фізичного моделювання);
2. аналогові моделі (заміняють один об'єкт на іншій зі схожими властивостями);
3. математичні моделі (абстрактні моделі, існують у формі спеціальних математичних конструкцій).

Під математичним моделюванням розуміють засіб дослідження різних процесів шляхом вивчення явищ, що мають різний фізичний зміст, але описуваних однаковими математичними співвідношеннями [2].

Серед великого числа пакетів візуального моделювання пакет Matlab займає особливе місце. Спочатку орієнтований на дослідницькі проекти, пакет в останні роки став робочим інструментом інженерів, студентів, керівників, фізиків,

зв'язківців. Однією з основних причин широкого використання пакета Matlab є великий спектр засобів, що надає користувачеві для рішення різноманітних задач у різних областях людської діяльності. Серед цих засобів особливе місце займає підсистема Simulink.

Simulink - це інтерактивне середовище для моделювання й аналізу широкого класу динамічних систем за допомогою блок-діаграм.

Simulink забезпечують інтерактивне середовище для моделювання, при цьому поведження моделі і результати її функціонування відображаються в процесі роботи, і існує можливість змінювати параметри моделі навіть у той момент, коли вона виконується. Simulink дозволяє створювати власні блоки і бібліотеки блоків з доступом із програм на Matlab, Fortran чи C, зв'язувати блоки з розробленими раніше програмами на Fortran і C, що містять вже перевірені моделі.

Починаючи з версії 3.0 у Simulink з'явилися спеціалізовані додатки, що значно збільшили міць даного середовища моделювання:

- Stateflow - графічний інструментарій для проектування складних систем керування. Stateflow дає можливість моделювати поведження складних подійно-керувальних систем, базуючись на теорії кінцевих автоматів. Це дозволяє користувачам Simulink додавати подійно-керувальне поведження до їхніх моделей.

- Stateflow Coder - генерація C коду для діаграм Stateflow . Використовуючи Stateflow і Stateflow Coder, користувач може генерувати код винятково для Stateflow-частин моделі Simulink.

- Real-Time Workshop - доповнює Simulink і Stateflow Coder, забезпечуючи автоматичну генерацію коду C з моделей Simulink. За допомогою Real -Time Workshop можна легко генерувати код для дискретних, безупинних і гібридних систем, включаючи системи, що містять підсистеми працюючі при виконанні визначених умов.

- DSP Blockset - бібліотеки блоків Simulink для створення, моделювання і макетування цифрових систем обробки сигналів.
- Nonlinear Control Design Blockset - інтерактивний підхід до автоматизованого проектування систем керування.
- Fixed-Point Blockset - бібліотеки блоків Simulink для моделювання поведінки систем керування і динамічних фільтрів з фіксованою крапкою.
- Simulink Report Generator - дозволяє створювати і будувати звіти з моделей Simulink і Stateflow у різних форматах, серед яких HTML, RTF, XML і SGML.

3.3. Випробування програм

У цій частині розділу тестуються змодельовані рішення і стратегії, запропоновані в попередньому розділі, а також в реальних умовах. Кілька форм перешкод тестуються для кожного сценарію; проте, в цьому розділі наведені результати найбільш значних змін.

Моделювання роботи в робочому просторі MATLAB, а результати нанесені згодом належним чином. З іншого боку, випробування в реальних умовах здійснюється з використанням платформи робота SMR і будови показано в Додатку Е. Цей підрозділ описується в трьох основні секціях, як кожен сценарій був перевірений в обох: змодельованих і реальних умовах, і обговорення відповідних результатів. І, нарешті, зроблений висновок отриманих результатів.

3.3.1 Коридор

Коридор заповнюється декількома перешкодами і вони розташовані паралельно до стіни (як це вимагається в розділі 3.2.1). Середовище для моделювання створюється належним чином.

Моделюємо умови для робота, що проїзжає відстань 3 м вздовж коридору, використовуючи розмір осередку 2 см. Результат може бути показано на малюнках 3.3 - 3.5.

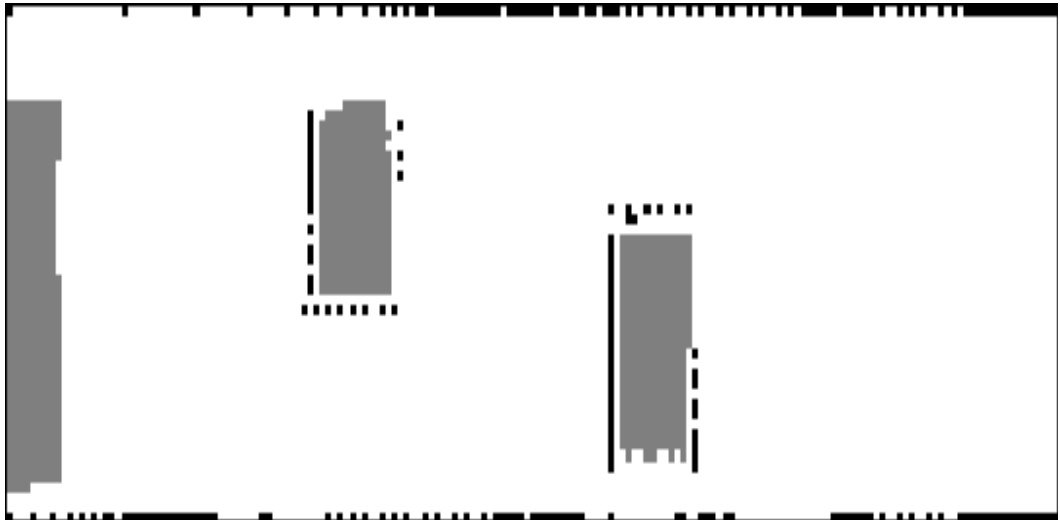


Рис.3.3 - Результат моделювання для сценарія "Коридор".



Рис.3.4 - це та ж карта після процесу розтягування

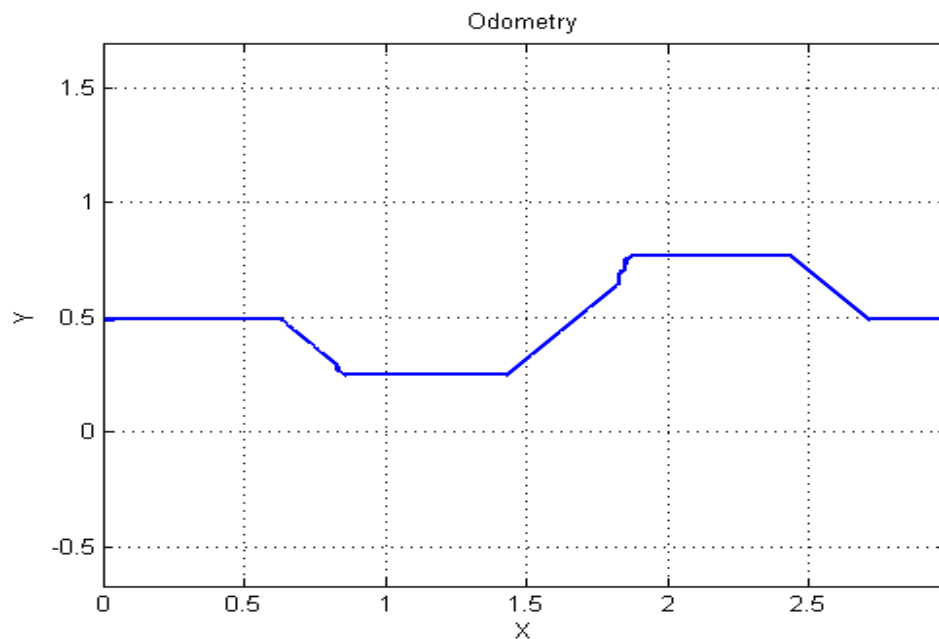


Рис.3.5 - показує одометр.

Як і слід було очікувати, після проведення змодельованого випробування і реального для того ж коридору, отримані результати з точки зору карти та одометра схожі. Робот правильно виконав своє завдання, йдучи по коридору на відстань, задану користувачем.

3.3.2 Вихід з кімнати

Щоб перевірити запропоновану стратегію, знайти вихід з кімнати, вона була побудована в реальному середовищі, обмеження яких коригується до умов, зазначених в розділі 3.2.1, і можна побачити на малюнку 6. Аналогічний експеримент створений для моделювання з чотирма стінами і перешкодою в одному з кутів.

Моделювання

У цьому тесті було показано, що робот досягне виходу і вийде з кімнати позаду нього, як це видно на малюнку 6

Робот починає рух всередині кімнати і, після того, як будується умовний шлях, він знаходить двері і проходить через них.

Розв'язання поставленої задачі запропоноване, знайти двері в кімнаті, а потім пройти через них, це і було продемонстровано. Було також підтверджено, що робот здатний досягнути виходу, навіть якщо є будь-які перешкоди всередині приміщення, але завжди маючи на увазі обмеження з приводу перешкод, необхідно зазначити. Зверніть увагу на те, що карта, отримана в ході реального тесту не орієнтована, як на це можна було б сподіватися. Це тому, що одна з цілей полягає в тому, що робот поміщається випадковим чином всередину приміщення і жоден з методів, щоб локалізувати його по відношенню до реального стану не використовується.

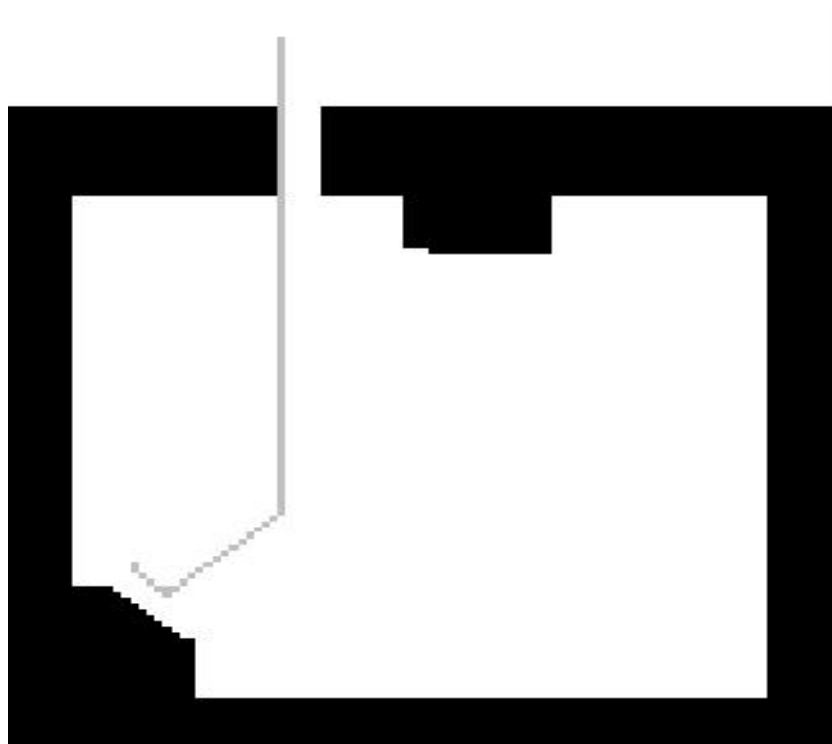


Рис.3.6 - Результат моделювання здійснюється на етапі «знайти двері»

3.3.3 Лабіринт

Моделювання задачі, робот вирішує односторонній лабіринт, запропонований лазерним сканером, йдучи по єдиному шляху і зупинках, для знаходження виходу з лабіринту. Результат показаний на малюнку 3.7. – 3.9.

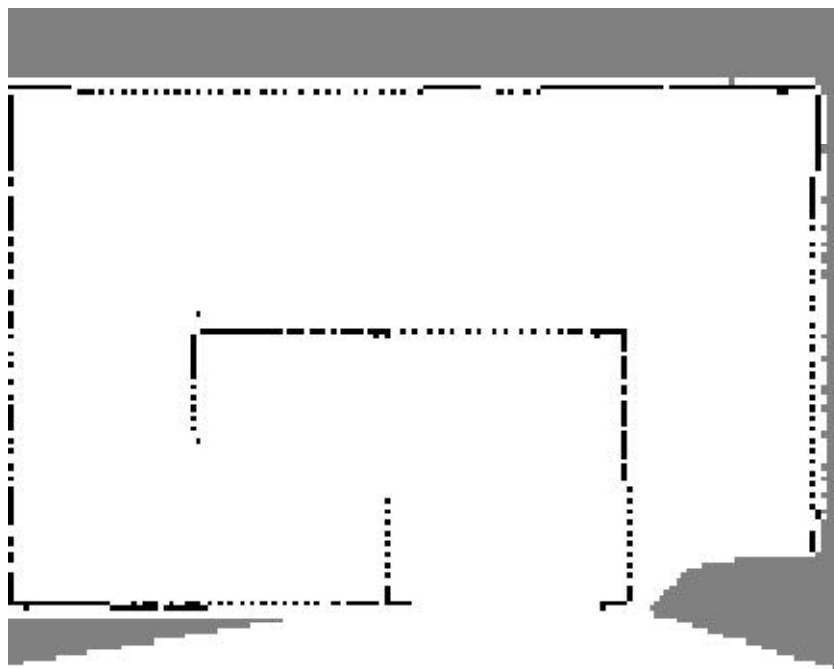


Рис. 3.7 - Результат після запуску робота, щоб вирішити лабіринт

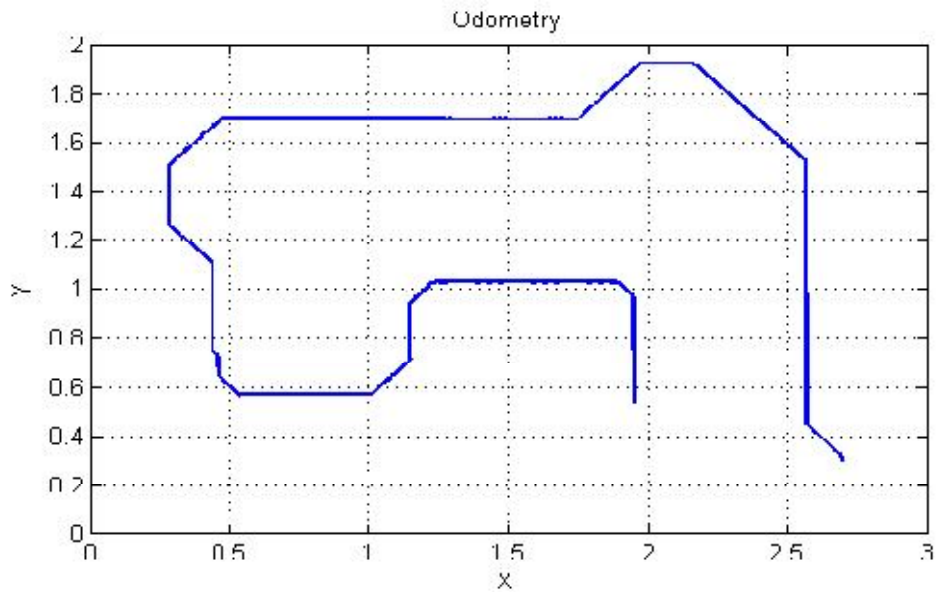


Рис.3.9 - є результатом процесу уповільнення ходу, застосовується до невіправленої мапи

Різноманітність конфігурацій, які успішно пройшли випробування для даного сценарію показав, що рішення, призначені для виконання завдань, сприймають різний стан навколишнього середовища. Це доводить, що робот безпосередньо не слідує прямим наказам заданих користувачем, що означає, що робот є автономним, що однією з головних цілей проекту. Були зроблені великі зусилля, щоб поліпшити виконання запланованого шляху, особливо при роботі з реальним роботом, таким чином, що час, витрачений, щоб закінчити завдання був якомога менше, але не втрачаючи при цьому здатність робота рухатися в динамічному середовищах. З результатів випробувань, проведених на SMR, можна зробити висновок про те, що клітинки, розміром менше, ніж 3 см, не підходять, через помилки, накопичені в одометрі. Просте рішення, уникати роботом запропонованих заповнених клітинок і пройти випробування успішно, полягає в очищенні поточної клітинки, якщо декілька сусідів заповненої клітинки, раптом стануть окупованими.

ВИСНОВКИ

Автономний мобільний робот працює в частково невідомих середовищах з використанням різних стратегій для вирішення простих завдань. Також був добре змодельований реальний лазерний сканер, що використовувався для сприйняття навколишнього середовища, який демонстрував би доцільність використання цього точного датчика для мобільних роботів. Відображення основи сітки використовується для зіставлення реальності настільки, наскільки це можливо, і служить в якості вхідних даних для пошуку алгоритму зірки, який вирахував необхідні маршрути для досягнення різних цілей. Отже, було вкладено велике зусилля в обробку запланованого шляху таким чином, що робот завершує свої завдання якомога швидше, але не втрачаючи при цьому можливості працювати в динамічному навколишньому середовищі. Було досягнуто глибоке розуміння всього процесу найпоширенішої схеми управління, яку використовують в мобільних роботах, і набору функцій, які імітують поведінку робота, і готові служити в якості основи для тестування нових сценаріїв до того, як робот працює в реальному середовищі. Крім того, були правильно застосовані необхідні дії для відображення точок в різних системах координат навколишнього середовища. З іншого боку, для вирішення реально поставлених завдань, були спрямовані великі зусилля на оптимізацію коду. Для цього, були взяті до уваги багато технічних документів про лінійну індексацію та векторизацію в середовищі MATLAB..

Основною метою моделювання є необхідність продемонструвати роботу стратегії, яка була обрана для вирішення конкретного завдання, але для того щоб її працювати для всіх можливих конфігурацій, необхідно багато часу. Більш цікаві результати можна побачити в реальному роботі замість цього. SMR-CL продемонстрував, що він може бути придатним інструментом для абстрагування від реального часу, шляхом проведення процесів, таких як обчислення залишеної відстані або виявлення поточного кута, які потребували багато часу під час

моделювання.

І, нарешті, маємо зазначити, що платформа робота SMR є закінченою платформою робота, з якої можна досягати хорошого розуміння в автономних мобільних роботах, навіть якщо знання користувача обмежене про це обладнання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Besl P., Jain R. Three-dimensional object recognition // ACM Computing Surveys, (CSUR). – ACM Press. –1985. – Vol. 17. – Issue 1. – P. 75-145.
2. Asymptotic Stability and Feedback Stabilization. In Differential Geometric Control Theory / Brockett R. W. / Edited by R. W. Brockett, R. S. Milman, and H. J. Sussman. – USA: Birkhдuser Boston, Inc., 2008. – 181-191.
3. Multi-sensor fusion: fundamentals and applications with software / Brooks R., Iyengar S. / Prentice-Hall PTR, 1998. – 416 p
4. Ching-Chang Wong, Hou-Yi Wang, Kuan-Hua Chen, Chia-Jun Yu and H. Aoyama. Motion controller design for two-wheeled robot based on a batch learning structure // SICE Annual Conference, Tokyo. – 2008. – P. 772-776
5. D. Chen, F. Bai and L. Wu. Kinematics control of wheeled robot based on angular rate sensors. //IEEE Conference on Robotics, Automation and Mechatronics, September 21-24, Chengdu, China. – 2008. – P. 598-602.
6. Dixon, W. E., D. M. Dawson, F. Zheang, and Z. Erkan. Global Exponential Tracking Control of a Mobile Robot System via a PE Condition // IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics. Vol. 31(1). – 2000. – P. 129-142.
7. E. Papadopoulos, I. Poulakakis and I. Papadimitriou, "On Path Planning and Obstacle Avoidance for Nonholonomic Platforms with Manipulators: A Polynomial Approach // The International Journal of Robotics Research, IJRR, vol. 21, 4. – April 2002. – P. 367-383.
8. Elfes A. Sonar-based real-world mapping and navigation // IEEE Trans. on Robotics and Automation. – 1987. – Vol.3(3). – P. 249-265.
9. Elfes A. Using occupancy grids for mobile robot perception and navigation // Intern. journal Computer. – 1989. – Vol. 22. (6). – P. 46-57.

10. Fierro, R., and F. L. Lewis. Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics // Proceedings of the IEEE Conference on Decision and Control. New Orleans, LA, USA. – 1995. – P. 3805-3810
11. Kanayama, Y., Y. Kimura, F. Miyazaki, and T. Noguchi. A Stable Tracking Control Method for an Autonomous Mobile Robot // Proceedings of the IEEE 102 International Conference on Robotics and Automation. Cincinnati, USA. – 1990. – P. 384-389.
12. Luca, A., Oriolo, G., Samson, C., and Laumond, J. P. Robot Motion Planning and Control, chapter Feedback Control of a Nonholonomic Car-like Robot. Lectures Notes in Control and Information Sciences. Springer. 1998 – P. 328
13. S. Hashim, Tien-Fu Lu. A new strategy in dynamic time-dependent motion planning for nonholonomic mobile robots // IEEE International Conference on Robotics and Biomimetics (ROBIO). – 2009. – P. 1692 - 1697
14. Vanmarcke E. Random fields: analysis and synthesis/ Vanmarcke E – MIT Press, Cambridge, Massachusetts. – 1983. – 372 p.
15. Walsh, G., D. Tilbury, S. Sastry, R. Murray, and J. P. Laumond. Stabilization of Trajectories for Systems with Nonholonomic Constraints // IEEE Transactions on Automatic Control. – Vol. 39(1). – 1994. – 216-222.
16. Y. Ichihara and K. Ohnishi. Path planning and tracking control of wheeled mobile robot considering robots capacity // IEEE International Conference on ICIT. – 2006. – P. 176 – 181
17. Yamamoto, Y.. Coordinating Locomotion and Manipulation of a Mobile Manipulator / Yamamoto, Y., X. Yun. – Singapore: World Scientific Publishing Co. Pte. Ltd, 1993. – P.157-181.
18. Батанов А. Ф., Грицынин С. Н., Муркин С. В. Робототехнические комплексы для обеспечения специальных операций // Специальная Техника. №6. – 1999. – С. 10–17.

- 19.Беляков К. Інформація організаційноправової сфери // Право України. – № 6. – 2004. – С. 88-92.
- 20.Бурдаков С.Ф. Системы управления движением колесных роботов / Бурдаков С.Ф., Мирошник И.В., Стельмаков Р.Э. – СПб.:Наука, 2001 .– 229с.
- 21.Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу. Затверджено МОН України, наказ №528 від 27.12.2001р.
- 22.Головко В. Нейроинтеллект: теория и применение. Книга 2 / Головко В. – БПИ, Брест. – 1999. – 227 с. 103
- 23.Голубев Ю.Ф. Построение движений робота-снейкбордиста / Голубев Ю. Ф. - М. : ИПМ, 2004. - 23 с.
- 24.ГОСТ 12.2.032-78 «ССТБ. Рабочее место при выполнении работ сидя. Общие эргономические требования».
- 25.Градецкий В.Г. Управляемое движение мобильных роботов по произвольно ориентированным в пространстве поверхностям / Градецкий В.Г., Вешников В.Б., Калиничко СВ. - М.:Наука. – 2001. – 360 с.
- 26.Гусев В.А. Математика: Справочные материалы: Кн. для учащихся.- 2е изд. / Гусев В.А., Мордкович А.Г. – М.: Просвещение. – 1990. – 416с.
- 27.ДБН В.2.5-28-2006. Інженерне обладнання будинків і споруд. Природне і штучне освітлення.
- 28.Юревич Е.И. Динамика управления роботами / Под ред. Е.И. Юревича. – М.: Наука. – 1984 . – 336 с.
- 29.ДСН 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку.
- 30.Ермолов И.Л., Мор Ф.Р., Подураев Ю.В, Шведов В.В. Мобильные роботы для инспекции и ремонта подземных трубопроводов: современное состояние и перспективы развития // Мехатроника. – №.1 – 2002.– С.30-33.

31. Климчук С. Проведення порівняльного аналізу законодавства у сфері інформації з обмеженим доступом України та країн-членів НАТО / С.Климчук. – №4ю – 2007. — Режим доступу:
<http://www.justinian.com.ua/article.php?id=2461>
32. Коваль В. Вдосконалений метод побудови локальної карти середовища мобільного робота // Вісник Тернопільського державного технічного університету ім. І.Пулюя. – 2003. – Т.8. – №2. – С. 80-88.
33. Коваль В. Спосіб виявлення перешкод мобільного робота з використанням технології злиття показів різнорідних сенсорів // АСУ и приборы автоматики. – Харьков. – 2004. – Вип. 126. – С. 128-135.
34. Коло. — Режим доступу: <http://uk.wikipedia.org/wiki/Коло>
35. Макаров И.М. Робототехника: история и перспективы / Макаров И.М., Топчиев Ю.И.-М.: Наука, Издательство МАИ. – 2003. – 101 с. 104
36. Медведев В. Управление роботами / Медведев В. А. Шиянов А.И. – Воронеж, 2003. – 187 с.
37. Мирошник И.В. Нелинейное и адаптивное управление сложным динамическими системами. / Мирошник И.В., Никифоров В.О., Фрадков А.Л. – СНБ.: Наука, 2000. – 325 с.
38. Охоцимский Д.Е., Павловский В.Е., Плахов А.Г., Туганов А.Н.. Моделирование игры роботов-футболистов и базовые алгоритмы управления ими // Искусственный интеллект. – №3. – 2000. – с. 534-540.
39. Павленко А. В. Новые технологии управления движением технических объектов // Сб. ст. по материалам 7-й Междунар. науч.-техн. Конф.- Новочеркасск .– №5. – 2004. – 184с.
40. Подураев Ю.В., Мехатроника : основы, методы, применение : учеб. пособие для студентов вузов, обучающихся по специальности "Мехатроника" направления подгот. "Мехатроника и робототехника".-М.: Машиностроение, 2006. –255с.

41. Попов Е., Письменный Г. Основы робототехники. Введение в специальность: Учебник для вузов. – М.: Высшая шк. – 1999. – 224 с.
42. Попов Е.П. Основы робототехники: Введение в специальность / Попов Е.П., Письменный Г.В. – М.: Высшая школа, 1990. — 224 с.
43. Уэно Х. Представление и использование знаний. Пер. с япон. / Нод. ред. Х. Уэно. – Исудзука. – М.: Мир, 1989.- 220 с.
44. Мачульский И.И. Робототехнические системы и комплексы./ Под.ред. И.И. Мачульского, М.: Транспорт, 1999. – 464с.
45. Спыну Г.А., Промышленные роботы. Конструирование и применение. Учеб. пособие. 2-е изд. / Спыну Г.А. – К.: Выща шк., 1991. – 311с.
46. Спыну Г.А., Роботы с искусственным интеллектом / Спыну Г.А. – К.: Техника, 1989. – 111с.
47. Андрианов Ю.Д. Управляющие системы промышленных роботов / Ю.Д. Андрианов, Л.Я. Глейзер, М.Б. Игнатъев и др.; Под общ. ред. И.М. Макарова. – М.: Машиностроение, 1984 – 288с. 105
48. Фу К. Робототехника / Фу К., Гонсалес Р., Ли К.. – М.: Мир, 1989. – 624с.
49. Шахинпур М. Курс робототехніки / Шахинпур М. Пер. с англ. – М.: Мир, 1990. – 527 с.
50. Юревич Е. И. Управление роботами и робототехническими комплексами / Юревич Е. И. — Л.: Машиностроение, Ленингр. отд-ние, 2000. — 235 с.
51. Юревич Е.И. Основы робототехники. 2-е изд., перераб. и доп / Юревич Е. И. - СПб.:БХВ - Петербург, 2005. – 416 с.
52. Ярушкина Н.Г. Основы теории нечетких и гибридных систем. Учебное пособие / Ярушкина Н.Г. – Финансы и статистика, 2004.- 320 с.