



БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ
ТЕХНОЛОГИИ

Денис Сафидинов Чикуртев

**ИЗСЛЕДВАНЕ И УПРАВЛЕНИЕ НА СЕРВИЗНИ РОБОТИ ЗА
ПОДПОМАГАНЕ НА ЧОВЕКА.**

А В Т О Р Е Ф Е Р А Т

НА ДИСЕРТАЦИЯ

за присъждане на образователната и научната степен „доктор“

по научна специалност “Приложение на принципите и методите на
кибернетиката в различни области на науката “

Професионално направление: 5.2. „Електротехника, Електроника и Автоматика“

Научен ръководител: проф. д-р Димитър Карастоянов

София, 2017 г.

Дисертацията е обсъдена и допусната до защита на разширено заседание на секция „Вградени интелигентни технологии“ на ИИКТ-БАН, състояло се на 2017г.

Дисертацията съдържа 106 стр, включващи 77 фигури, 2 таблици, приложения и литература, съдържаща 75 заглавия.

Защитата на дисертацията ще се състои на2017 г. от часа в зала 507 на блок 2 на ИИКТ-БАН на открито заседание на научно жури в състав:

1. Проф. д-р Тодор Нешков – ТУ-София
2. Проф. д-р Костадин Костадинов – ИМех.
3. Доц. д-р Любен Ключков – ТУ-София
4. Проф. д-р Димитър Карастоянов – ИИКТ
5. Доц. д-р Владимир Монов – ИИКТ

Резервни членове:

6. Проф. д-р Мара Кандева – ТУ-София
7. Доц. д-н. Любка Дуковска - ИИКТ

Материалите за защитата са на разположение на интересуващите се в стая 215 на ИИКТ-БАН, ул. „Акад. Г. Бончев“, бл. 25А.

Автор: *Денис Сафидинов Чикуртев*

Заглавие: ***ИЗСЛЕДВАНЕ И УПРАВЛЕНИЕ НА СЕРВИЗНИ РОБОТИ ЗА ПОДПОМАГАНЕ НА ЧОВЕКА.***

Цел и задачи

Целта на дисертацията е изследване, подобряване и създаване на методи, алгоритми и системи за управление на сервизни роботи, предназначени за подпомагане на човека. Провеждането на тези изследвания трябва да допринесе за развитието на сервизните роботи, като подобри и разшири техните възможности за изпълнение на различни по сложност задачи. Сервизните роботи трябва да могат да се движат автономно, да разпознават основни предмети, да взимат и донасят тези предмети, да взимат решения в определени ситуации, да могат да се управляват дистанционно и да бъдат на достъпна цена. Изследванията обхващат следните проблемни области при сервизните роботи: локализация и навигация, компютърно зрение, операционната система за роботи (ROS), интернет интерфейс, сензорни системи, интелигентни системи за управление, работи за обслужване на възрастни и инвалиди и бюджетни решения за намаляване и оптимизиране на цената на роботите.

Достигането на целта на дисертацията обхваща набор от задачи. Решението на тези задачи ще доведе до резултати, които да удовлетворят зададената цел. Формулираните задачи са следните:

- Анализ на настоящото състояние на научните изследвания и проблемите в областта на сервизната роботика;
- Изследване на методи за подобряване на локализация и навигация при мобилните роботи;
- Изследване на методи от компютърното зрение за разпознаване на предмети и хора
- Изследване на възможностите и приложението на Операционната система за роботи – ROS;
- Изследване на методи за управление на роботизирани манипулатори за работа в домашна среда;
- Разработване на методи за управление на сервизен робот, чрез различни устройства и интерфейси;
- Изследване на методи и техники за подпомагане на възрастни хора и инвалиди
- Провеждане на реални експерименти за да се покажат резултатите от проведените изследвания и разработки.

Публикации на докторанта по темата на дисертацията

На Международни конференции в чужбина:

Chivarov, N.; Shivarov, S.; Yovchev, K.; **Chikurtev, D.**; Shivarov, N. - Intelligent Modular Service Mobile Robot ROBCO 12 for Elderly And Disabled Persons Care; Robotics in Alpe-Adria-Danube Region (RAAD), 2014 23rd International Conference on, Smolenice, Slovakia, 3-5 Sept. 2014, Print ISBN: 978-1-4799-6797-1. p. 343-348

На Международни конференции у нас:

Denis Chikurtev - Application of C# Language for Programming Chain Mobile Robot with Infrared Thermos-Camera FLIR P640 for Inspection; International Conference Robotics, Automation and Mechatronics'14 RAM 2014, November 5-7 2014, Sofia, Bulgaria, p. 79-84, ISSN 1314-4634.

D. Chikurtev, M. Grueva, D. Karastoyanov – Intelligent Service Mobile Robots – Localization, Navigation And Observation Using IR Thermal Camera, ADP 2016, June, Sozopol, Bulgaria; p. 200-207, ISSN 1310-3946.

D. Karastoyanov, **D. Chikurtev**, M. Groueva, Robot Control And Modern ICT for Energy Efficiency And Better Quality of Innovative Technological Processes Using Smart Lab Equipment, International Scientific Congress “Machines. Technologies. Materials. 2016”, 14-17 September 2016, Varna, Bulgaria, volume 1, Section “Technologies” pp. 35-38 ISSN: 1310-3946.

D. Chikurtev, Service Mobile Robots – Localization And Recognition of Rooms And Buildings, Using Pre-Made Maps, International Conference Robotics, Automation and Mechatronics'16 RAM 2016, Byaga, Bulgaria, October 3-4, 2016, стр. 33-38, ISSN 1314-4634.

Denis Chikurtev – CORRECTION OF ODOMETRY ERRORS IN SERVICE MOBILE ROBOTS, International Conference on Tribology BULTRIB'16, 27- 29 October 2016, Sofia, Bulgaria, p. 352-358, ISSN: 1313-9878.

В Списание у нас:

D. Chikurtev - Optimizing the Navigation for Mobile Robot for Inspection by Using Robot Operating System, PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, Vol. 66, Sofia – 2015, ISSN 0204-9848, Online ISSN: 1314-409X p. 63-74

D. Chikurtev, K. Yovchev, E. Chikurtev, Design and functionality of Web User interface for control of service mobile robot through the Internet, PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, Vol 67, Sofia, 2016, p. 51-60, ISSN 0204-9848

D. Chikurtev, Face detection and recognition for intelligent service robots, PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, Vol 68, Sofia, 2017, ISSN 0204-9848

D. Chikurtev, Vision system for recognizing objects using Open Source Computer Vision (OpenCV) and Robot Operating System (ROS), PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, Vol 68, Sofia, 2017, ISSN 0204-9848

Цитати:

Едно цитиране на:

Chivarov, N.; Shivarov, S.; Yovchev, K.; **Chikurtev, D.**; Shivarov, N. - Intelligent Modular Service Mobile Robot ROBCO 12 for Elderly And Disabled Persons Care; Robotics

in Alpe-Adria-Danube Region (RAAD), 2014 23rd International Conference on, Smolenice, Slovakia, 3-5 Sept. 2014, Print ISBN: 978-1-4799-6797-1. p. 343-348

Цитиране от:

Koubâa, Mohamed-Foued Sriti, Yasir Javed, Maram Alajlan, Basit Qureshi, Fatma Ellouze, Abdelrahman Mahmoud, "Turtlebot at Office: A Service-Oriented Software Architecture for Personal Assistant Robots Using ROS", Autonomous Robot Systems and Competitions (ICARSC) 2016 International Conference on, pp. 270-276, 2016.

Място на цитиране:

Autonomous Robot Systems and Competitions (ICARSC) 2016 International Conference on

УВОД

Сервизната работи е област, която се развива с огромни темпове през последните години. Идеята на тези работи е да заменят част от ежедневните човешки дейности, които са свързани с работа във вредни среди, работа на труднодостъпни места, извършване на подпомагащи дейности, инспектиране на различни места и други. Повечето от сервизните работи са специализирани за конкретни задачи, понеже направата на универсален робот и много сложно и скъпо начинание. Класификацията на сервизните работи се прави според тяхното предназначение.

В този труд са засегнати някои фундаментални проблеми като навигация, локализация и одометрия при мобилните работи, компютърно зрение и видео обработване и управление на антропоморфни роботизирани манипулатори.

Настоящата дисертационна работа е структурирана както следва:

ГЛАВА 1: анализирано е състоянието на пазара на сервизните работи. Анализирани са типовете сервизни работи според тяхното предназначение. разгледани са нуждите от сервизни работи за подпомагане на човека. Разгледани са проблемите при управление на сервизните работи в следните области: автономно движение на сервизните работи, взаимодействие на роботите с средата, хората и предмети, приложение на роботите за подпомагане на хората.

ГЛАВА 2: в тази глава е описан сервизния робот, който се използва за прилагане и провеждане на експерименти, свързани с работата по дисертацията. Описани са възможностите и принципа на работа на Операционна система за управление на работи (ROS). Описани са методи за оптимизиране на навигационна система и одометрията. Описани са методи за реализиране на компютърно зрение за разпознаване на лица и обекти. Представени са алгоритми за подобряване на споменатите системи.

ГЛАВА 3: в тази глава са засегнати специфични проблеми при сервизните работи. Представени са методи за дистанционно управление на роботите чрез различни устройства и интерфейси. Разгледани са възможности за заснемане на изображения с помощта на термо камера. Реализиран е интернет базиран потребителски интерфейс, който предоставя различни функции за управление. Показани са нови подходи за итеративно управление на антропоморфен манипулатор.

ГЛАВА 4: изложени са резултатите от проведените научни експерименти, върху работа. Направен е анализ на системите на работа, начина по който се е подобрило цялостното поведение на работа. Отчетени са точността при подобрената навигация и одометрия, успеваемостта при разпознаване на лица и предмети, прецизността при управление на работа посредством различни устройства и интерфейси. Всички експерименти са проведени в реални условия с помощта на работа.

ГЛАВА 1. Литературен обзор.

1.1. Дефиниция и класификация на сервизните работи.

1.1.1. Дефиниция за робот и сервизен робот.

В резултат на съвместните усилия започнали през 1995г., Икономическата комисия на ООН (UNECE) и Международната федерация по Роботика (IFR) [1] се ангажират върху създаване на определение за сервизен робот и схема за класифициране на сервизните работи. Според извлечения от ISO стандарта 8373 [2] дефинициите за робот и сервизен робот са:

- Робот е задвижван механизъм, програмируем в две или повече стави със степен на автономност, движещ се в своята среда за да изпълнява зададени цели. В този контекст автономност се разбира като способността да се изпълняват зададени цели според текущото състояние и наблюдение, без човешка намеса.
- Сервизен робот е робот, който изпълнява полезни задачи за хората или оборудване с изключение на приложения в индустриалната автоматизация. Трябва да се отбележи, че класификацията на робот дали е индустриален робот или сервизен робот се прави според предвиденото му приложение.
- Персонален сервизен робот или сервизен робот за персонално ползване е сервизен робот, който се използва за некомерсиални задачи, обикновено от хора на легло. Примери за такива работи са домашен робот сервитьор, автоматизиран инвалиден стол, персонален мобилен асистент робот и робот за забавление на домашните любимци.
- Професионален сервизен робот или сервизен робот за професионални ползване е сервизен робот за комерсиални цели. Обикновено тези работи се управляват от обучен оператор. Примери за такъв робот са почистващ робот на обществени места, робот за доставки в офисите или болниците, робот за гасене на пожари, робот за рехабилитация, робот, извършващ операции в болниците и други. В този контекст оператор на сервизен робот е човек, който е правоспособен да стартира, наблюдава и прекратява дадено действие на робот или роботизирана система.

Роботизирана система е система съставена от робот(и), изпълнителен краен механизъм и всякакви други механизми, устройства или сензори, които да подпомагат робота, докато той изпълнява своите задачи.

Във връзка с дефиницията за сервизен робот, манипулационните индустриални работи (които мога да се монтират на дадено място или на мобилна платформа), също могат се разглеждат като сервизни работи, при условие, че те са инсталирани в непроизводствени процеси. Сервизните работи могат да са оборудвани с манипулатор или да не са оборудвани с такъв, както в някои случаи при промишлените работи. Много често, но не винаги сервизните работи са мобилни [3].

В някои случаи, сервизните работи се състоят от мобилна платформа, на която са монтирани една или няколко ръце и се управляват в същия режим, както ръцете на индустриален робот. Освен това за разлика от своите индустриални аналози, сервизните работи не трябва да бъдат автоматични или автономни. В много от

случаите тези машини могат да подпомагат човека потребител или да бъдат теле – управлявани.

Поради многообразието на форми и структури, както и на области на приложение е трудно да се направи точно определение за сервизен робот.

1.2. Резюме и прогнози на Международната Федерация по Роботика за развитието и продажбите на сервизни работи през последните години.

1.2.1. Резюме за периода 2014-2015 г.

В проучването на международната федерация по роботика, са докладвани 707 броя роботизирани мобилни платформи за обща употреба, 32% повече спрямо 2014г. Продажбите на работи за професионално почистване са се удвоили през 2015г. в сравнение с тези през 2014 до почти 600 броя. Около 568 строителни и разрушителни системи са доставени през 2015г., като се предполага че продажбите са повече. Доставени са 275 системи за инспекция и поддръжка и 131 подводни системи. Подводните системи са сред най-ценните професионални сервизни работи със средна цена от около 0.69 млн. долара. Общата стойност на продажбите е около 91 млн. долара, което представлява дял от 2% от общата стойност на продажбите на професионални сервизни работи за 2015г. Този брой може да бъде значително по-голям.

Роботите асистенти за хора в затруднено положение, през последните години изпълняват все повече от изискванията за тяхната работа. През 2015г. общо 4713 работи са продадени, в сравнение с 4416 през 2014 – увеличение с 7%. Многобройните национални изследователски проекти в различни държави се фокусират върху този огромен бъдещ пазар за сервизни работи. За разлика от домакинските и развлекателните работи, тези работи са високотехнологични продукти.

Изчисленията показват, че през 2015г. са продадени повече от 3.7 млн. работи за битови нужди, включително работи за почистване, косене на трева, миене на прозорци и други. Което е с 11% повече отколкото за 2014г. Реалният брой, обаче може да бъде значително по-голям, тъй като проучването на МФР не може да обхване изцяло световния пазар на работи. Стойността на продажбите е около 1.2 милиарда долара. В сравнение с 2014г. те са се покачили с 1%.

1.2.2. Описание на развитието и проблемите при сервизните работи за подпомагане на човека.

В този научен труд се разглеждат сервизни работи, които да подпомагат хората, както в техните домове, така и в обществени сгради като болници, музеи, офиси и други. Роботите предназначени за домашни помощници трябва да могат да помагат на хората, като извършват повечето от техните ежедневни дейности.

През последните десетилетия са разработени много роботизирани платформи, повечето от които включват мобилност, някаква автономна навигация и - напоследък - и манипулационни способности. Списъкът на обслужващите работи, изобразен на фиг. 1.5 далеч не е завършен, показвайки само тези с по-сложна функционалност. Трудно е обаче да се намерят платформи за сервизни работи, които освен това съдържат способност за визия с разпознаване на обекти, разпознаване на жестове и моделиране на сцени,

изчерпателни разсъждения и планиране на компоненти и разработване на концепции за взаимодействие на потребителите.

Повечето от тези работи са общи: те са чисти платформи за разработка с малък акцент върху свързаните с крайния потребител въпроси като дизайн или използваемост. Целта е да се разработи цялостна концепция, подходяща за продуктова визия, съчетаваща горепосочените технологични аспекти с компактен и лесен за използване дизайн.

Техническите средства позволяват на възрастните хора и хората с увреждания да живеят независимо в своите домове, ако те пожелаят. Именно сервизните работи са тези технически средства, които да изпълняват тези функции [7]. Тези работи могат да вземат и носят различни предмети или да изпълняват други задачи за подпомагане на хората. ето защо роботизираният домашен асистент трябва да може да изпълнява следните задачи:

- Домакински задачи:
 - Вземане и носене на обекти, например книги, лекарства и др.;
 - Подпомагане при хващане, носене и вдигане на предмети;
 - Извършване на ежедневни задачи като сервиране на напитки, приготвяне на масата, работа с микровълнова печка, прости задачи за почистване;
 - Контрол на домашната инфраструктура, например отоплителната система, климатик, осветление, прозорци, врати, алармена система и др.
- Помощ при движение:
 - Подпомагане при ставане от легло или стол;
 - Интелигентен помощник за ходене.
- Комуникативност и социална интеграция:
 - Управление на медийните устройства (телевизия, стерео и др.);
 - Мениджър за напомняне (време за прием на лекарства и др.);
 - Надзор на жизнените показатели и спешно повикване.

От сега и в бъдеще роботите ще ни помогнат да се възстановим след удар, като заменят тежката работа на рехабилитаторите. Като се добавят и факторите като повторяемост, точност и издръжливост, роботите могат да се превърнат в по-добри системи за рехабилитация, а самите рехабилитатори да определят методите за лечение на пациентите. В крайна сметка те биха могли да се насочат към леглото, или дори да заменят психиатри [11] [12].

ГЛАВА 2. Алгоритми за управление, навигация и локализация и компютърно зрение.

2.1. Описание на характеристиките на мобилния робот за подпомагане на човека

Робота върху, който се провеждат всички изследвания и експерименти е специално създаден за учебни и изследователски цели (фигура 2.1). Съставните модули на робота са мобилна платформа, антропоморфна роботизирана ръка РОВКО 01, два контролера за

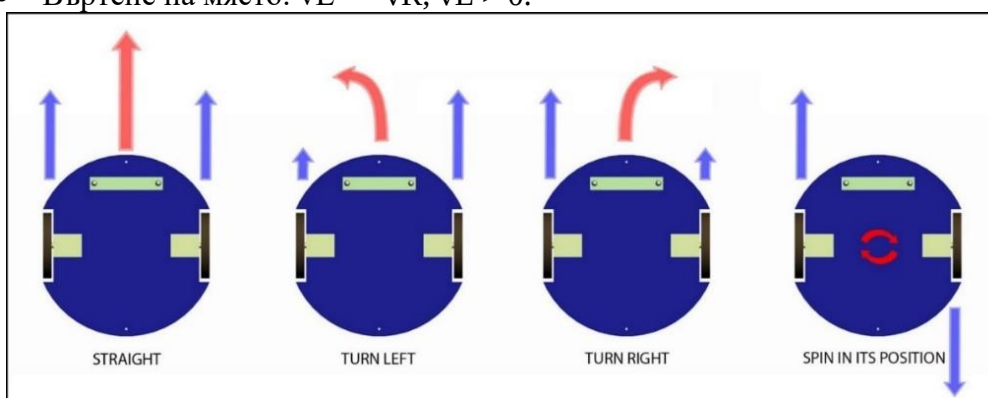
управление на електромоторите на платформата и ръката, Кинект сензор, инфрачервени и ултразвукови сензори и батерии.



Фигура 2.1 Сервизен мобилен робот

На фигура 2.2 е показан принципа на управление на диференциално разположените колела. Ако и двата мотора се въртят с еднаква скорост, роботът ще се движи право напред или назад. Ако моторите се въртят с различна скорост, тогава роботът ще се движи по траектория с определен радиус. И когато моторите се въртят с еднаква скорост в противоположни посоки, роботът ще се върти около центъра си:

- Движение право напред/назад: $v_L = v_R$, $v_L > 0$
- Движение по дадена траектория: $v_L > v_R$, e.g. $v_L = 2v_R$
- Въртене на място: $v_L = -v_R$, $v_L > 0$.



Фигура 2.2. Диференциално управление

2.2.Операционна система ROS

ROS (Robot Operating System) [35] е мета-операционна система, която има за цел да осигури единно решение на софтуер за различни видове роботи. Терминът мета-операционна система означава, че вече има инсталирана операционна система на

хардуера и ROS е инсталирана върху тази операционна система. За момента Linux дистрибуцията Ubuntu е операционната система, която поддържа изцяло ROS.

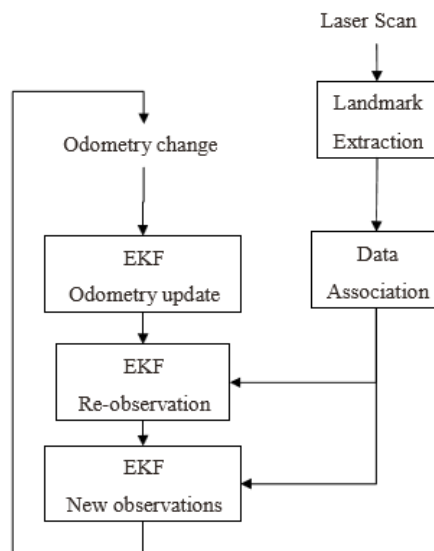
ROS се състои от три нива: файлово ниво, изчислително ниво и ниво общност.

2.3. Навигация и локализация

Навигационните системи за мобилни роботи се нуждаят от достатъчно надеждно измерване на текущото местоположение на робота и прецизна карта на зоната за навигация. Тези системи се разделят на две нива на управление: глобално планиране на път и управление на локалното движение.

Локалните навигационни системи са способни да изчисляват нов път в следствие на промени в средата. Тези системи може да се разделят на два подхода, единият е за посоките, а другият е за ускоренията [40]. Подходите за определяне на посоката като метода за потенциала на полето [41], виртуалното поле на силите [42], които се разгръщат до хистограма на векторното поле [43] и алгоритъм за диаграма на разстоянията [44], определят посоката, по която роботът се движи. Подходите за ускорение в пространството като Curvature Velocity method [45], Lane Curvature method [46] и Dynamic Window method [47], изпълняват функции за намиране на правилните команди за управление на робота като транслационното и ротационното ускорение директно върху пространството.

За реализиране на навигационната система на робота използваме метода SLAM (Simultaneous Localization and Mapping) – едновременно локализация и картографиране [48]. SLAM процесът (фигура 2.8) се състои от няколко стъпки. Основната цел на процеса е използва данните от средата, в която се намира роботът за да опреснява позицията му.



Фигура 2.3. Принципна схема на SLAM процесът

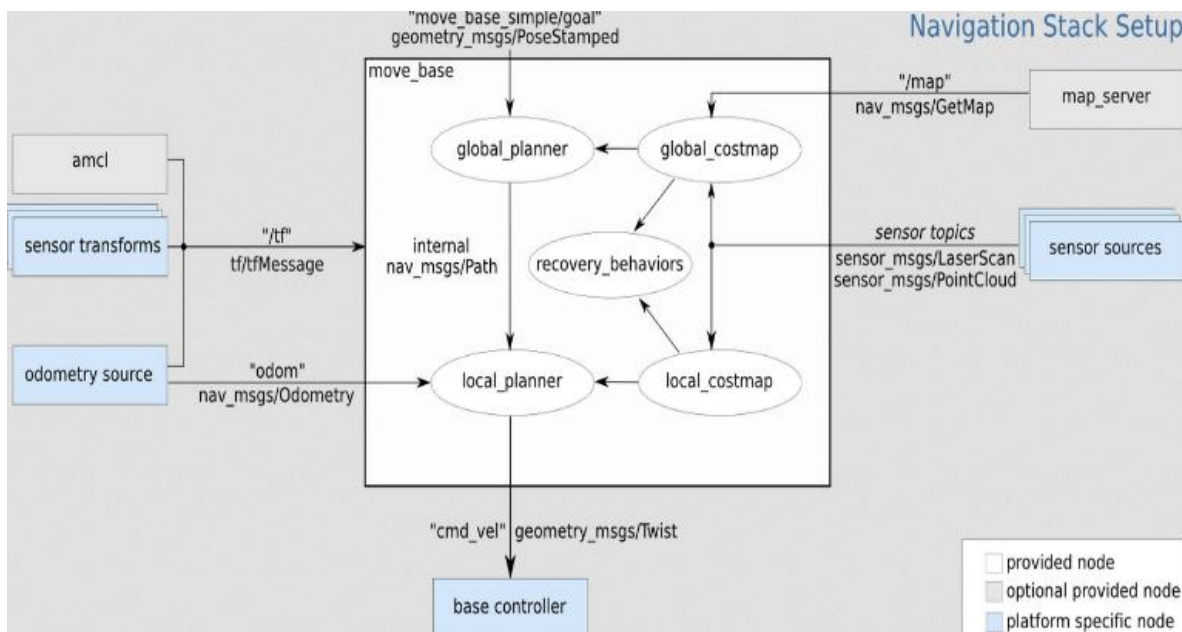
2.3.1. Конфигурация на данните от сензорите в средата на ROS

Четенето и публикуването на данните от сензорите чрез ROS [49] е важно за правилното и безопасно използване на навигационния пакет. Съществуват много сензори, които могат да се използват за предоставяне на информация на навигационния пакет: лазерни скенери, камери, ултразвукови сензори, инфрачервени сензори, електромеханични сензори и други.

За работи с лазерни скенери, ROS предоставя специален тип съобщение от пакет със съобщенията, наречен LaserScan. Пакетът който предоставя възможността на ROS да се свързва и обменя данни с дълбочинни сензори се нарича openni_camera. Топиците, от които се нуждаем са за дълбочинния сензор на Kinect-a:

2.3.2. Навигационен пакет в ROS

Навигационният пакет в ROS е прост процес на концептуално ниво. Той взема данни от одометрията и сензорните системи и връща команди за ускорението като изходен сигнал за управление на мобилната платформа на робота. Приложението на навигационния пакет върху произволен робот обаче е доста сложна задача.



Фигура 2.4. Принципна схема на работата на Навигационния пакет

Навигационния пакет изисква от робота да публикува информацията за връзките между координатните рамки като се използва библиотеката „tf“ [50].

2.3.3. Информация от одометрията

Одометрията е много важен метод, използван за изчисляване на текущата позиция на мобилни системи. Колкото по точни са одометричните данни толкова по добри са точността и прецизността на позициониране. Реализиран е алгоритъм за изчисляване на одометричните данни.

Описание на алгоритъма:

- Изчисляване на изминатата дистанция за всяко колело поотделно в метри:

$$\text{left_distance} = (\text{current_left_encoder_ticks} - \text{last_left_encoder_ticks}) / \text{LEFT_TICKS_PER_METER}$$

$$\text{right_distance} = (\text{current_right_encoder_ticks} - \text{last_right_encoder_ticks}) / \text{RIGHT_TICKS_PER_METER}$$

$$\text{last_left_encoder_ticks} = \text{current_left_encoder_ticks}$$

$$\text{last_right_encoder_ticks} = \text{current_right_encoder_ticks}$$
- Изчисляване на общо изминатата дистанция:

$$\text{distance} := (\text{left_distance} + \text{right_distance}) / 2.0$$
- Изчисляване на завъртането на робота:

$$\text{theta} := \text{theta} + (\text{left_distance} - \text{right_distance}) / \text{DISTANCE_BETWEEN_WHEELS}$$

- Определяне на сегашната позиция

$x := x + \text{distance} * \cos(\text{theta})$

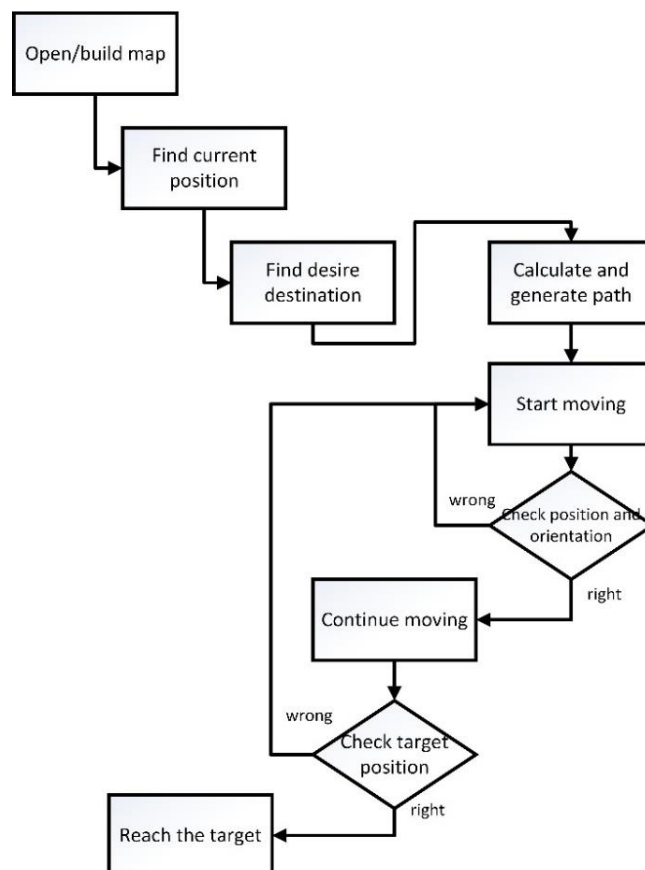
$y := y + \text{distance} * \sin(\text{theta})$

тези изчисления са необходими на библиотеката „tf“ за да може да се реализират функциите за генериране на път и за управление на мобилната платформа.

2.3.4. Управление на платформата

Навигационният пакет може да изпраща команди за скоростта като използва `geometry_msgs/Twist message`, при положение, че всичко е синхронизирано с координатната рамка на робота на топика `cmd_vel`. Това означава, че трябва да има възел, който да подсказва на `cmd_vel` топика, че може да приема параметрите за скоростта и да ги преобразува в команди за моторите, които да изпраща към мобилната платформа (v_x, v_y, v_{θ}) $\Leftrightarrow$ (`cmd_vel.linear.x`, `cmd_vel.linear.y`, `cmd_vel.angular.z`).

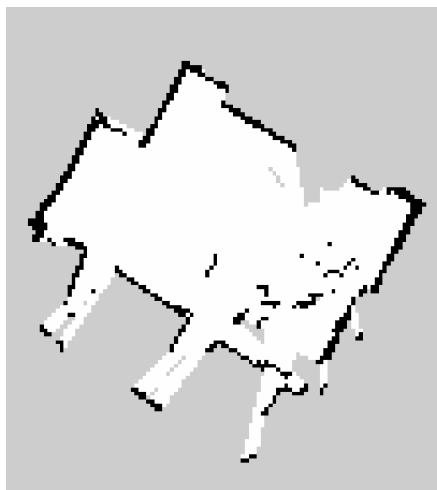
Алгоритъмът за подобряване на навигацията на робота е показан на фигура 2.10. В началото на алгоритъма е зареждане на картата на средата или създаване на нова карта ако средата е непозната. Алгоритъма за изчисляване на път е използвания в изкуствения интелект алгоритъм A^* , който винаги намира най-кратък път до целта.



Фигура 2.5. Алгоритъм за оптимизиране на навигацията

2.3.5. Картографиране - Mapping

Тази функция се използва за да се създават и обработват карти на средата. Тя е изключително полезна за навигацията. Всички карти могат да се използват от навигационния пакет за да се постигнат по-добри резултати при ориентиране и позициониране. На фигура 2.11 е показана карта на едно от сканираните помещения, направена в ръчен резим на сканиране – управляваме робота с джойстик.



Фигура 2.6. Карта направена от робота.

Картата се генерира като роботът обхожда и сканира желаната среда. За създаване на самоата карта се грижи нода “slam_gmapping”.

2.3.6. Локализация и разпознаване на помещения и сгради

Основната идея на тази разработка е при стартиране роботът да направи сканиране от място на заобикалящата го среда и от заснетото да разбере каква е неговата точна позиция в помещението, в което се намира. След това сравнява направената картина дали съвпада с някоя от предварително направените карти, намиращи се в базата данни да робота.

2.3.7. Алгоритъм за сканиране при старт.

В основата на алгоритъма стои пакета „gmapping“ [51]. Този пакет ни дава възможност да реализираме SLAM процес, като използваме лазерния скенер на Kinect сензора. „slam_gmapping“ е името на възела в операционната система за роботи, чрез който можем да създаваме двумерни картни на помещенията. За създаване на тези карти се използват данните от лазерния скенер и позицията на робота.

Последователността на изпълнението на отделните задачи в алгоритъма за сканиране при стартиране на робота са както следва:

- ✓ Включване на робота
- ✓ Стартиране на пакетите gmapping и navigation stack
- ✓ Завъртане на робота на 360 градуса за да сканира
- ✓ Сравнение на сканираната карта с базата данни
- ✓ Ако има съвпадение – преминава в режим на готов за работа
- ✓ Ако няма съвпадение – стартира процедура по сканиране на новото помещение
- ✓ След приключване на новото сканиране се запамятава новата карта
- ✓ Робота отива в изходна позиция в режим готов за работа.

При стартиране на робота този алгоритъм се активира автоматично ако не подадем никакви команди на робота 10 секунди след старта.

2.3.7.1. Сравняване на новата карта с запаметените карти.

За да се реализира началното сканиране, се използват едновременно пакетите gmapping и navigation stack. Когато робота направи сканирането получената карта се сравнява с предварително направените карти запаметени в базата данни на робота.

При намиране на съвпадение робота прекратява процедурата по сканиране на помещението и преминава в режим на очакване на задачи от потребителя. В този случай пакета gmapping се спира и робота започва да се управлява само от navigation stack.

2.3.7.2. Режим на запис на нова карта.

Когато този режим се активира робота обхожда помещението, в което се намира като в това време е активиран пакета gmapping заедно с навигационния пакет. Робота се движи докато не получи пълна картина на помещението, след което се връща в изходната си позиция. Ново-сканираната карта се добавя към базата данни на робота и се деактивира пакета gmapping. Този етап от алгоритъма все още е в процес на разработване и експерименти.

2.4.Коригиране на одометрични грешки

Одометрията се основава на уравнения, които се изчисляват сравнително лесно и използват данни от енкодерите на всяко от колелата. Все пак одометрията разчита на представянето, че оборотите на колелата могат да се преобразуват в линейно преместване спрямо пода.

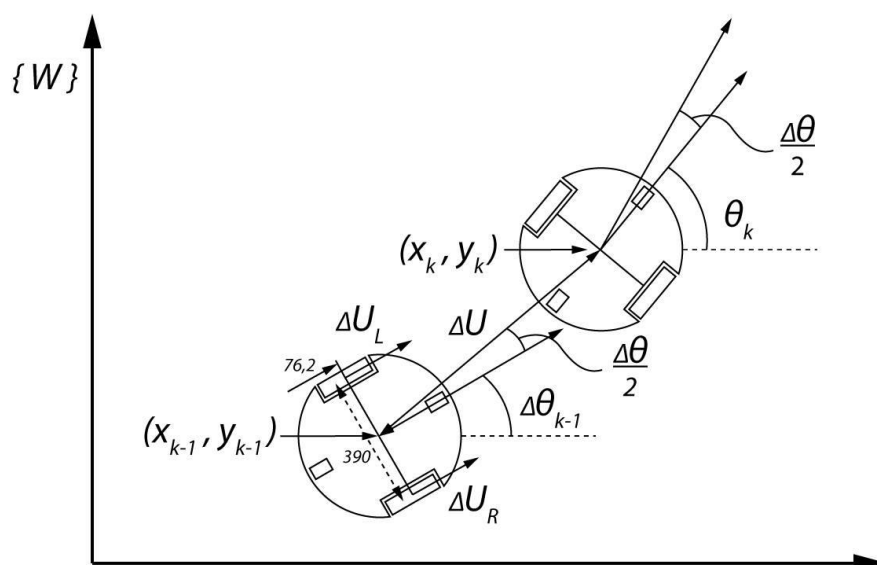
За да се изчисли изминатото разстояние на едно колело трябва да се преобразуват импулсите от енкодера в линейно преместване (c_m):

$$c_m = \pi D_n / n C_e ;$$

Където, D_n – диаметър на колелото; C_e – резолюция на енкодера; n – предавателни число на редуктора между мотора и колелото;

Така след кратък интервал I енкодерите на двете колела показват натрупаните импулси за всяко колело N_L и N_R . След това общото изминато разстояние за всяко колело е:

$$\Delta U_{L,I} = c_m N_{L,I}; \Delta U_{R,I} = c_m N_{R,I} ;$$



Фигура 2.7. Параметри на одометрията.

2.4.1. Коригиране на систематичните грешки.

Основните грешки, които оказват влияние са: разлика в диаметъра на колелата (E_d) и нееднаквост на колесната платформа на робота (E_b). Коригирането на грешките става, като се измери тяхното влияние върху поведението на робота, след което това се компенсира софтуерно при генериране на управляващите сигнали към моторите.

Както е показано в [54], грешката E_b се проявява само, когато имаме завиване, а грешката E_d оказва своето влияние върху движението на платформата, когато тя се движи направо. E_b и E_d са неизмерими стойности, изразени като процент от номиналната стойност. Можем да дефинираме:

$$E_d = D_L/D_R \quad ;$$

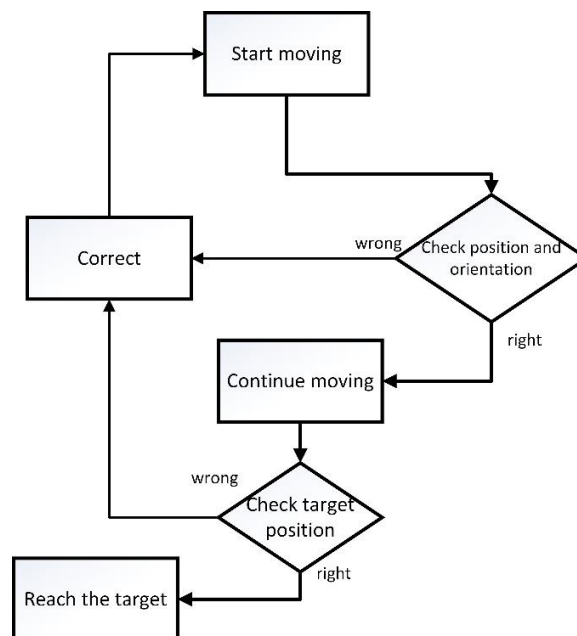
Където D_L и D_R са съответно действителните диаметри на лявото и дясното колела. Също така можем да дефинираме:

$$E_b = b_{actual}/b_{nominal} \quad ;$$

Където b е междуосието на мобилната платформа.

2.4.2. Коригиране на несистематичните грешки.

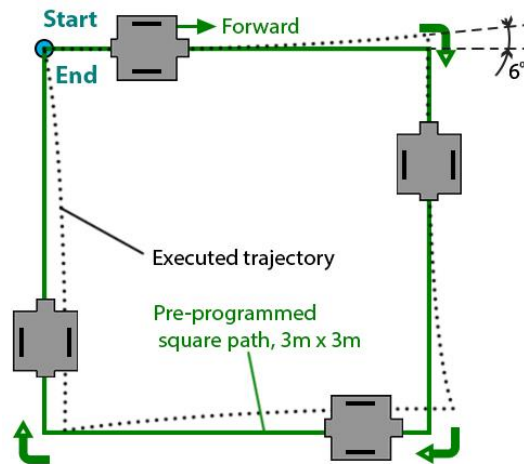
Понеже следено за отклонение се случва в реални време тук не се налага да се изчисляват определени грешки, а се прилага алгоритъм за коригиране на позицията на робота. Този алгоритъм е добавен като част от SLAM процеса, като основната му цел е да подобри цялостната работа на навигационната система на робота. Алгоритъма е показан на фигура 2.13, като основната му функция е при измерено отклонение в ориентацията или движението на робота, да се коригират тези отклонения като се регулират допълнително ускоренията и скоростите към моторите на колелата.



Фигура 2.8. Алгоритъм за коригиране на несистематичните грешки в реално време.

2.4.3. Коригиране на систематичната одометрична грешка чрез сензор за ориентация и ускорение

За да определим систематичните грешки използваме принципа на двупосочното обхождане на път във формата на квадрат. Робота е програмиран да прави 90-градусови завои по посока на часовника и обратна на часовника (фиг. 2.14).



Фигура 2.9. Обхождане на път във формата на квадрат.

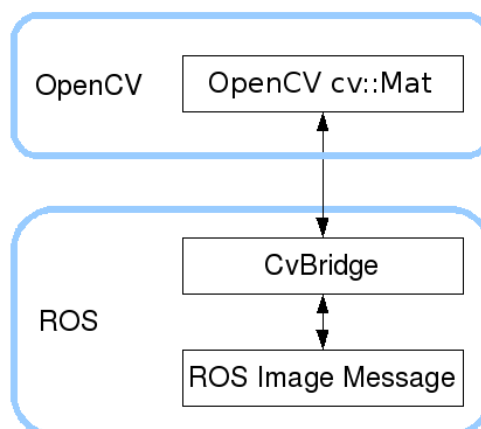
След провеждане на този експеримент се установи, че използването на данните от енкодерите не е достатъчно ако искаме да измерваме прецизно позицията на робота. Получената грешка е голяма, поради неравности на пода и боксуване на колелата в някои моменти. Поради това добавихме вградено измервателно устройство (IMU) [55] към робота.

2.5. Компютърно зрение

2.5.1. Съвместимост между ROS и OpenCV

Нашият робот се контролира от ROS. Налице е vision_opencv стек, който осигурява опаковане на OpenCV библиотеката за ROS. Има два основни пакета за използване на OpenCV в ROS:

- cv_bridge: осигурява мост между ROS и OpenCV за изпращане и получаване на съобщения;
- image_geometry: набор на методи за работа с геометрия на изображения и пиксели.



Фигура 2.10. комуникация между ROS и OpenCV

Image_geometry [58] съдържа библиотеки Python и C ++, които опростяват геометричното тълкуване на изображения чрез параметрите от sensor_msgs / CameraInfo.

2.5.2. Намиране / разпознаване на обекти

Както казахме по-рано нашата система за компютърно зрение включва Kinect и стерео камера. Kinect-ът има RGB камера, която използваме за разпознаване и сензор за дълбочина, за да измерим разстоянието до обекта. VideoCapture може да изтегли следните данни:

- a. Данните, дадени от генератора на дълбочината:
 - o CV_CAP_OPENNI_DEPTH_MAP - дълбочинни стойности дадени в милиметри (CV_16UC1);
 - o CV_CAP_OPENNI_POINT_CLOUD_MAP - XYZ в метри (CV_32FC3);
 - o CV_CAP_OPENNI_DISPARITY_MAP - несъответствие в пикселите (CV_8UC1);
 - o CV_CAP_OPENNI_DISPARITY_MAP_32F - несъответствие в пикселите (CV_32FC1);
 - o CV_CAP_OPENNI_VALID_DEPTH_MASK - маска на валидни пиксели (не е скрита, не е засенчена и т.н.) (CV_8UC1).
- b. Данни, дадени от BGR генератор на изображението:
 - o CV_CAP_OPENNI_BGR_IMAGE – цветно изображение (CV_8UC3);
 - o CV_CAP_OPENNI_GRAY_IMAGE – сиво изображение (CV_8UC1).

За да разпознаем обектите, получаваме краищата на сцената. След това системата търси конкретни форми. Стойността L съдържа стойност за лекота на входното изображение, където L се изчислява, както е показано:

$$L = \frac{\max(R,G,B) + \min(R,G,B)}{2};$$

Където L е средната стойност, R е червеният канал на входното изображение, G е зеленият канал на входното изображение и B е синият канал на входното изображение.

2.5.3. Намиране / разпознаване на лица

Разпознаването на лица като цяло и разпознаването на придвижването на хора в естествени сцени по-специално изискват набор от визуални задачи, които да бъдат изпълнени с голяма степен на робасност [59].

Показано е от Дейвид Хубел и Торстен Уейлз [60], че мозъкът ни има специализирани нервни клетки, които отговарят на специфичните местни особености на сцена, като линии, ръбове, ъгли или движение.

Една от първите автоматизирани системи за разпознаване на лица е описана в [61]: маркерните точки (положението на очите, ушите, носа,) са използвани за изграждане на вектор на функция (разстояние между точките, ъгъл между тях,). Някои от най-новите разработки на геометричното разпознаване на лица са извършени в [62].

2.5.3.1. Откриване на лица.

OpenCV вече съдържа много предварително обучени класификатори за лице, очи, усмивка и др. Тези XML файлове се съхраняват в папката opencv / data / haarcascades /. Да създадем детектор на лице и очи с OpenCV.

Първо трябва да заредим необходимите XML класификатори. След това заредаме входното изображение (или видеоклипа) в режим на сивото..

```
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
img = cv2.imread('sachin.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Сега откриваме лицата в образа. Ако се намерят лица, то връща позициите на откритите лица като Rect (x, y, w, h). След като получим тези местоположения, можем да създадем ROI за лицето и да приложим откриване на очите на тази ROI (тъй като очите са винаги на лицето).

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

2.5.3.2. Разпознаване на лица.

OpenCV 2.4 сега идва с новия клас FaceRecognizer за разпознаване на лица, който осигурява единен достъп до всички алгоритми за разпознаване на лица в OpenCV. Всеки FaceRecognizer е алгоритъм (Algorithm), така че лесно може да се получат / зададат всички вътрешни модели (ако това се позволява от разпознаването).

Algorithm осигурява следните характеристики за всички извлечени класове:

- Така нареченият "виртуален конструктор". Това означава, че всеки производствен алгоритъм е регистриран при стартиране на програмата и можете да получите списъка с регистрирани алгоритми и да създадете копие на определен алгоритъм с името му (**Algorithm :: create ()**).
- Ако сте използвали функцията за заснемане на видео от модул OpenCV highgui,
- Четене и записване на параметри от / към XML или YAML файлове.

Освен това всеки FaceRecognizer поддържа:

- Обучение на FaceRecognizer с FaceRecognizer :: train () върху даден набор от изображения (вашата база данни за лица!).
- Предсказване на дадено примерно изображение, което означава лице. Изображението се дава като метода Mat.
- Зареждане / запазване на състоянието на модела от / към даден XML или YAML.
- Настройка / Получаване на информация за етикети, която се съхранява като низ. Информацията за етикети за низове е полезна за запазване на имената на разпознатите хора.

Налични към момента алгоритми:

- Eigenfaces (**createEigenFaceRecognizer ()**)
- Fisherfaces (**createFisherFaceRecognizer ()**)

- Графики на локалните двоични шаблони (**createLBPHFaceRecognizer()**)

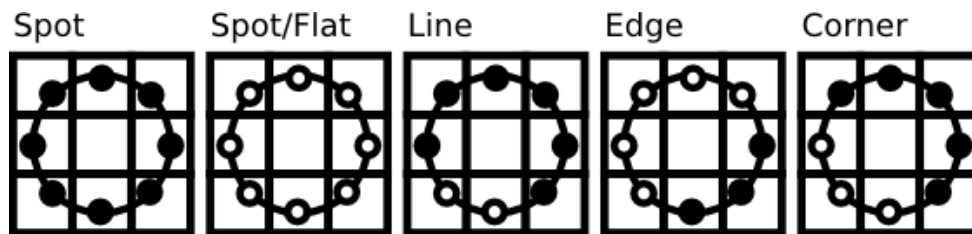
По-формално описание на оператора за локални бинарни шаблони (LBP) може да бъде дадено като:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

, $i_c(x_c, y_c)$ като централен пиксел с интензитет i_c ; и i_p представляващ интензитета на съседен пиксел. s е знаковата функция дефинирана както следва:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

Това описание дава възможност да се заснемат много фини подробности в изображенията. Така че операторът е разширен, за да използва променлива съседна функция в [63]. Идеята е да се приведе един съкратен брой съседи на кръг с променлив радиус, който позволява да се уловят следните квартали:



Фигура 2.11. Графики на локалните двоични шаблони – принцип на работа.

За дадена точка (x_c, y_c) позицията на съседна точка (x_p, y_p) , $p \in P$ може да се изчисли от следните изрази:

$$x_p = x_c + R \cos\left(\frac{2\pi p}{P}\right)$$

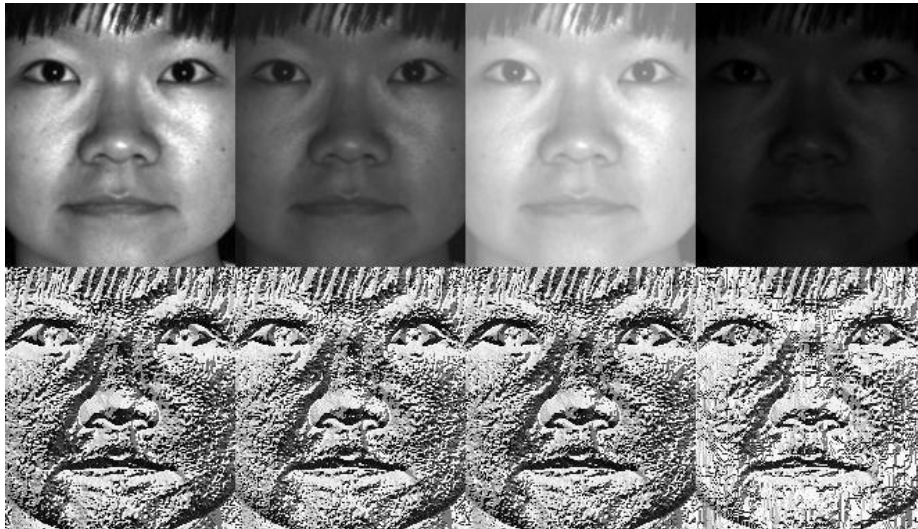
$$y_p = y_c - R \sin\left(\frac{2\pi p}{P}\right)$$

Където R е радиусът на кръга, а P е броят на единичните точки.

Операторът е разширение на оригиналните LBP кодове, така че понякога се нарича разширен LBP (наричан още "кръгов LBP"). Ако координатите на точки в окръжността не съответстват на координатите на изображението, то то се интерполира. Компютърната наука има куп интелигентни интерполационни схеми, изпълнението на OpenCV прави двулинейна интерполация:

$$f(x, y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}.$$

По дефиниция операторът на LBP е устойчив срещу монотонни сиви мащабни трансформации. Можем лесно да проверим това, като разгледаме изображението на изкуствено модифицирания образ на LBP:



Фигура 2.12. LBP изображение след обработка.

ГЛАВА 3. Сервизни роботи за специфични цели.

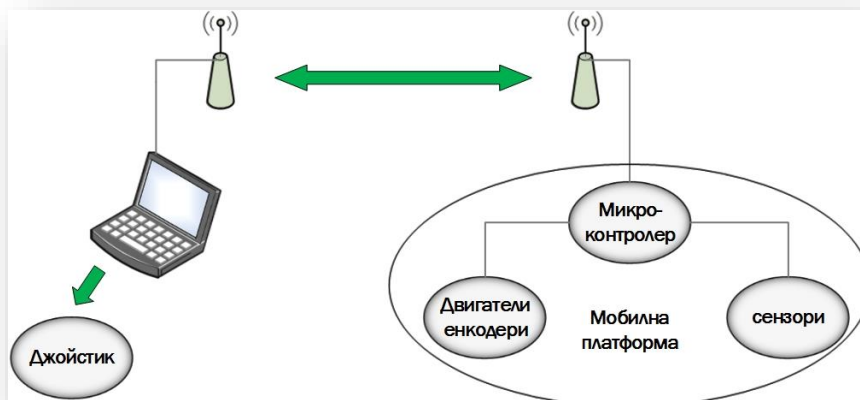
3.1. Дистанционно управление на сервизни роботи за инспекция с термо камера.

3.1.1. Управление на мобилната платформа.

В тази точка са описани системата за ръчно управление на робота, както и са представени някои особености при заснемане на изображения с термо-камера.

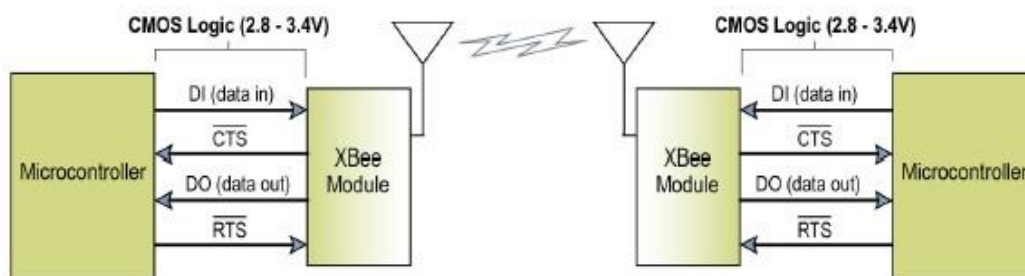
Схемата на фигура 3.1 показва как ще се осъществява управлението на робота. От едната страна на схемата е управляващият компютър свързан към джойстик и модул за безжична комуникация, а от другата е роботът, който също е свързан към модул за безжична комуникация.

Този тип свързване позволява да се използват изчислителната мощ и възможностите на компютъра за формиране на управляващите сигнали и обработка на получените данни от робота, което позволява работа с множество информация, реализиране на сложни алгоритми и бързодействие [64].



Фигура 3.13. Управление на робота за инспекция.

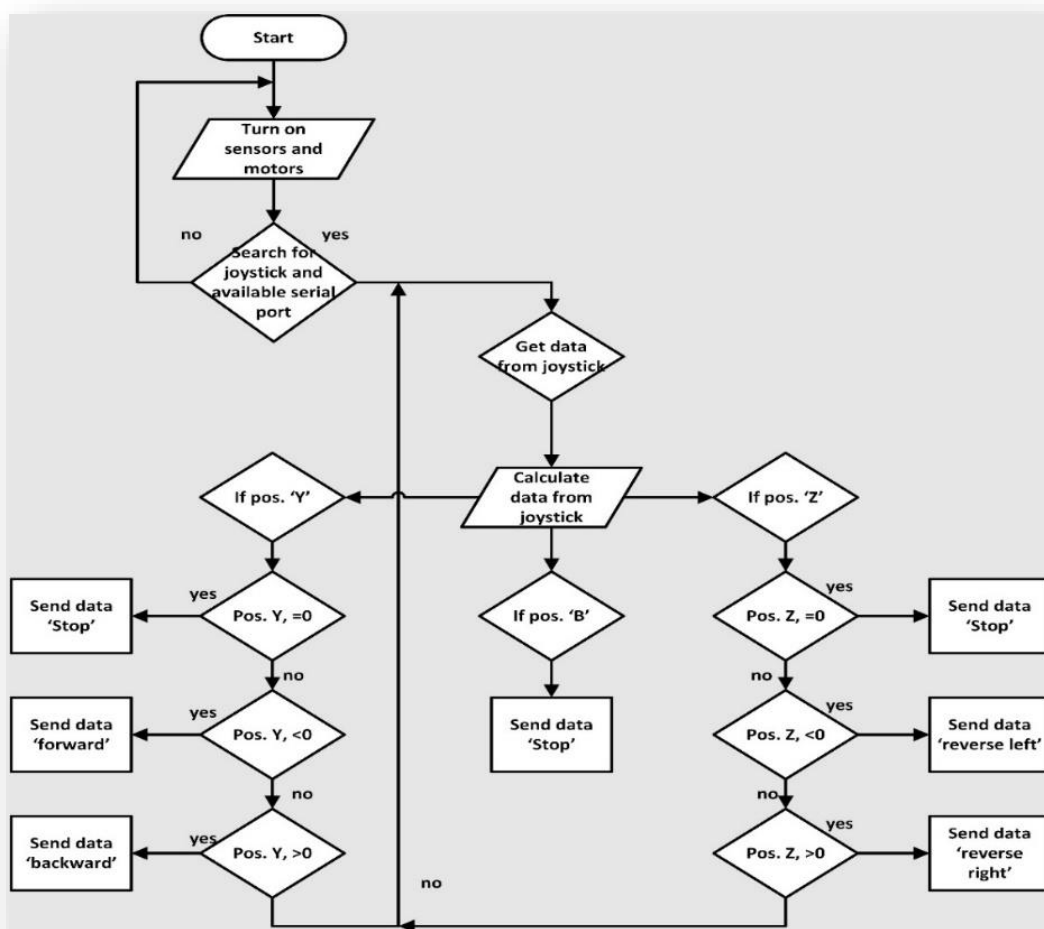
Използваният модул за безжична комуникация е X-bee модул. Принципа на работа на модулите и свързването им е показан на фигура 3.2. XBee модула се свързва с множеството устройства, през логическо ниво на асинхронен сериен порт [65].



Фигура 3.14. Принципна схема на комуникацията.

За реализиране на управлението на робота ще се разработи приложение написано на програмния език C#, върху платформата .NET, разработени от Майкрософт. C# е съвременен обектно-ориентиран език за програмиране с общо предназначение, създаден и развиван от Microsoft редом с .NET платформата [66].

Формирането на пакетите от данни става, чрез написания софтуер на база на алгоритъма за управление даден на фигура 3.3.



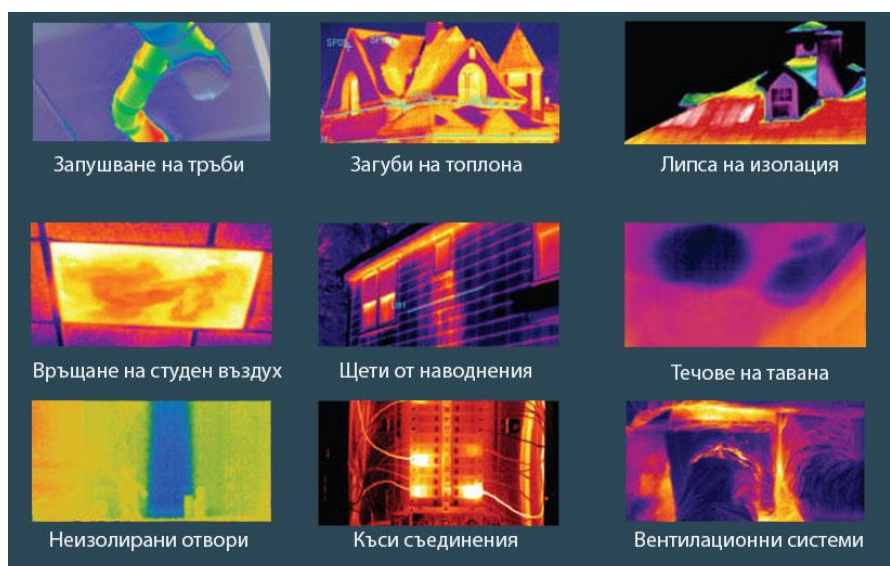
Фигура 3.15. Блок схема на алгоритъма за управление на робота.

„static void Main()“ е главната функция, която изпраща данните от другите функции към робота. Тя отваря серийния порт за изпращане на данните и изпраща пакети при получаване на нови параметри от другите функции.

„static void MainForJoystick()“ е функцията, която прочита данните от джойстика и го изпраща към третата функция. Има добавени няколко проверки преди да стартира работата на програмата, реализирани във функцията „MainForJoystick()“:

3.1.2. Заснемане и анализ на изображения с термо камерата.

Характеристики на инфрачервената термо камера FLIR P640: зрително поле 24°/16°, минимален диапазон на температурно измерване от -40°C до +500°C (опция до 2000°C), температурна чувствителност 0.06°C при 30°C (опция за 0.006°C), детекторна матрица 640/480 пиксела, интегрирана цветна цифрова камера, настройки на изображението, режими на измерване, различни обективи, памет 1 GB. На фиг. 3.4 са показани приложенията на термокамета в различни области и примерни изображения от заснемането им.



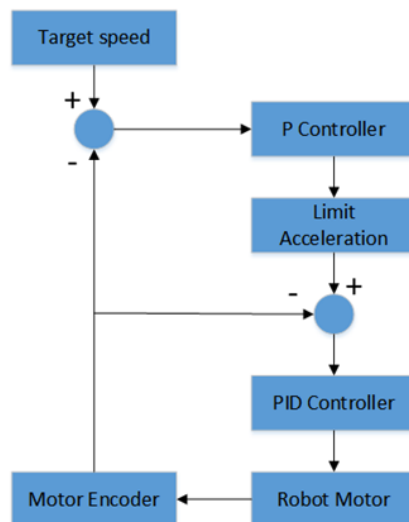
Фигура 3.16. Приложения на термо камерата.

Поради необходимост от заснемане на сложни обекти, се налага да се филтрират и анализират заснетите изображения. За анализ на направените изображения ще се използва специализирания софтуер FLIR Reporter, предназначен специално за работа с изображения.

3.2. Роботи за възрастни и инвалиди, ценово ориентирани решения

3.2.1. Управление на задвижването.

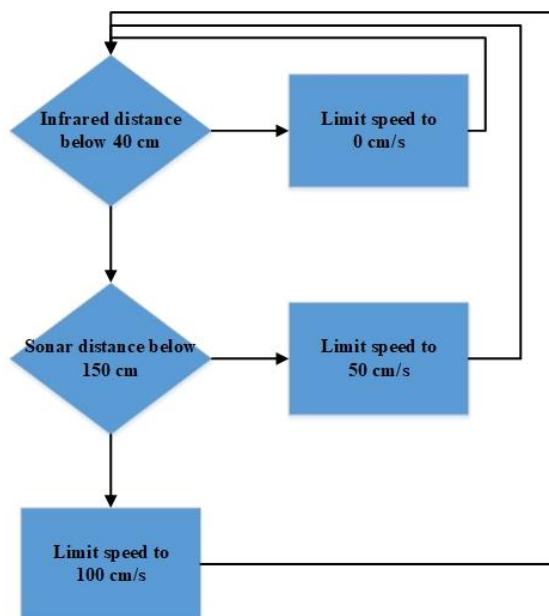
Следващото нещо, което направихме, за да постигнем стабилен контрол, беше използването на регулаторни органи въз основа на гореспоменатата PID техника. Този тип контрол беше интегриран във фърмуера на използвания Eddie Control Board в нашия робот. На всяко колело имаше два отделни контролера, работещи по следния принцип: (показано на фигура 3.6).



Фигура 3.17. Схема на управлението на двигателите.

3.2.2. Избягване на препятствия.

Следващото нещо, което направихме, беше да използваме данните от нашите сензори за разстояние, за да избегнем сблъсък с близките обекти. Използваните сензори имат следните ограничения: инфрачервеното може да усети обекти, които са много близо до робота, на не повече от 80 cm; Сонарният сензор имаше по-голямо разстояние - 300 cm от мобилната платформа.



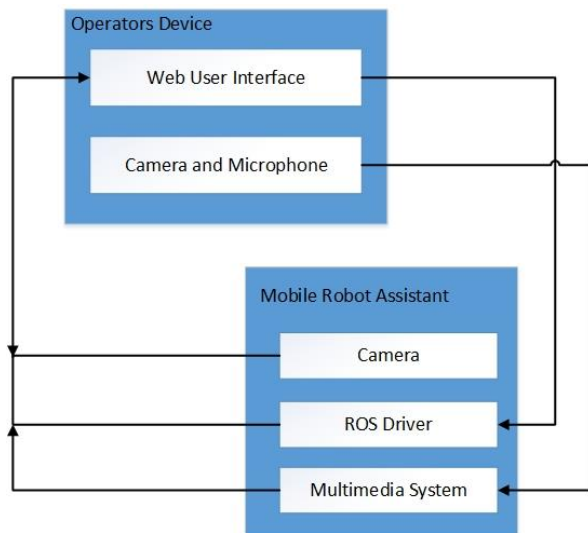
Фигура 3.18. Логика за ограничение на скоростите.

3.2.3. Изследване на методите за управление на сервизния робот.

- Физически джойстик:
- Виртуален джойстик:
- Метод за управление чрез проследяване на главата:
- Контрол чрез проследяване на очите:

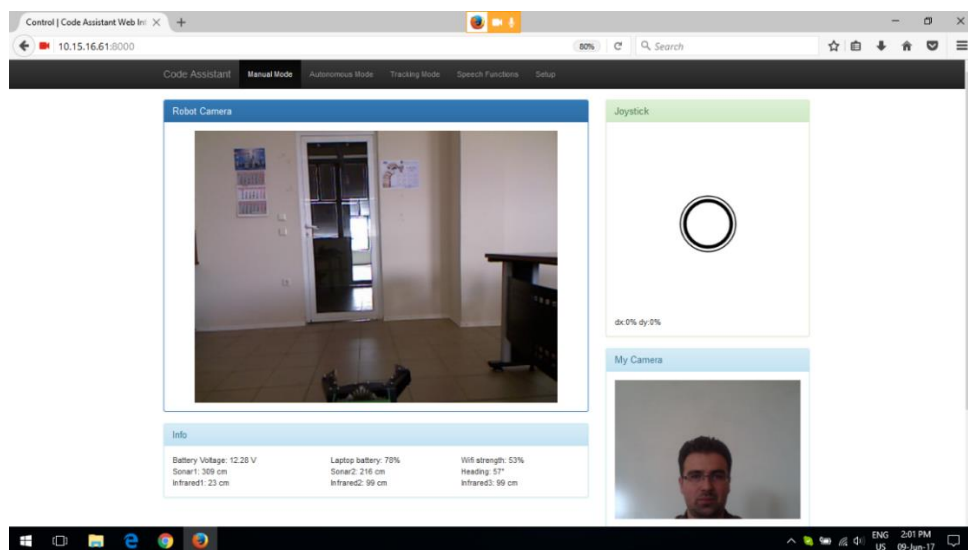
3.3. Потребителски интерфейс за управление на сервизни роботи

Комуникацията между робота помощник и потребителите е наистина важен въпрос. Така се определя колко бързо роботът ще се адаптира към системата и ще започне да използва пълния си потенциал. Ето защо създадохме потребителски интерфейс за управление на робота Фигура 3.8.



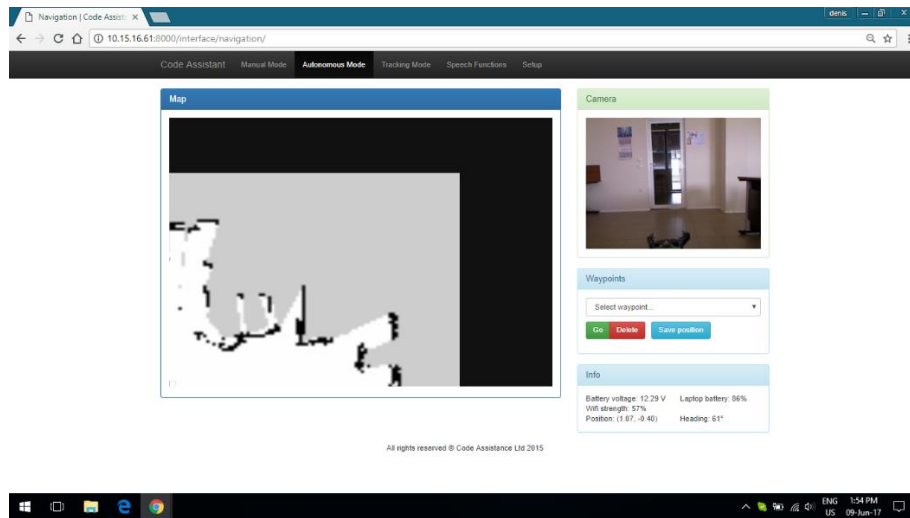
Фигура 3.19. Принцип на работа на интерфейса на робота.

3.3.1. Меню за управление в ръчен режим.



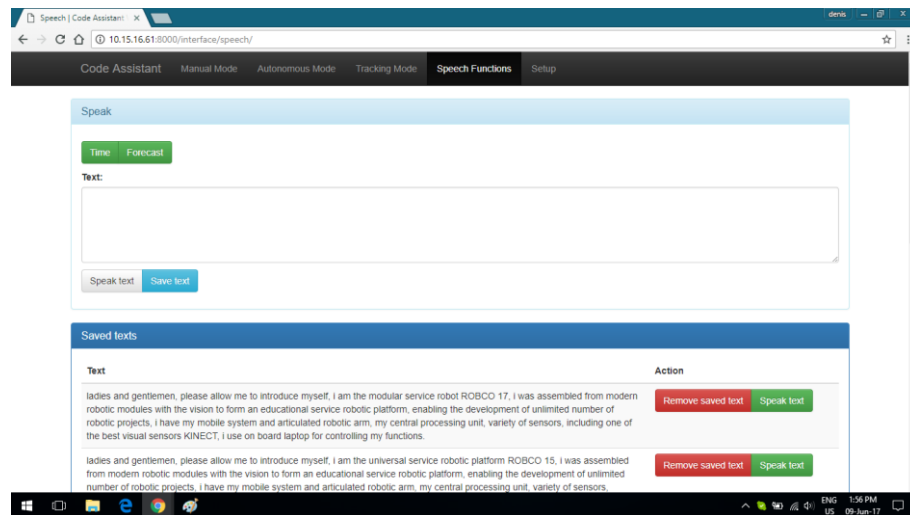
Фигура 3.20. Меню за управление в ръчен режим

3.3.2. Меню за управление в полу-автономен режим.



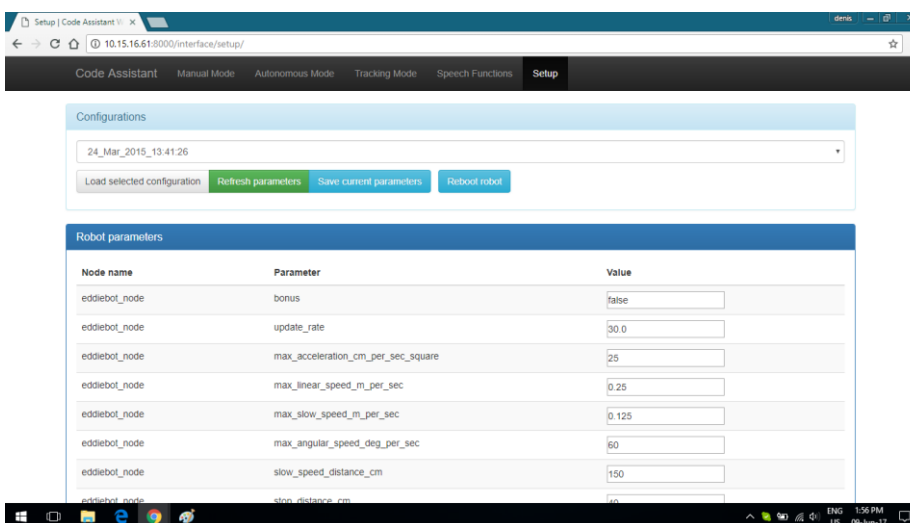
Фигура 3.21. Меню за полу-автономно управление.

3.3.3. Меню за управление на гласовите команди на робота.



Фигура 3.22. Меню за гласови команди.

3.3.4. Меню за настройки на системите и функциите на робота.



Фигура 3.23. Меню за настройки на робота.

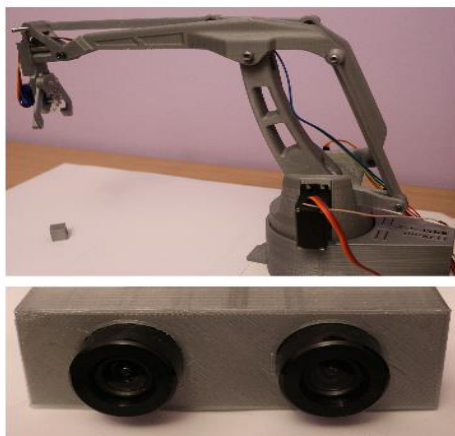
3.4. Управление на роботизирани манипулатори.

Транспортирането на обекти е сложен процес, който включва множество видове роботизирани системи. Този процес може да се раздели на следните стъпки:

Стъпка 1: разпознаване на определен обект и придвижване на мобилната платформа възможно най-близо до него, така че обекта да застане в зоната на достижимост на ръката на робота. Освен това, докато се движите по картата, роботът трябва да разпознава обектите, които ги интересуват, и да ги постави на картата за последваща употреба [68].

Стъпка 2: точно позициониране на хващача на ръката до определена точка от обекта. Това е много предизвикателна задача. Някои изследвания се фокусират върху много сложни системи изградени от няколко камери, за да постигнат висока надеждност при разпознаване на обект от различни гледни точки, при процеси на сглобяване [69]. След като се разпознае обекта, трябва да се определи неговата позиция [70]. Тогава ако се знаят кинематиката и динамиката на ръката е сравнително лесно да се премести хващача до определената позиция за захващане.

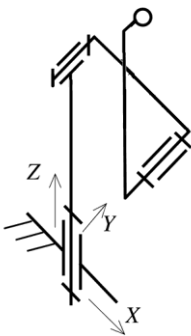
Стъпка 3: Последната стъпка е хващане на обекта. За да бъде стабилен този процес трябва да се използват различни видове сензори: машинно зрение, измерване на разстояние, тактилни [71]. След като обектът се улови, мобилният робот може да го достави.



Фигура 3.24. Антропоморфен манипулатор и стерео камера.

3.4.1. Алгоритъм за прецизно позициониране на манипулатора.

Разглеждаме антропоморфен манипулатор с поне три степени на свобода, както е показано на кинематичната схема на фигура 3.14.

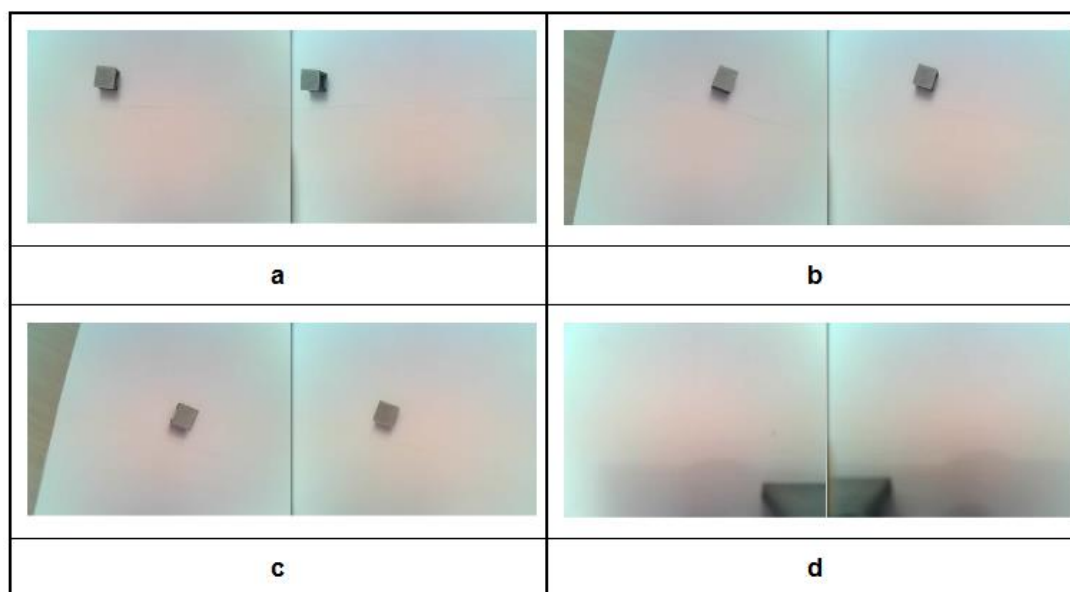


Фигура 3.25. Кинематична схема на манипулатора.

3.4.2. Алгоритъм.

- (S1) Отвеждане на роботизираната ръка в изходна позиция.
- (S2) Разпознаване на обекта и на лявата и на дясната камери на стерео камерата (виж фигура 3.15.a.).
- (S3) Завъртане около оста Z, докато разстоянията между центровете и разстоянията от обекта до хващача и на двете камери на стерео камерата не се изравнят (виж фигура 3.15.b.).
- (S4) преместване на хващача напред, докато обекта не застане в центъра на вертикалата (виж фигура 3.15.c.).
- (S5) Започване на спускане на хващача право надолу, перпендикулярно на равнината XY, докато обектът се намира в областта на захващане, както е показано на фигура 3.15.d.
- (S6) затваряне на хващача и доставяне на обекта.

Обяснение на алгоритъма: След стъпка S4 на горния алгоритъм захващащото устройство се поставя непосредствено над обекта, представляващ интерес. По време на стъпка S5 захващащото устройство се спуска и накрая спира до обекта. Ако движението е перпендикулярно на равнината XY, разстоянието от обекта до централната ос постепенно намалява и остава равно в лявото и дясното изображение. Това спускане право надолу е много важно да бъде по права линия. Само по този начин е сигурно, че ще бъде намерена точната точка на захващане и манипулаторът няма да се сблъска в друг обект. Такава траектория може да бъде гарантирана, ако са известни точните кинематични и динамични модели. Ако тези модели са неточни, алгоритъмът може да продължи да функционира, ако се прилага алгоритъм за гранични грешки (BEA) за ILC. ILC е клас от само настройващи се алгоритми, които многократно изпълняват зададени задачи на движенията на роботи, за да сведат до минимум грешките при позициониране или проследяване на траектории [72]. BEA е модификация на ILC, която решава проблема с преходния растеж на ILC [73] и позволява директното му прилагане в контрола на роботизираните манипулатори [74, 75].



Фигура 3.26. Фази на изпълнение на алгоритъма за позициониране.

3.4.3. Приложение на BEA за ILC.

Стъпка S5 от алгоритъма може да се развие и пренапише както следва:

- (S5.1) Запомняне на настоящата позиция и задаването на същата като изходна позиция.
- (S5.2) Отиване до изходната позиция на BEA.
- (S5.3) Започване на движение надолу докато не се отчете неприемлива грешка или не се достигне до желаната позиция.
- (S5.4) Ако обекта е достигнат, тогава продължи към следваща стъпка, иначе изпълни процедурата на BEA и ILC и обнови управляващите сигнали и премини на стъпка S5.2.

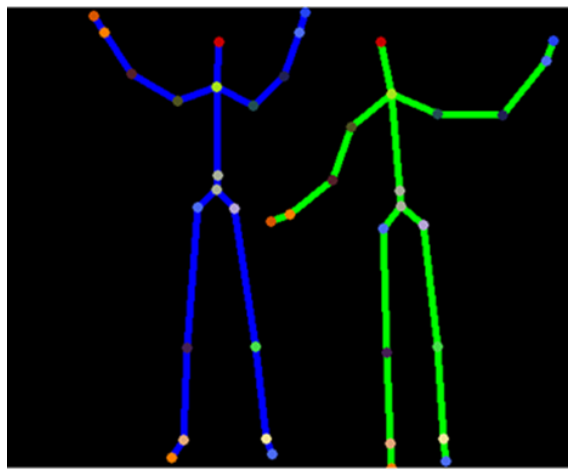
3.4.4. Оптимизация на процеса.

Може да се приложи стандартна техника на моторизация. Времето за изпълнение на алгоритъма може да се намали ако управляващите сигнали след последното опресняване на ILC се свържат с началната позиция и се запамятат. Тогава след изпълнение на стъпка S4 и преди началото на стъпка S5.1, управляващите сигнали свързани с най-близката до текущата стартова позиция се използват като начални.

3.5. Интелигентни системи за сервизни роботи

3.5.1. Алгоритъм за следене на човек и разпознаване на падане.

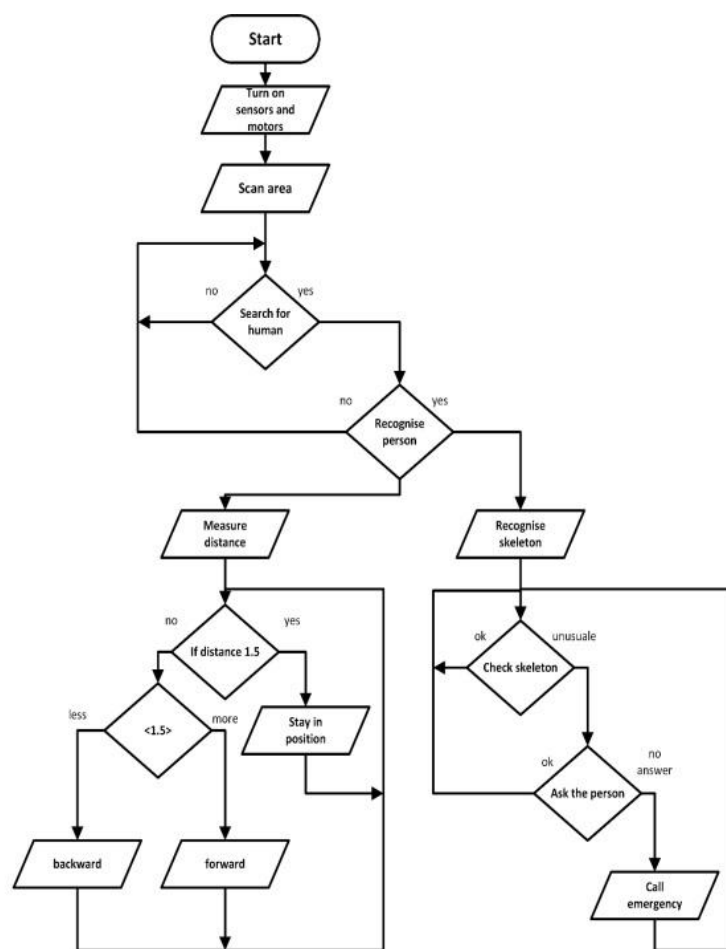
Предложеният алгоритъм в тази точка обединява много от разгледаните системи и проблеми до сега в дисертационния труд. За да може да се приложи робота трябва да разполага с система за автономна навигация, система за машинно зрение, система за гласово разпознаване и възможност за комуникиране с човека.



Фигура 3.27. Създаване на модел на скелета на човек.

Основна роля в изпълнението на този алгоритъм е на системата за компютърно зрение. Чрез нея и свойствата на сензора Kinect можем да разпознаваме определен човек. Също така този сензор има свойството да прави кинематичен модел на човека - (фигура 3.16).

Алгоритъмът за следене и разпознаване на падане - (фиг. 3.17):



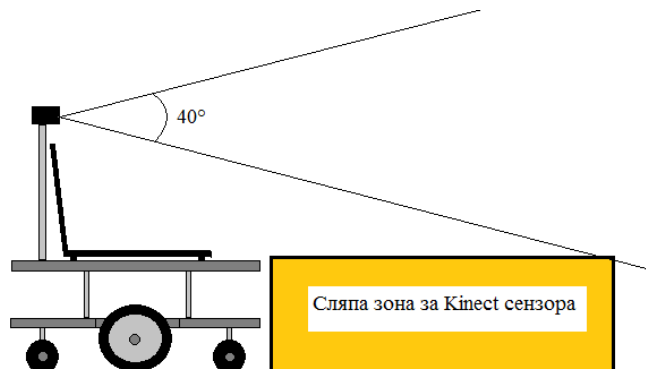
Фигура 3.28. Алгоритъм за следене и разпознаване на падане.

- Робота инициализира средата, в която се намира и започва да търси своя собственик. Търсенето на собственик се разделя на два етапа: 1 – намиране на хора, 2 – разпознаване на собственика.
- Следва процедура по изпълнение на съществените функции на алгоритъма: 1 следене на човека, 2 – разпознаване на падане.
- Следенето е базирано на свойството на дълбочинния сензор на Кинекта да измерва разстоянието до обекти.
- Разпознаването на падане е базирано на следене на точките на скелета и по конкретно тези над кръста на човека. Понеже не може да сме сигурни дали това е така се активират процедура за изясняване на ситуацията, робота пита човека дали е паднал:
 - ако няма отговор до определено време робота автоматично се свързва с спешна помощ и близките на човека;
 - ако има отговор и той е положителен тогава робота отново се свързва с спешна помощ и близките на човека;
 - ако отговора отрицателен робота изчаква човека да стане до определено време.
 - Ако човека не стане робота повтаря въпросите отново, докато човека не се изправи.

3.5.2. Сензорна система за избягване на препятствия.

- Описание на проблема.

Проблема се състои в това, че пред самия робот се получава зона, в която Kinect сензора не може да вижда (Фиг. 3.18). Зоната, която трябва да се обхване е с формата на паралелепипед с размери височина 25 см, дължина 60 см и ширина 45 см. Така характеризирани тази зона изисква сензори с максимален обхват на измерване поне 70 см, за да се постигне точност при измерванията.



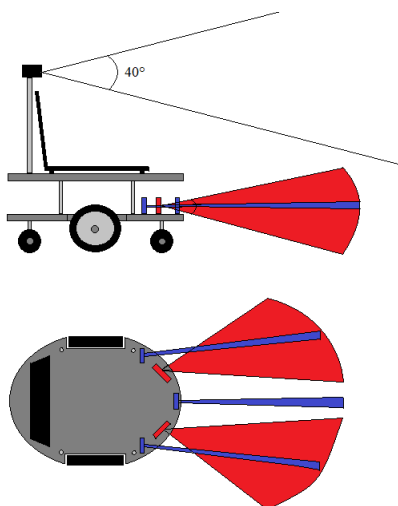
Фигура 3.29. Проблемна зона.

- **Решение на проблема.**

Решението на проблема се състои от два етапа.

а) първи етап – добавяне на допълнителни сензори (Фиг. 3.19).

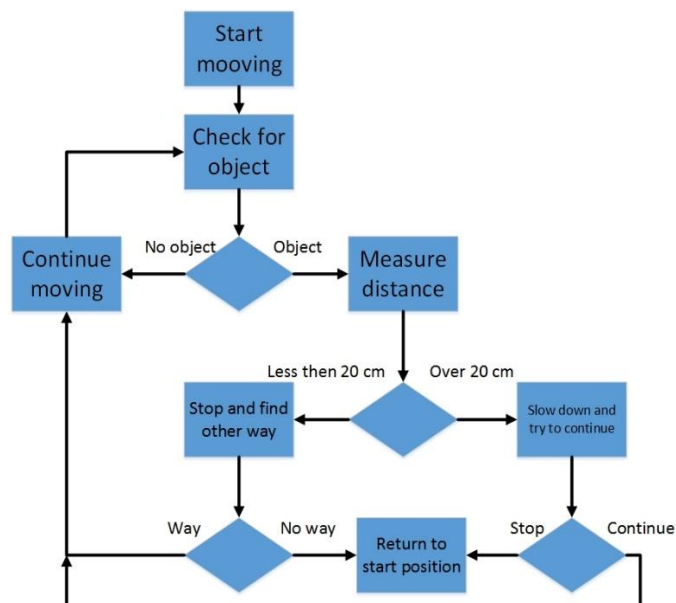
Допълнителните сензори са монтирани в предната част на долната платформа на робота. Те се намират на височина около 15 см, което се предполага, че ще бъде достатъчно за да се регистрират малки обекти, намиращи се непосредствено пред робота. Максималният обхват напред на всички сензори е ограничен до 70 см, понеже след това се припокриват с Kinect сензора.



Фигура 3.30. Обхват на сензорите.

б) втори етап – създаване на алгоритъм за съвместна работа на Kinect с допълнителните сензори.

Алгоритъма за осигуряване на съвместна работа на всички сензори е добавен към алгоритъма за навигационната система - фигура 3.20.



Фигура 3.31. Алгоритъм за следене на препятствия.

Измерванията на допълнителните сензори са в началото на алгоритъма, след което се продължава с нормалната работа на алгоритъма на навигацията. Същността на този алгоритъм е при наличие на обект в обхвата на допълнителните сензори да се предвиди спиране или заобикаляне.

ГЛАВА 4. Експерименти и получени резултати.

Всички методи и изследвания са приложени върху робота, след което са проведени експерименти, които да покажат получените резултати и цялостната функционалност на робота. Проведени са експерименти по отделно за всяка система, след което са проведени експерименти върху съвместната работа на всички системи на робота. Подробности за експериментите и резултатите са описани в следващите точки.

4.1. Експерименти и резултати от навигацията и локализацията

Проведени са следните експерименти на закрито, след като е направена предварително карта на помещението:

- Намиране на точната локация на робота;
- Придвижване на робота до зададени координати с точност до 10см;
- Отчитане на времето необходимо за достигане на целта, преди подобряване на системата и след това;
- След поставяне на допълнителни обекти в помещението, навигацията трябва да намери алтернативен път и да заобиколи обектите.

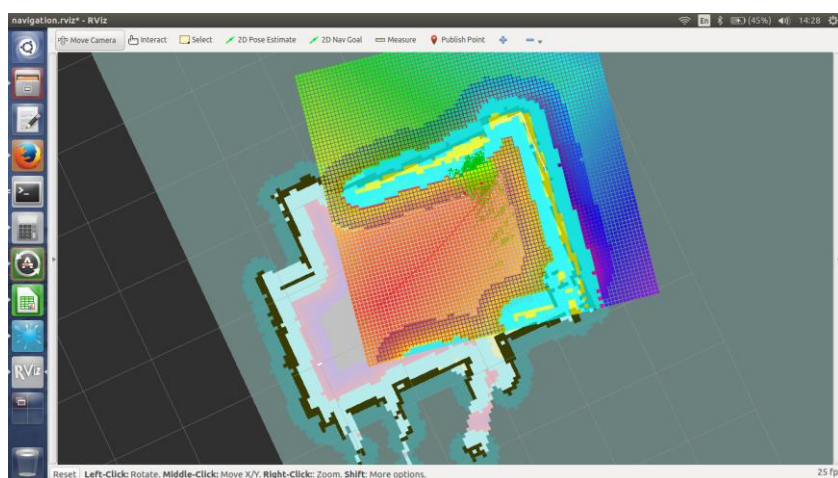
На фигура 4.1 в цветното поле е показана картата на помещението, местоположението на робота и изчислената траектория до другия ъгъл на стаята. Това цветно поле представлява обхвата на дълбочинния сензор на Кинект-а. оцветените в жълто стени се използват като маркери за ориентиране. И понеже няма препятствия траекторията на движение е права.

На фигура 4.2 отново е показана същата карта, но този път са поставени допълнителните обекти. В този случай те са разположени, така че робота да може да премине между тях.

Резултати от експеримент 1. След като е избрана картата на помещението, в което се намира робота навигационната система успешно локализира позицията и ориентацията на робота, на база характерните маркери на помещението.

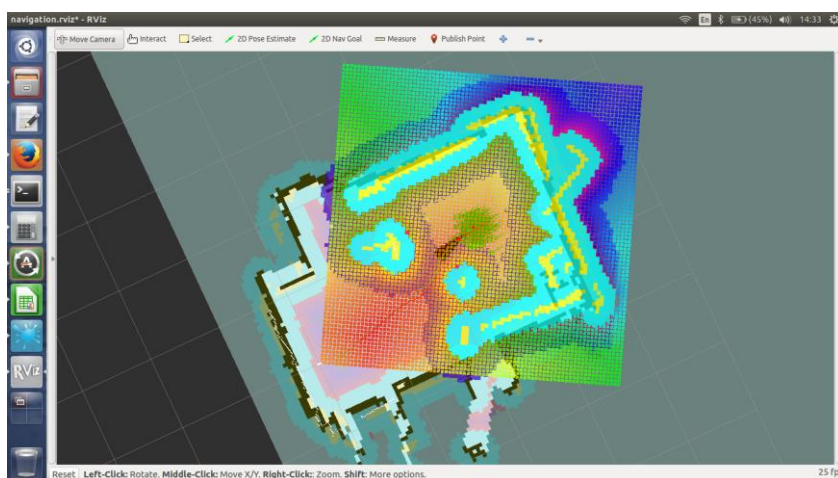
Резултати от експеримент 2. На робота му бе зададена точка, до която да достигне. Навигационната система успешно контролира робота до достигане на целта, като отклонението между центъра на мобилната платформа на робота и зададената точка е около 15см.

Резултати от експеримент 3. Базовата разработка на навигацията за тази мобилна платформа, работи значително по бавно и има повече отклонения, по време на движение. След въвеждане на допълнителните алгоритми и методи робота следва изчислената траектория на движение с малки отклонения и достига до целта за много по-кратко време. За изпълнение на еднакви траектории с дължина шест метра времената са следните: базова навигация – 25 секунди, подобрена навигация – 16 секунди.



Фигура 0.32. Локализиране на робота и задаване на дестинация.

Резултати от експеримент 4. След поставяне на допълнителни предмети в помещението, навигацията успешно регистрира тяхното местоположение върху картата на помещението, без предварително сканиране. При задаване на дестинация, алгоритъма изчислява траектория, която преминава или между новите обекти или ги заобикаля по най-краткия път. След което робота успешно достига до зададената цел.



Фигура 0.33. Разпознаване на допълнително поставените обекти и изчисляване на нов път.

4.2. Експерименти и резултати след коригиране на грешката на одометрията

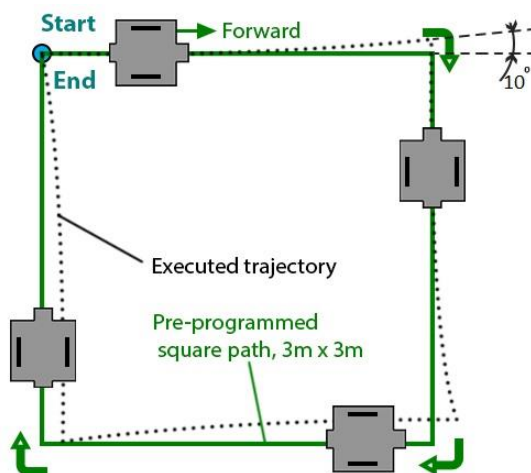
Грешката на одометрията е от голямо значение за правилната работа на навигацията. Поради това проведохме редица експерименти, с които да изчислим точно каква е грешката, след което въведохме допълнителни сензори и методи за намаляване на тази грешка. Проведени са следните експерименти:

- Измерване на одометричната грешка чрез метода на изпълнение на квадратна траектория;
- Измерване на одометричната грешка след въвеждане на коригирано управление на моторите, което компенсира грешката;
- Измерване на одометричната грешка след добавяне на IMU сензор.

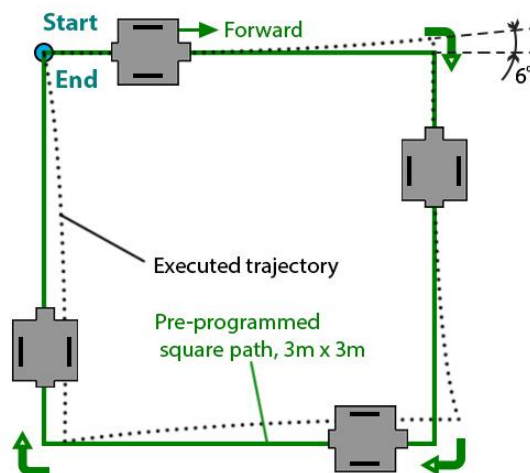
Всички изследвания са проведени като се използва принципа на дву-посочното изпълнение на квадратна траектория. Робота се програмира предварително да изпълнява такава траектория, без да се използват никакви допълнителни сензори освен енкодерите на моторите.

Резултати от експеримент 1. Одометричната грешка бе измерена след няколко обиколки на робота по зададената траектория. Макар да изглежда, че всичко по робота е пропорционално и добре направени тази грешка се оказва доста голяма. Отклонението на робота от траекторията е значително: отклонението на робота при ъглите е около 10 градуса, при движение в квадрат с дължина 3м.

Резултати от експеримент 2. При този експеримент повтаряме предходния, като измерената одометрична грешка е добавена в програмата за управление и по този начин компенсираме систематичната грешка. Получените резултати показват отклонение от 6 градуса. Това е значително по-малка грешка, но като се има предвид, че робота изминава само 3 метра до следващ завой, при изминаване на повече разстояние това отклонение отново не достатъчно.

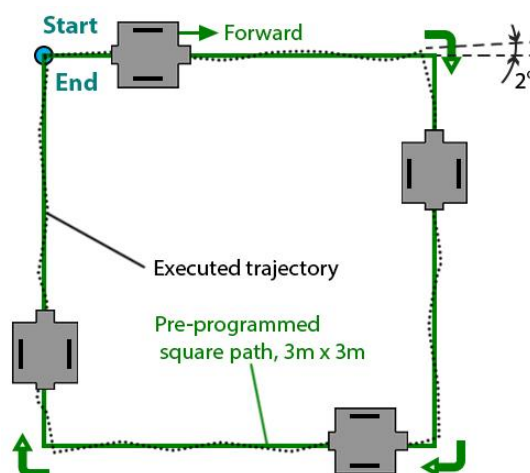


Фигура 0.34. Одометрична грешка.



Фигура 0.35. Одометрична грешка след корекция на систематичната грешката.

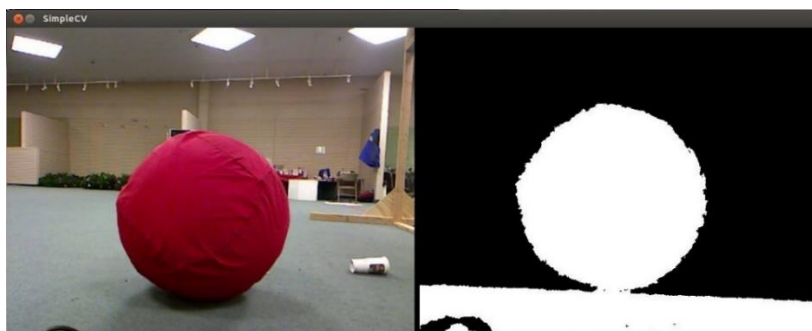
Резултати от експеримент 3. За провеждане на този експеримент добавихме сензор за измерване на ускорението и ориентацията. Сензорът отчита данните в реално време и ги изпраща към системата. На база получените данни от сензора алгоритъмът изчислява моментната грешка и изпраща управляващи за коригиране на позицията. Робота следва зададената траектория като този път се движи с много малки отклонения от пътя. Отчетената максимална грешка е 2 градуса, но понеже робота коригира позицията си в реално време, на практика грешката се свежда почти до 0.



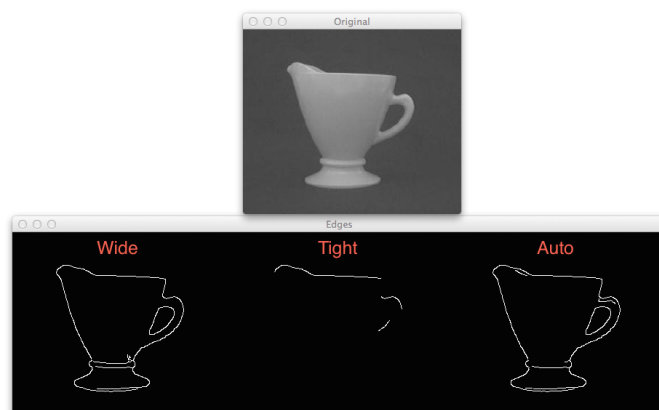
Фигура 0.36. Оптимизиране на движението на робота с IMU сензор.

4.3. Експерименти и резултати на системата за компютърно зрение

В резултат се получават някои фигури на детекцията на ръба (фиг.4.7), разпознаване на цветовете (фигура 4.6) и разпознаване на формата и цветовете (фиг.4.8). На фигура 4.7 са показани три вида детекция на ръба. На фигура 4.6 разпознаваме червения цвят и изображението на резултата показва само този цвят. На фигура 4.8 е показано разпознаването на формите и цветовете на обектите. В зависимост от интензивността на светлината можем да коригираме ръчно или автоматично нивата на светлините на камерата.



Фигура 0.37. Разпознаване на червен цвят.



Фигура 0.38. Разпознаване на предмет според детекция на ръбовете.



Фигура 0.39. Разпознаване на форма и цвят.

4.4. Експерименти и резултати при изпълнение на алгоритмите за подпомагане на възрастни и инвалиди

Тук е изпробвана работата на алгоритъма за следене на човека и разпознаване на падане. Резултати от действието на следене са задоволителни. Робота успява да следва човека на разстояние между 1.5 – 2.0 метра. Благодарение на направените до момента подобрения движението на робота е плавно и той не застрашава човека.

Тестовите на разпознаването на падане са проведени при следните условия: в помещение на закрито, даден човек застава пред робота на разстояние между 1.5 – 2.5 метра. В случаен момент човека рязко се навежда до земята, като по този начин имитира падане. От проведени 30 опита с 3-ма човека с различен ръст, системата успешно разпознава около 83% от паданията.

4.5.Експерименти и резултати при процес на хващане на обекти

Първата група тестове беше без използване на ILC и оптимизации. Скоростта на снижаване е много бавна и с много корекции на траекторията. Ръката бе способна да хване предмети, но поради неточни модели скоростта на снижаване трябваше да бъде много бавна и траекторията имаше много корекции.

Втората група тестове беше с използването на алгоритъма за ILC. Изпробвани са повече от 100 произволно генерирани точки на захващане. Всички те бяха в работната зона на робота. За да завърши изпълнението на стъпка 5 на алгоритъма, ILC имаше нужда от около 11 повторения. Въпреки това, общото време за изпълнение е близко до това на първата група от тестове и понякога е дори по-бавно.

Третата група се възползва от оптимизацията. След тренировка със същата серия от 100 точки за захващане, броят на повторенията беше намален до 5. При следващите случайно избрани точки времето за изпълнение е по-бързо от общото време от първата група тестове.

4.6.Експерименти и резултати на работата на различните устройства и интерфейси за управление на сервизния робот

Експерименти и резултати при управление с джойстик: роботът реагира на момента. Голямата резолюция на бутоните позволява промяна в скоростта с много малки степени. Бутона за аварийно спиране работи надеждно.

Експерименти и резултати от управлението на робота посредством веб базирания интерфейс. Всяко меню предоставя добри възможности за управление. Единствено има забавяне на видео сигнала от камерата на робота в зависимост от устройството, което се използва за управление. Към робота могат да се свържат едновременно поне 10 устройства, които могат независимо да подават команди.

ЗАКЛЮЧЕНИЕ

В дисертационния труд са разгледани някои от основните проблеми при сервизните мобилни роботи предназначени за подпомагане на човека. Засегнатите проблемни области са следните: локализация, навигация и картографиране, операционни системи за сервизни роботи, компютърно зрение за разпознаване на обекти и образи, управление на антропоморфен манипулатор за пренасяне на обекти, методи, устройства и интерфейси за управление на сервизните роботи.

В процеса на работа са проведени изследване, подобряване и създаване на методи, алгоритми и системи за управление на сервизни роботи, предназначени за подпомагане на човека. Провеждането на тези изследвания допринася за развитието на сервизните

роботи, като подобрява и разширява техните възможности за изпълнение на различни по сложност задачи. Това показва, че целта на дисертацията е изпълнена.

Разработени са методи, техники и алгоритми за подобряване и реализиране на сигурност и надеждност в системите за управление на сервизните роботи. След повеждане на експерименти, постигнатите резултати показват добри постижения в разгледаните проблемни области. Подобрена точност и скорост на движение при навигацията, коригиране на одометричната грешка в реално време, разпознаване на лица и предмети по характерни признаци, разработен е иновативен метод за управление на манипулатор, решени са проблеми свързани с подпомагане на възрастни хора и инвалиди и са представени различни методи и интерфейси за управление на роботите. Тези резултати удовлетворяват решението на дадените задачи за изпълнение.

За в бъдеще ще се продължат изследванията върху системите за управление на сервизните роботи за подпомагане на човека, като се изследват и разработят нови алгоритми за изпълнение на сложни движения и задачи, които роботите ще трябва да изпълняват. Също така ще се изследват проблеми с управлението на робите при работа на открито. Подобряването на сервизните роботи и намаляването на тяхната цена, чрез множество научни изследвания ще допринесе за скорошното разпространение на тези роботи в домовете на хората.

Приноси в дисертационния труд

С оглед на работата извършена в дисертацията и резултатите, получени в хода на изследванията и изложени по-горе, могат да бъдат формулирани следните научно-приложни приноси:

1. Анализирани са състоянието и развитието на научните изследвания и проблемите в областта на сервизната роботика. Изследвани са методи за подобряване на локализация и навигация при мобилните роботи, като алгоритми за работа в реално време, коригиране на систематичната и несистематичната грешки и подобряване управлението на моторите;
2. Реализирани са и изследвани методи и алгоритми за управление на камерата на робота за разпознаване на предмети и хора с помощта на специализираните библиотеки OpenCV;
3. Разработени са приложения на Операционната система за роботи – ROS върху робота. Описани са основните функции и принципа на работа на операционната система. Реализирано е многоканално управление на робота, чрез различни устройства;
4. Разработен е метод за итеративно управление на манипулатор, чрез използване на компютърно зрение за обратна връзка и методи за оптимизиране на управлението чрез меморизация и коригиране на грешките;
5. Разработени са методи и техники за подпомагане на възрастни хора и инвалиди като следене на човека и разпознаване на падане. Също така е представена сензорна система за разпознаване на динамични обекти, преминаващи пред робота;
6. Направен е сравнителен анализ на работата на подобрените системи, спрямо първоначалните възможности на системите за управление на сервизните роботи;

Благодарности

Благодаря на своя научен ръководител проф. д-р Димитър Карастоянов, за ползотворната съвместна работа, за всички съвети и градивните критики, които отправи.

Благодаря на доц. д-р Найден Шиваров и доц. Недко Шиваров от Европейски Политехнически Университет за методологичната помощ, съветите и препоръките, без които настоящата дисертация не би се случила.

Списък на използвана литература

1. Международна федерация по роботика - <https://ifr.org/service-robots/>
2. Международна организация за стандартизация - <https://www.iso.org/standard/55890.html>
3. ISO 8373 - <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>
4. Класификация на сервизните роботи - https://ifr.org/img/office/Service_Robots_2016_Chapter_1_2.pdf
5. Статистически данни за сервизните роботи - https://ifr.org/downloads/press/02_2016/Executive_Summary_Service_Robots_2016.pdf
6. Статистическите данни на ООН <https://esa.un.org/unpd/popdev/Profiles/ageing2015/index.html>
7. M. Hans ; B. Graf ; R.D. Schraft - Robotic home assistant Care-O-bot: past-present-future, Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on, 27-27 Sept. 2002, Berlin, Germany
8. Schraft, R. D.; Graf, B.; Traub, A.; John, D.: "A Mobile Robot Platform for Assistance and Entertainment". In Industrial Robot Journal, Vol. 28, 2001, pp. 29-34.
9. Роботът Pepper - <https://www.ald.softbankrobotics.com/en/cool-robots/pepper>
10. Робот за операции „Да Винчи“ - <http://www.davincisurgery.com/>
11. R. RIENER, "Control of robots for rehabilitation," EUROCON 2005-The International Conference on Computer as a Tool, Serbia & Montenegro, Belgrade, vol.I, pp. 33-36, 2005
12. H. I. Krebs, B. T. Volpe, M. L. Aisen, and N. Hogan, "Increasing productivity and quality of care: Robot-aided neuro-rehabilitation," Journal of Rehabilitation Research and Development, vol.37, pp. 639-652, 2000.
13. Робот за рехабилитация на Toyota: Welwalk WW-1000 - <https://www.forbes.com/sites/bertelschmitt/2017/04/12/toyota-sends-robots-to-the-hospital/#1adc2841702e>
14. PR2 - <http://www.willowgarage.com/pages/pr2/overview>
15. Care-O-bot 4 - <http://www.care-o-bot-4.de/>
16. Asimo - <http://asimo.honda.com/>
17. Цени на някои сервизни роботи - <https://www.smashingrobotics.com/thirteen-advanced-humanoid-robots-for-sale-today/>
18. Проект KSERA - www.ksera-project.eu
19. Проект CompanionAble – www.companionable.net
20. Проект FLORENCE - www.florence-project.eu
21. Проект SRS - www.srs-project.eu
22. Проект MobiServ - www.mobiserv.info
23. Проект ACCOMPANY - <http://accompanyproject.eu>
24. Проект DALi- www.ict-dali.eu
25. Проект Robot-Era - <http://www.robot-era.eu/robotera/>
26. Проект DOME0 - www.aal-domeo.eu
27. Проект ALIAS - www.aal-alias.eu
28. Проект Silver - <http://www.silverpcp.eu/>
29. Bräunl, Thomas (2008). *EMBEDDED ROBOTICS- Mobile Robot Design and Applications with Embedded Systems*, Springer-Verlag, Berlin Heidelberg.
30. Bräunl, Thomas (2008). *EMBEDDED ROBOTICS- Mobile Robot Design and Applications with Embedded Systems*, Springer-Verlag, Berlin Heidelberg.

31. Borenstein, Johann and Liqiang Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots," in *Transactions on Robotics and Automation*, 1996.
32. E. B. Mathew, D. Khanduja, B. Sapra, B. Bhushan, „Robotic arm control through human arm movement detection using potentiometers“, Recent Developments in Control, Automation and Power Engineering (RDCAPE), 2015 International Conference on, 12-13 March 2015
33. Сензор Kinect, https://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges. Accessed 10 04 2017
34. Shivarov, N.; Educational robots and flexible manufacturing systems for use in training; United Nations, Economic Commission for Europe; Symposium on Management Training Programs and Methods: Implications of New Technologies. Geneva, 17-19 November 1987
35. Операционна система ROS - <http://www.ros.org/wiki/>
36. П. Нортън, Мрежи, SAMS 1999
37. Х. Остерло, TCP/IP – Пълно ръководство, SAMS 2002
38. М. Лутз, Д. Ашер, Да научим Python, O'Reilly 2006
39. J. Roberge, S. Brandle, D. Whittington, A Laboratory Course in C++ Data Structures, Jones and Bartlett Publishers, Sudbury, Massachusetts, 2003
40. Seder, M.; Macek, K.; Petrovic, I. (2005). An integrated approach to real time mobile robot control in partially known indoor environments. Proceeding of the 31st Annual Conference of the IEEE Industrial, Electronics Society, pp.1785.
41. Khatib, O. (1986). Real time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research, Vol.5, Issue 1, Spring 1986, pp. 90-98, ISSN 0278-3649.
42. Borenstein, J. & Koren, Y. (1990). Real-time obstacle avoidance for fast mobile robots in cluttered environments, proceeding of the IEEE International Conference on Robotics and Automation, Vol. 1, pp. 572-577, ISBN 0-8186-9061-5, Cincinnati, 13-18 May 1990, OH, USA.
43. Borenstein, J. & Koren, Y. (1991). The vector field histogram – fast obstacle avoidance for mobile robots, IEEE Transactions on Robotics and Automation, Vol. 7, Issue 3, June 1991, pp. 278-288, ISSN 1042-296X.
44. Minguez, J. & Montano, L.(2000). Nearness diagram navigation (ND): A new real time collision avoidance approach, proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'00, pp.2094-2101, 2000, Takamatsu, Japan.
45. Simmons, R. (1996). The curvature velocity method for local obstacle avoidance, Proceeding of the IEEE International Conference on Robotics and Automation, ICRA'96, pp. 3375–3382, ISBN 0-7803- 2988-0, Minneapolis MN, 22-28 April 1996, USA.
46. Ko, N.Y. & Simmons, R. (1998). The lane curvature method for local obstacle avoidance, Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'98, pp. 1615-1621, Victoria, October 1998, Canada.
47. Fox, D.; Burgard, W. & Thrun, S. (1997). The dynamic window approach to collision avoidance, IEEE Robotics & Automation Magazine, Vol. 4, Issue 1, Mar 1997, pp. 23-33, ISSN 1070-9932.
48. Riisgaard, S. & Rufus, M. B. (2007).SLAM for Dummies, A Tutorial Approach to Simultaneous Localization and mapping.
49. O’Kane, Jason M., (2013). A Gentle Introduction to ROS, University of South Carolina, Columbia.

50. Foote, Tully, "tf: The transform library", Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on 2013, ISSN - 2325-0526.
51. ROS, "gmapping," ROS.org, [Online]. Available: <http://wiki.ros.org/gmapping>.
52. ROS, "Setup and Configuration of the Navigation Stack on a Robot," ROS.org, [Online]. Available: <http://wiki.ros.org/navigation/Tutorials/RobotSetup>.
53. Borenstein, Johann and Liqiang Feng, "Correction of Systematic Odometry Errors in Mobile Robots," in IROS '95, Pittsburgh, 1995.
54. J. a. F. L. Borenstein, "UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots," in *SPIE Conference on Mobile Robots*, Philadelphia, 1995.
55. Вградено измервателно устройство - http://en.wikipedia.org/wiki/Inertial_measurement_unit
56. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1. IEEE, 2001, pp. 1-511.
57. Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.
58. Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004 (2004), 469–481.
59. <http://web.archive.org/web/20040829092603/http://www.dcs.qmul.ac.uk:80/research/vision/publications/papers/bmvc97/node1.html>
60. https://en.wikipedia.org/wiki/David_H._Hubel
61. Kanade, T. Picture processing system by computer complex and recognition of human faces. PhD thesis, Kyoto University, November 1973.
62. Brunelli, R., Poggio, T. Face Recognition through Geometrical Features. European Conference on Computer Vision (ECCV) 1992, S. 792–800.
63. Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004 (2004), 469–481.
64. Shkenderov I., **Chivarov, N.**, and V. Galabov. Synthesis of Power-Polynomial Motion Laws; Problems of Engineering Cybernetics and Robotics, Vol. 59, 2008, pp. 74–81, Marin Drinov publishing house, ISSN 0204-3848
65. X-bee модул - <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module>
66. Светлин Наков, Веселин Колев, Въведение в програмирането със C#
67. "Django documentation," Django Software Foundation, [Online]. Available: <https://docs.djangoproject.com/en/1.10/>.
68. S. Ekvall, P. Jensfelt, D. Kragic, "Integrating Active Mobile Robot Object Recognition and SLAM in Natural Environments", Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 5792-5797, 2006.
69. W. Mustafa, N. Pugeault, N. Krüger, "Multi-View Object Recognition using View-Point Invariant Shape Relations and Appearance Information", IEEE International Conference on Robotics and Automation, (ICRA), Karlsruhe, Germany May 6-10, 2013.
70. M. Lutz, D. Stampfer, C. Schlegel, "Probabilistic Object Recognition and Pose Estimation by Fusing Multiple Algorithms", IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany May 6-10, 2013.
71. S. Ye, K. Suzuki, Y. Suzuki, M. Ishikawa, M. Shimojo, "Robust Robotic Grasping Using IR Net-Structure Proximity Sensor to Handle Objects with Unknown Position and Attitude", Proc. IEEE Int. Conf. on Robotics and Automation, pp. 3271-3278, 2013.

72. S. Arimoto, S. Kawamura, F. Miyazaki, "Bettering Operation of dynamic systems by learning: A new control theory for servomechanism of mechatronics systems", Proc. of 23rd Conference on Decision and Control, Las Vegas, NV, pp. 1064-1068, (1984).
73. R.W. Longman, "Iterative learning control and repetitive control for engineering practice", International Journal of Control, Vol. 73, No. 10, pp. 930-954, (2000).
74. K. Delchev; Iterative learning control for robotic manipulators: A bounded-error algorithm; International Journal of Adaptive Control and Signal Processing; 8 December 2014
75. K. Yovchev, K. Delchev, E. Krastev, 2016, June. Computer Simulation of Bounded Error Algorithm for Iterative Learning Control. In International Conference on Robotics in Alpe-Adria Danube Region (pp. 136-143). Springer International Publishing.