

Българска академия на науките
Институт по информационни и комуникационни технологии
Секция: „Вградени интелигентни технологии“

автор: маг. инж. Денис Сафидинов Чикуртев

ДИСЕРТАЦИЯ

за получаване на образователна и научна степен „ДОКТОР“

Тема:

**Изследване и управление на сервизни работи за
подпомагане на човека.**

научна област: 5. Технически науки

професионално
направление: 5.2 Електротехника, Електроника и Автоматика

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

научен ръководител: **проф. д-р Димитър Неделчев Карастоянов**

Благодарности

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

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.

Съдържание

Увод.....	1
ГЛАВА 1. Литературен обзор.....	2
1.1. Дефиниция и класификация на сервизните роботи.	2
1.1.1. Дефиниция за робот и сервизен робот.	2
1.1.2. Други дефиниции свързани със роботи и сервизни роботи.	3
1.1.3. Класификация на сервизните роботи според приложението.	3
1.2. Резюме и прогнози на Международната Федерация по Роботика за развитието и продажбите на сервизни роботи през последните години [5].	6
1.2.1. Резюме за периода 2014-2015 г.	6
1.2.2. Прогнози за професионалните сервизни роботи за периода 2016-2019:	10
1.2.3. Прогнози за непрофесионалните сервизни роботи за периода 2016-2019:.....	11
1.3. Описание на развитието и проблемите при сервизните роботи за подпомагане на човека.	12
1.4. Актуални проекти за изследвания на сервизни роботи за подпомагане на хората.	16
1.5. Мобилност на сервизните роботи.	19
1.5.1. Видове мобилни платформи.	19
1.5.2. Характеристики на мобилните роботи.	26
1.5.3. Локализация и навигация.	28
1.5.4. Одометрия.	29
1.6. Манипулатори при сервизните роботи.	30
1.6.1. Популярни роботизирани манипулатори.	30
1.6.2. Управление.	31
1.7. Управление на сервизните роботи.	31
1.7.1. Работа в реално време.	31
1.7.2. Операционни системи и софтуер за роботи.	33
1.7.3. Изкуствен интелект.	33
1.8. Цел и задачи.	34
ГЛАВА 2. Алгоритми за управление, навигация и локализация и компютърно зрение.	36
2.1. Описание на характеристиките на мобилния робот за подпомагане на човека.	36
2.2. Операционна система ROS.	39
2.2.1. Файлово ниво.	40
2.2.2. Изчислително ниво.	40
2.2.3. Ниво общност.	42
2.3. Навигация и локализация.	42
2.3.1. Конфигурация на данните от сензорите в средата на ROS.	44

2.3.2.	Навигационен пакет в ROS.....	45
2.3.3.	Информация от одомерията.....	46
2.3.4.	Управление на платформата.....	47
2.3.5.	Картографиране - Mapping	48
2.3.6.	Локализация и разпознаване на помещения и сгради.....	49
2.3.7.	Алгоритъм за сканиране при старт.....	49
2.3.7.1.	Сравняване на новата карта с запаметените карти.....	50
2.3.7.2.	Режим на запис на нова карта.	50
2.4.	Коригиране на одомерични грешки.....	50
2.4.1.	Описание на систематичните и несистематичните грешки	51
2.4.2.	Коригиране на систематичните грешки.	52
2.4.3.	Коригиране на несистематичните грешки.	53
2.4.4.	Коригиране на систематичната одомерична грешка чрез сензор за ориентация и ускорение.....	54
2.5.	Компютърно зрение	55
2.5.1.	Цветни пространства в OpenCV.....	55
2.5.2.	Трансформация на Фурие.....	55
2.5.3.	Високочестотен и нискочестотен филтър.....	55
2.5.4.	Намиране на градиент на интензитета на изображението.....	56
2.5.5.	Хистерезис: Последната стъпка. cv2.Canny използва два прага (горен и долен):.....	56
2.5.6.	Съвместимост между ROS и OpenCV	56
2.5.7.	Намиране / разпознаване на обекти.....	57
2.5.8.	Намиране / разпознаване на лица	58
2.5.8.1.	Откриване на лица.....	59
2.5.8.2.	Разпознаване на лица.	59
ГЛАВА 3.	Сервизни работи за специфични цели.....	63
3.1.	Дистанционно управление на сервизни работи за инспекция с термо камера.....	63
3.1.1.	Управление на мобилната платформа.	63
3.1.2.	Заснемане и анализ на изображения с термо камерата.....	68
3.2.	Работи за възрастни и инвалиди, ценово ориентирани решения.....	69
3.2.1.	Управление на задвижването.	69
3.2.2.	Избягване на препятствия.....	70
3.2.3.	Изследване на методите за управление на сервизния робот.	71
3.3.	Потребителски интерфейс за управление на сервизни работи.....	73
3.3.1.	Меню за управление в ръчен режим.....	74
3.3.2.	Меню за управление в полу-автономен режим.	74
3.3.3.	Меню за управление на гласовите команди на робота.	75

3.3.4. Меню за настройки на системите и функциите на работа.	76
3.4. Управление на роботизирани манипулатори.	77
3.4.1. Алгоритъм за прецизно позициониране на манипулатора.	78
3.4.2. Предварителни условия.	78
3.4.3. Алгоритъм.	79
3.4.4. Приложение на ВЕА за ILC.	80
3.4.5. Оптимизация на процеса.	80
3.5. Интелигентни системи за сервизни работи.	80
3.5.1. Алгоритъм за следене на човек и разпознаване на падане.	80
3.5.2. Сензорна система за избягване на препятствия.	83
ГЛАВА 4. Експерименти и получени резултати.	86
4.1. Експерименти и резултати от навигацията и локализацията.	86
4.2. Експерименти и резултати след коригиране на грешката на одометрията.	88
4.3. Експерименти и резултати на системата за компютърно зрение.	90
4.4. Експерименти и резултати при изпълнение на алгоритмите за подпомагане на възрастни и инвалиди.	91
4.5. Експерименти и резултати при процес на хващане на обекти.	91
4.6. Експерименти и резултати на работата на различните устройства и интерфейси за управление на сервизния робот.	92
ЗАКЛЮЧЕНИЕ.	93
НАУЧНО-ПРИЛОЖНИ ПРИНОСИ.	94
БИБЛИОГРАФИЯ.	95

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. Освен това са разгледани и конкретизирани проблеми при сервизни работи предназначени за инспекция, подпомагане на възрастни хора и инвалиди и други, някои от тези проблеми са разпознаване на човешки образи и конкретни предмети от бита, заснемане на термални изображения, интерфейси за управление и комуникация между хората и роботите и алгоритми за разпознаване на движения.

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.

Според определението за робот „степен на автономност“ е изискване към сервизните работи да варира от частична автономност (включително взаимодействие между човек и робот) до пълна автономност (без човешка намеса в работата на робота). Затова в допълнение към напълно автономните сервизни работи, статистически се добавят и системи, които се базират на различна степен на взаимодействие между човек и робот или дори изцяло теле-управляеми работи. В този смисъл взаимодействието човек – робот означава обмен на информация и действия между човек и робот, за да се изпълни определена задача, с помощта на потребителски интерфейс.

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.

1.1.2. Други дефиниции свързани със работи и сервизни работи.

Система за управление – набор от логическо управление и силови функции, които позволяват наблюдение и управление механичната конструкция на робот и комуникация с околната среда.

Манипулатор – машина, в която механизмите са изградени от серия от сегменти, които са съединени и се движат един спрямо друг, като целта е да се хващат и/или да се преместват обекти. Манипулаторите обикновено разполагат с няколко степени на свобода.

Мобилен робот – робот способен да се движи със собствено управление. Мобилен работи може да бъде мобилна платформа, с или без манипулатор.

Оператор – човек, който е правоспособен да стартира, наблюдава и прекратява дадено действие на робот или роботизирана система.

Интелигентен робот – робот способен да изпълнява задачи, като наблюдава и разпознава заобикалящата го среда и/или взаимодействия с външни източници и адаптира поведението си.

1.1.3. Класификация на сервизните работи според приложението.

IFR събира статистически данни от производители на сервизни работи по целия свят за да може да се класифицират сервизните работи по категории и начини на взаимодействие [4]. Тази информация се събира от 2005 година насам. Производителите на сервизни работи редовно и систематични изпращат мейли с отчети, разпределени по области на приложение, както е показано на таблици 1 и 2.

Раздел 1	Тип работи: сервизни работи за лични/домашни нужди
1-6	Работи за домашни задачи
1	Работи асистенти
2	Работи прахосмукачки, чистачи

3	Роботи за косене на трева
4	Роботи за почистване на басейни
5	Роботи за почистване на прозорци
6	Други
7-10	Роботи за забавление и развлечение
7	Роботи играчки
8	Мултимедия/дистанционно присъствие
9	Образователни и изследователски работи
10	Други
11-13	Роботи за подпомагане на възрастни и инвалиди
11	Роботизирани инвалидни колички
12	Лични апарати и помощни устройства за подпомагане
13	Други функции за подпомагане
14	Личен транспорт
15	Охрана на дома и наблюдение
16	Други лични/домашни работи

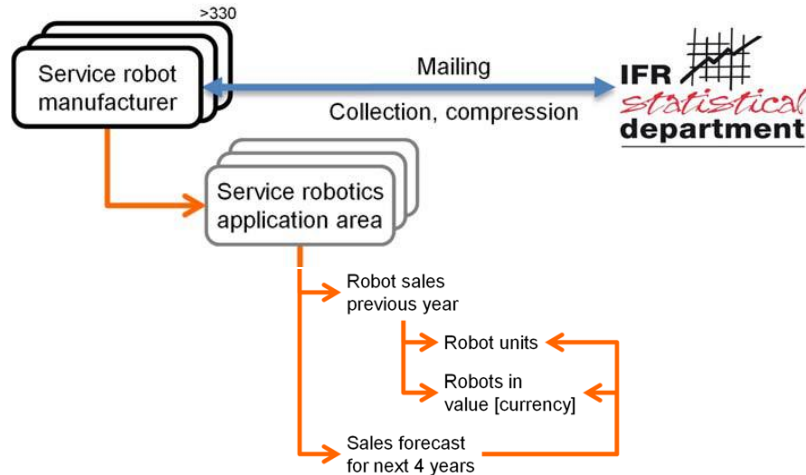
Таблица 1.2. Класификация на сервизните работи според областта на приложение и вида на работите; сервизни работи за лични/домашни нужди.

Раздел 2	Сервизни работи за професионални нужди
17-23	Наземни работи
17	Селскостопански работи
18	Роботи за доене
19	Други работи за животновъдство
20	Горско стопанство и животновъдство
21	Роботи миньори
22	Космически работи
23	Други наземни работи
24-28	Професионално почистване
24	Почистване на под
25	Почистване на стени и прозорци (включително катерещи се работи)
26	Почистване на резервоари, тръби и тръбопроводи
27	Почистване на корпуси (включително на самолети)
28	Други задачи за почистване
29-31	Системи за инспекция и техническа поддръжка
29	Съоръжения, растения
30	Резервоари, тръби и канализации
31	Други системи за инспекция и поддръжка
32-35	Строителство и разрушаване
32	Ядрено разрушаване и разпад
33	Строеж на сгради
34	Роботи за тежката/гражданско строителство
35	Други системи за строителство и разрушаване
36-39	Логистични системи

36	Автоматизирани управляеми превозни средства (АУПС) за производствена среда
37	АУПС за непроизводствена среда, на закрито
38	Натоварване на стока, логистика на открито
39	Други логистични системи
40-43	Медицинска роботика
40	Диагностични системи
41	Роботи помагачи при операции или терапии
42	Рехабилитационни системи
43	Други медицински роботи
44-46	Приложения за сигурност и спасителни операции
44	Роботи за борба с пожари и бедствени ситуации
45	Роботи за сигурност и наблюдение
46	Други роботи за сигурност и спасителни операции
47-51	Роботи за отбрана
47	Роботи за разминирание
48	Безпилотни летателни обекти
49	Безпилотни наземни превозни средства
50	Безпилотни подводни превозни средства
51	Други роботи за отбрана
52	Подводни системи (обща употреба)
53	Екзоскелетони
54	Безпилотни летателни апарати (обща употреба)
55	Мобилни платформи за обща употреба
56-60	Безпилотни наземни превозни средства (обща употреба)
56	Роботи в хотели и ресторанти
57	Мобилни пътеводители, информационни роботи
58	Роботи в маркетинга
59	Роботи екскурзоводи
60	Други (роботи в библиотеки)
61	Други професионални сервизни работи не споменати досега

Таблица 1.2. Класификация на сервизните работи по области на приложение и типа на роботите.

Базата данни на IFR се използва, за да поиска от производителите на сервизни работи да изпратят информация относно продажбите на работи (количество и цена) от предходната година за всяка от категориите и да пресметнат очакваните продажби за следващите четири години.



Фигура 1.1. Процедура по събиране и селектиране на статистическите данни за роботите.

Сервизните роботи получават масово медийно отразяване, което допринася за разпространяването на тази нова област от роботиката до по-широка аудитория. Роботите очевидно са във възход: в производството и все по-често в ежедневието. Нарастващият интерес към сервизната роботика от части се дължи на разнообразието и броя на новите предприятия, които сега възлизат на повече от 25% от всички фирми за роботи. Освен това големите компании инвестират все повече в роботиката, често чрез придобиване на нови дружества.

В професионалните приложения, сервизните роботи вече оказват значително влияние в различни области като селско стопанство, хирургия, логистика и подводни приложения и увеличават икономическото си значение. Водени от повишаващата се заплаха за сигурността, има нарастваща нужда от следене на ежедневните среди, което води до увеличаване и усложняване на управлението на задачи и потоците от данни. За да се задоволят тези големи нужди, роботите ще изиграят много важна роля на пазара за поддръжка, сигурност и спасяване.

Роботиката за персонално и домашно приложение, преживя силен глобален растеж с относително малък брой продукти за масова употреба: роботи за почистване на подове, роботи косачки и роботи за образование и развлечение. Бъдещото развитие на продуктите сочи към домашни роботи с по-висока сложност, възможности и качество, като роботи асистенти за подпомагане на възрастни, за да помагат с домакинската работа и за забавление.

1.2. Резюме и прогнози на Международната Федерация по Роботика за развитието и продажбите на сервизни роботи през последните години [5].

1.2.1. Резюме за периода 2014-2015 г.

Общия брой на професионалните сервизни роботи, продадени през 2015 година се е повишил значително с 25% до 41 060 броя, спрямо продажбите за 2014, които са 32 939. Паричната стойност на продажбите се е увеличила с 14% до 4.6 милиарда долара. От

1998 година до сега са преброени общо около 220 000 сервизни работи за професионална употреба, които са включени към статистиката. Много трудно и невъзможно е да се определи, колко от тези работи все още функционират, поради голямото разнообразие на сервизни работи, което води до различно време на използване. Някои работи (например подводни работи) може да работят повече от 10 години, сравнени със средната продължителност на работа от 12 години на индустриалните работи. Други като роботите за отбрана, могат да служат само за кратък период от време.

През 2015 са инсталирани 19 000 логистични системи, което е с 50% процента повече спрямо 2014 – 12 652, което представлява 46% от общия брой продажби и 17% от общите продажби на професионални сервизни работи. Данните за логистичните системи от 2015 година са събрани и ревизирани благодарение на по-големия обхват на събраната информация. 3410 автоматизирани управляеми превозни средства (АУПС) за производствени среди и 15 515 за непроизводствени среди са произведени, което представлява увеличение с 51% в сравнение с продадените АУПС през 2014. Предполага се, че действителния брой на новите логистични системи пуснати в употреба е много повече. Стойността на продажбите на логистични системи се оценява на около 779 милиона долара, което представлява увеличение с 52% в сравнение с 2014 година.

Сервизните работи с приложение в отбраната отчитат продажби от 11 207 броя, което е 27% от общия брой продадени сервизни работи за професионална употреба през 2015. От тях безпилотните летателни апарати изглежда, че намират най-голямо приложение и са увеличили продажбите си с 4% до 9391 броя. Броят на продадените безпилотните наземни апарати включително работи за разминиране е 1516, което е с 7% по-малко спрямо тези за 2014. Броят на роботите за разминиране през 2014 беше 350, а през 2015 намалява до 300. Стойността на роботите за отбрана може само грубо да бъде пресметната. Тази стойност беше около 1.038 млн. долара, само 1% повече, отколкото през 2014. Този брой остава постоянен и е причина за около 23% от общите продажби на професионални сервизни работи. Въпреки това истинския брой на тези работи, както и стойността им може да бъдат значително по-високи.

Общо 5665 доилни работи са били продадени през 2015 година, в сравнение с 5180 през 2014 година, което представлява 9% увеличение. 160 броя други работи за животновъдство и земеделие, като например мобилни чистачи на обори или роботизирани пазачи за автоматичен контрол на пашата са били продадени през 2015 година, колкото и през 2014г.. Общият брой роботите за обработване на ниви продадени през 2015г. е около 6440, което представлява дял от 16% от общия брой доставки на професионални сервизни работи. Стойността на продажбите на работи за обработване на ниви се е увеличил с 11% до 1.1 млн. долара, което представлява 24% от общата стойност на продажбите на професионални сервизни работи. Селскостопанските работи също стоят стабилно на пазара. Техните продажби са се увеличили от 123 броя през 2014г. до 309 през 2015г. Автоматизацията на земеделието и животновъдството се увеличава.

Продажбите на медицински сервизни работи през 2015 са се увеличили с 7% спрямо тези през 2014г. до 1324 броя, което представлява дял от 3% от общите продажби на

професионални сервизни работи. Най-важните приложения са в хирургията с роботизиран асистент и лечението, с продажби около 1107 броя през 2015г., 12% повече отколкото 2014, докато продажбите на други медицински работи са намалели от 246 до 217. Общата стойност на продажбите на медицински сервизни работи се е увеличила до 1463 млн. долара, което представлява 32% от общата стойност на продажбите на професионални сервизни работи. Медицинските работи са най-скъпите сервизни работи със средна цена за един брой около 1 млн. долара, включително аксесоари и услуги. Ето защо доставчиците на такива работи предоставят възможности за лизингови договори при закупуване. Медицинските работи, както и логистичните системи са добре познати сервизни работи със значителен потенциал за бъдещи растеж и развитие.

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.

Продажбите на моторизираните човешки екзоскелетони са скочили от 273 броя за 2014г. до 370 броя за 2015г. Тези работи успешно се използват за рехабилитация и допълнителна подпора за намаляване на усилияето при вдигане на тежки товари.

През 2015г. са продадени около 5.4 млн. сервизни работи за персонална и домашна употреба, 16% повече спрямо 2014г. Стойността на продажбите се е увеличила с 4% до 2.2 млн. долара.

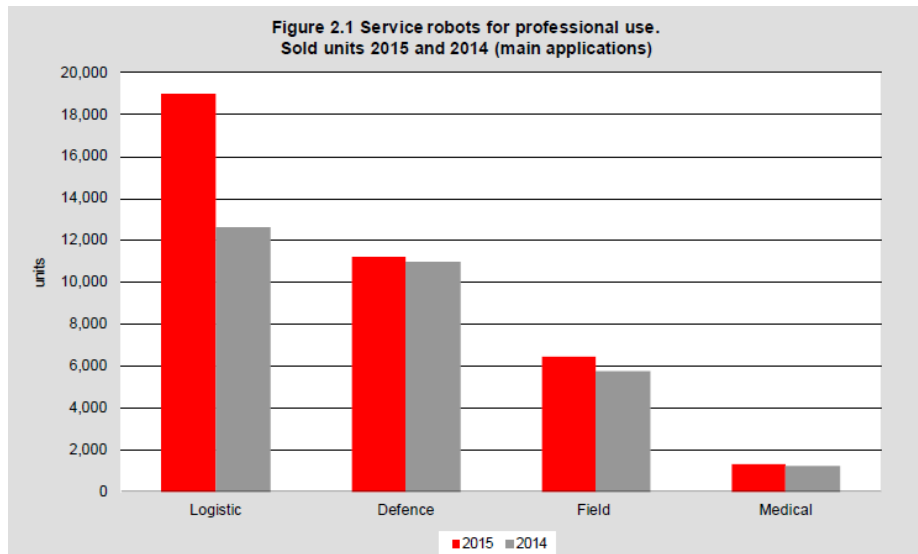
Сервизните работи за персонална и домашна употреба се вписват отделно, тъй като единична стойност е само част от много много видове сервизни работи за професионална употреба. Те също така се произвеждат за масовия пазар с напълно различни ценови и маркетингови схеми.

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.

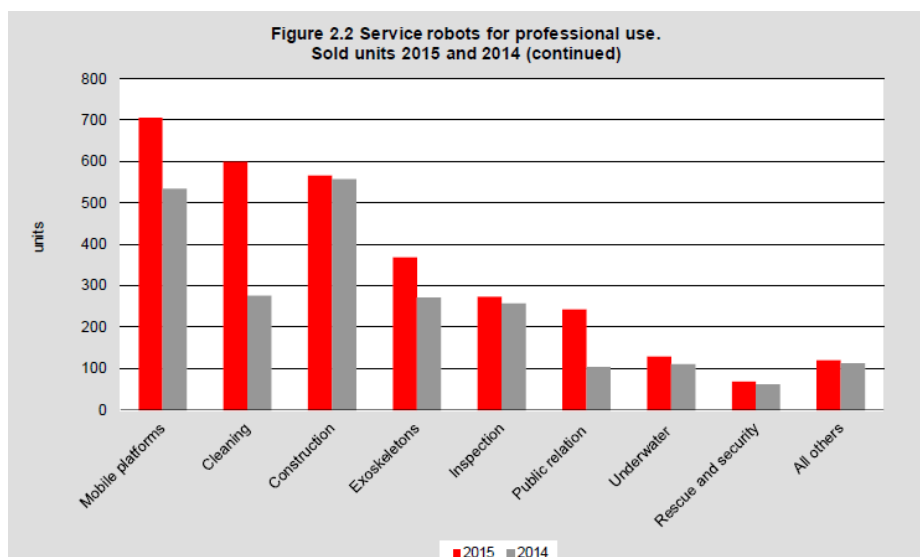
Изчисленията показват, че през 2015г. са продадени повече от 3.7 млн. работи за битови нужди, включително работи за почистване, косене на трева, миене на прозорци и други. Което е с 11% повече отколкото за 2014г. Реалният брой, обаче може да бъде значителни по-голям, тъй като проучването на МФР не може да обхване изцяло

световния пазар на роботи. Стойността на продажбите е около 1.2 милиарда долара. В сравнение с 2014г. те са се покачили с 1%.

Относно роботите за развлечение, около 1.7 млн. броя са отчетени през 2015, 29% повече, отколкото за 2014г. Множество фирми, особено Азиатските предлагат евтини роботи-игралки. Но сред тези масови продукти, има все по-усъвършенствани продукти на пазара за домашно забавление. В продължение на много години програмата на LEGO Mindstorms принадлежи към по-добро качество на продуктите, предлагайки софтуерна среда, която се прилага успешно във високотехнологичната роботика.



Фигура 1.2. Продажби на професионални сервизни роботи според предназначението им.



Фигура 1.3. Продажби на сервизните роботи според предназначението им.

1.2.2. Прогнози за професионалните сервизни работи за периода 2016-2019:

Очаква се около 333 200 нови професионални сервизни работи да бъдат пуснати в действие.

Що се отнася до прогнозите за периода 2016-2019г. прогнозите за продажбите показват увеличение с около 333 200 броя на стойност 23.1 милиарда долара.

Продажбите на логистични системи ще се повишат значително през този период. Повече от 175 000 броя са предвидени, от които около 174 650 са автоматизирани управляеми превозни средства. Логистичните системи съставляват 53% от общото количество сервизни работи в настоящия период на предвиждане.

За този период се предвижда да се продадат 74 800 работи за отбрана. Те са следвани от роботите за обработване на земи – около 34 600 броя. Това по-скоро е консервативно предположение. Тези две групи от сервизни работи формират други 33% от общото количество сервизни работи в настоящия период на предвиждане.

Друг силно нарастващ сектор се предвижда да бъде професионалните работи за почистване. Около 11 700 такива ще бъдат продадени между 2016 и 2019 година, главно работи за почистване на подове.

Очаква се продължително увеличаване на медицинските работи. Очаква се да бъдат продадени около 8150 броя. Секторът за мобилни платформи за масова употреба ще нараства плавно. Доставчиците на сервизни работи очакват продажби на около 7500 мобилни платформи, които ще бъдат предназначени за множество различни приложения. Екзоскелетоните ще се утвърдят на пазара, като се очакват продажби на около 6600 броя.

Силно развиващ се сектор е роботите за обществена комуникация, който ще се увеличи до над 6500 работи. Тези работи се използват все по-често в супермаркети, изложби, музеи и други, като водачи или информатори. Значително по-голям брой работи за инспекция и поддръжка ще бъдат необходими за периода 2016-2019, около 3600.

Друга силно нарастваща приложна група е за строителство и разрушение. Очаква се да се продадат около 2800 работи от този тип. Това също е по-скоро консервативна прогноза. Роботите за строителство и разрушение се използват все по-често на места, където е опасно и нездравословно за хората.

Продажбите на работи за сигурност и спасителни операции са очаква да бъдат около 700, главно работи за наблюдение и сигурност, както и 700 подводни работи.

Тези прогнози, както бе споменато по-рано са базирани на индивидуални прогнози за продажби на фирми и професионални организации. Според статистическия отдел на МФР на тези прогнози трябва да се гледа като на тенденции по отношение на движенията на пазара, а не като на реални и точни прогнози за продажбите.

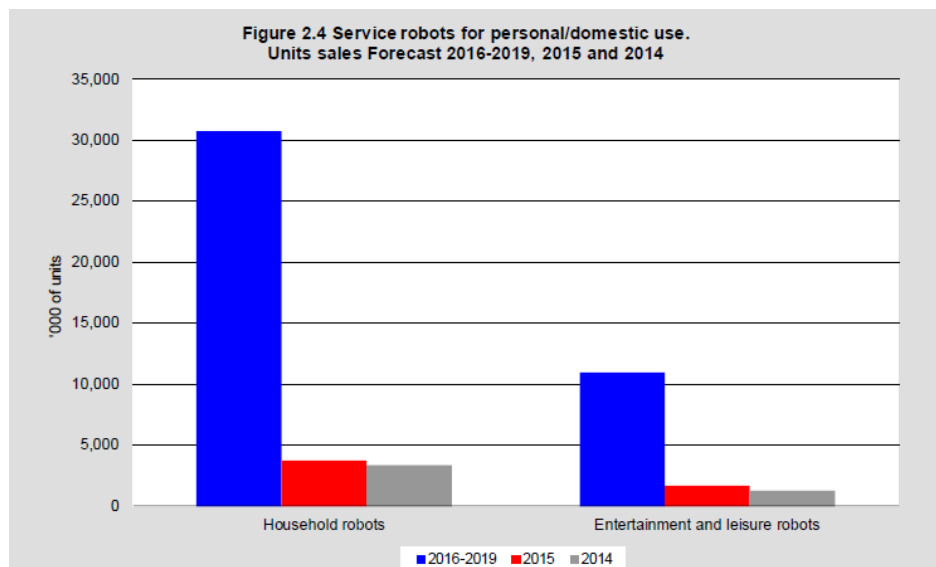
1.2.3. Прогнози за непрофесионалните сервизни работи за периода 2016-2019:

Очаква се да бъдат продадени около 42 милиона сервизни работи за персонално и домашно ползване.

Роботите за почистване на подове ще навлизат все повече в домовете на хората по света. Предвидено е че в периода 2016-2019 ще бъдат продадени около 30 млн. робота за почистване. Що се отнася до роботите за косене на трева се прогнозира продажби от около 910 000 броя.

Доставчиците на сервизни работи вече предвидиха силно увеличение на продажбите на работи придружители/асистенти/хуманоиди през 2010г. Но сега се предвижда, че между 2016 и 2019г. ще се продадат около 8100 броя. Въпреки това, до сега все още няма значителни продажби на хуманоидни работи като придружители за да изпълняват типичните ежедневни задачи в производството, офисите или домовете на хората. Няколко Японски компании (Honda, Kawada, Toyota и други), както и Американски, Корейски и Европейски компании са в процес на разработване на тези работи асистенти с общо предназначение, които са много различни от роботите за игра или хоби. Първите доставки на тези хуманоидни работи започнали през 2004г. до международните лаборатории и университети като високотехнологични роботизирани платформи за изследвания и разработки. Така че тези прогнози изглеждат реалистични за периода 2016-2019, особени като се има в предвид последните успехи в тази област.

Прогнозите са, че продажбите на всички видове работи за домашни задачи, могат да достигнат почти 31 млн. единици за периода 2016-2019г., с прогнозна стойност на продажбите от 13.2 милиарда долара. Големината на пазара за работи играчки и хоби системи се очаква да е около 8 млн. броя, повечето от които по обясними причини са на много ниска цена. Броят на продажбите за същия период на работи за образование и научни изследвания се очаква да е около 3 млн.



Фигура 1.4. Продажби на сервизните работи за персонална употреба.

Около 11 млн. броя роботи за игра и хоби, се предвижда се продадат, на стойност около 9.1 млн. долара. Продажбите на роботи за подпомагане на възрастни хора и инвалиди ще бъдат около 37 500 броя за периода 2016-2019. Този пазар се очаква да нарасне значително в рамките на следващите 20 години.

1.3. Описание на развитието и проблемите при сервизните роботи за подпомагане на човека.

В този научен труд се разглеждат сервизни роботи, 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 роботи, 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 роботи, 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.



Фигура 1.5. Обслужващи роботи със сложна функционалност.

Повечето от тези роботи 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.

Според статистическите данни на ООН от 2015 година, броят на възрастното население в световен мащаб се увеличава [6]. Един от начините за решаване на проблемите със застаряващото население е да се създадат сервизни роботи, които да бъдат техни лични помощници или да заменят някои от ежедневните човешки дейности.

Техническите средства позволяват на възрастните хора и хората с увреждания да живеят независимо в своите домове, ако те пожелаят. Именно сервизните роботи са тези технически средства, които да изпълняват тези функции [7]. Тези роботи 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:
 - Взимане 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.

Роботите предназначени за обществените сгради трябва да имат добра навигационна система, да общуват с хората и да работят безопасно и надеждно. Пример за успешен проект са роботите инсталирани в „Музея на Комуникациите Берлин“, където те автономно се движат сред посетителите, комуникират и взаимодействат с тях (фиг. 1.6). Роботите са създадени, за да общуват и да забавляват посетителите в музея. Техните задачи включват посрещане на посетители, водене на обиколка с екскурзовод през музея или игра с топка. Описание на хардуерната платформа на роботите и техните функции като навигационни и комуникационни умения, тяхната концепция за безопасност и работа са публикувани в [8]. Освен това е описан основният софтуер за управление на роботите. Накрая се представя приложението на роботите в Музея за комуникация в Берлин. От 25 март 2000 г. роботите работят в този музей без забележителни проблеми.



Фигура 1.6. Роботи в Музея на Комуникациите Берлин

Две белгийски болници са добавили иновативен персонал към рецепциите си: хуманоидни роботи наречени Раррег [9]. В тези болници роботите са помощник рецепционисти. Това е първият робот в света, който ще бъде използван за да посреща хората в болниците. Раррег може да разпознае човешкият глас на 20 езика и да определи дали говори с мъж, жена или дете. Цената на този робот е 30 000 евро (23 000 долара). Основното предназначение на робота е разпознава емоциите на хората и да избере поведение, което е най-подходящи според ситуацията (фиг. 1.7).



Фигура 1.7. Роботът Раррег

Въз основа на вашия глас, изражението на лицето ви, движенията на тялото и думите, които използвате, Раррег ще интерпретира вашите емоции и ще предложи подходящо съдържание. Той също така ще реагира лично на настроението на момента, изразявайки се чрез цвета на очите си, таблета или тона на гласа си.

Друг тип много важни роботи предназначени за болниците са роботи за опериране и роботи за рехабилитация. Роботът Da Vinci се използва за извършване на операции от

няколко години. С хирургичната система "Да Винчи" хирурзите работят само с няколко малки разреза. Системата da Vinci разполага с увеличена триизмерна система за видеонаблюдение с висока разделителна способност и мънички развълнувани инструменти, които се огъват и въртят далеч по-силно от човешката ръка. В резултат на това da Vinci дава възможност на хирурзите да работят с подобрена визия, прецизност и контрол [10]. Системата da Vinci представлява най-новото в областта на хирургическите и роботизираните технологии. Хирурзите могат на 100% да контролират системата da Vinci по всяко време. Технологията da Vinci превежда ръчните движения на хирурга в по-малки и прецизни движения на малки инструменти в тялото ви.

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 повтораемост, 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].

Toyota демонстрира своя робот Welwalk WW-1000, машина, която може да рехабилитира жертвите на инсулт с около 60% по-бързо от обикновената физиотерапия. Дружеството показва и проблясъци на други роботизирани технологии, като например робот за поддръжка на хора, който взима неща, изтегля завеси и изпълнява други задачи, които лекарят обикновено трябва да се обади на медицинска сестра [13]. Тойота споменава и робот, облечен като плюшено мече, който се занимава с дълги разговори с пациента. Машините се произвеждат в роботизирана лаборатория за 200 души в Хирос, Япония и се отдават под наем на болници. Robot Welwalk WW-1000 може да бъде отдаден под наем за около \$ 3,200 месечно. За момента ботът е достъпен само в Япония. При продажбите на други земи би трябвало да се адаптира към структурата на населението. Сегашният модел работи само с пациенти до 95 кг и височина до 190 см. По-тежките и по-високите трябва да търсят ръчна грижа.

Интересен е въпросът с развитието и разработките на работи, предназначени да живеят в дома на хората. Има редица такива работи, които на този етап или могат да изпълняват само по няколко задачи или могат по-голям брой задачи за сметка на изключително висока цена за потребителя.

Роботът Pepper например, не може да взаимодейства с околната среда за да взема и доставя предмети на хората, а може основно да комуникира с тях. Други сервизни работи като PR2 и Care-O-bot имат значителни постижения в автономното придвижване, манипулация с предмети и компютърно зрение, но те изпълняват основно команди зададени от потребителски интерфейс, а цената им е много висока [14], [15].

Друг робот със завидни възможности е производеният от Хонда робот – Asimo. Той е крачещ робот с възможности за директна комуникация с хората, манипулация с обекти както и автономно движение в помещения, включително изкачване на стълби [16]. Тук отново остава проблема с цената на работа. Част от актуалните цени на някои работи са показани на [17]. Цените са следните:

- Asimo - US \$2,500,000
- Care-o-bot 3 – US \$250,000
- PR2 - US \$280.000
- Pepper - US \$16,000
- NAO - US \$7500
- DARwIn-OP (ROBOTIS OP) - US \$9600
- HUBO 2 Plus - US\$ 400,000

1.4. Актуални проекти за изследвания на сервизни работи за подпомагане на хората.

В тази точка са разгледани някои популярни и значими проекти отнасящи се за сервизни работи. Като по този начин се показват основните проблеми свързани с изследванията и разработките в сферата на сервизната роботика.

- Проекти за работи, които да подпомагат възрастни хора и инвалиди.
- **KSERA**: Knowledgable SErvice Robots for Aging: разработване и изследване на интегрирането на подпомагащи технологии за дома и сервизни работи, които да подпомагат възрастни потребители в домовете им. Системата KSERA помага на възрастните хора и специално на хората с белодробни заболявания с ежедневните им движения и осигурява грижи и средства за ефективно самоуправление. Основната цел на този проект е да се проектира приятен, лесен за използване и социално активен робот асистент. Този робот ще осигурява полезна информация и постоянна грижа на правилното място, като за това използва информация от сензори, разположени в дома на възрастните хора [18].
- **CompanionAble** - Integrated Cognitive Assistive & Domatic Companion Robotic Systems for Ability & Security: комбинира интелигентните домашни системи с робота Хектор (Hector), напълно автономен робот, създаден да изпълнява ролята на „компаньон“ за възрастните хора (най-вече за тези, които живеят сами или прекарват по-голямата част от деня сами). Роботът ще им помага да запазят независимия си начин на живот, ще осигурява сигурност и удобство, чрез механизми за разпознаване на падане интегрирани с спешни повиквания или услуги за дистанционен мониторинг. Персонализиран диалог/взаимодействие между робота и човека ще показват емоционалното състояние за да се избегнат чувства на самота, за да напомнят за вземане на важни предмети като ключове, портфейл, ще предлагат познавателни игри, както и лесна видео връзка със семейството и приятели. Без подходящо стимулиране на когнитивните умения, възрастните хора страдащи от деменция и депресия могат лесно да се влошат [19].
- **FLORENCE** - Multi Purpose Mobile Robot for Ambient Assisted Living: системата FLORENCE със своята многофункционална мобилна роботизирана платформа въвежда употребата на роботите в доставянето на нов вид услуги на възрастните хора и хората, които се грижат за тях. Основната цел на този проект е идеята да се възприеме добре от потребителите и да се постигне рентабилност за обществото и болногледачите. FLORENCE ще постави робота като свързващото звено между

няколко самостоятелни услуги в домашна обстановка, както и между услугите и възрастния човек [20].

- **SRS** - Multi-Role Shadow Robotic System for Independent Living: проектът демонстрира иновативна, практическа и ефективна система наречена „робот сянка“ за персонализирани домашни грижи. Решенията на SRS са предназначени да позволят на робота да действа като сянка на този, който го управлява. Например възрастните родители могат да имат робот, който да бъде в ролята на сянка на техните деца или гледачи. В такава ситуация техните деца или гледачи могат да им помагат дистанционно и физически със задачи като ставане или лягане в леглото, оправяне на прането и настройване на ИКТ оборудването, сякаш децата или обслужващите хора живеят в къщата [21].
- **MobiServ** - An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly: целта на този проект е да се създадат и използват обновени технологии по координиран, интелигентен и лесен начин за да подпомагат самостоятелния живот на възрастни хора, колкото се може повече в техните домове или различни степени на институционализация. Подкрепата ще се предоставя във вътрешни ежедневни ситуации. Произвежда се персонална интелигентна платформа, състояща се от различни софтуерни пакети и устройства, плюс първичен набор от функции. Физическите елементи са роботизирана платформа, оборудвана с камери, безжични устройства за комуникация, умна домашна инфраструктура (Wi-Fi, сензори, централен домашен сървър) и интелигентни тактилни вградени сензори [22].
- **ACCOMPANY** - Acceptable robotiCs COMPanions for AgeiNg Years: системата ACCOMPANY е изградена от роботизиран компаньон като част от интелигентната среда, която предоставя услуги на възрастните потребители по мотивиращ и социално приемлив начин, за да улеснят независимия живот у дома. Системата ще предостави физическа, когнитивна и социална подкрепа в ежедневните задачи. Услугите за потребителя ще се доставят чрез социално интерактивно, приемливо и емпатично взаимодействие, основаващо се на изчислителни модели на социалното познание и взаимодействие на роботите. Предполагамата връзка на потребителя с робота е тази на съучастник - робот и потребител, които предоставят взаимопомощ на потребителя да не бъде доминиран от технологията, а да бъде овластен, физически, когнитивно и социално [23].
- **DALi** - Devices for Assisted Living: DALi е насочен към потребителска група, състояща се от възрастни хора с възникващи нетежки когнитивни увреждания. Те често се усложняват от влошаване на слуховите и визуални способности. DALi предлага когнитивна навигационна протеза, устройство, което подпомага навигацията в неструктурирана и претъпкана среда чрез придобиване на сензорна информация, като прогнозира намерението на човешките агенти и решава пътя, който свежда до минимум риска от въздействия или антисоциално поведение. Със скорошните постижения в сензориката, автоматизирано разсъждение и моделиране на тълпата, тази амбициозна цел става технологично осъществима. Проектът ще

създаде прототип на нов когнитивен асистент за ходене (c-Walker), устройство, базирано на стандартна мобилна роботизирана платформа [24].

- **Robot-Era (IP)** - Implementation and integration of advanced Robotic systems and intelligent Environments in real scenarios for the ageing population. Целта на Robot-Era е да разработи, приложи и демонстрира общата осъществимост, научно/техническата ефективност и социално/правната правдоподобност и приемливост от крайните потребители на множество съвременни роботизирани услуги, интегрирани в интелигентна среда. Тези системи активно ще работят в реални условия и ще си сътрудничат с истински хора и между тях, за да подкрепят независимия живот, да подобрят качеството на живот и ефективността на грижите за възрастните хора [25].
- **DOMEO** - Domestic Robot for elderly assistance: Две роботизирани платформи се разработват в проекта DOMEO, RobuMate за когнитивна стимулация и ежедневна помощ и RobuWalker за помощ при ходене. Тези две платформи са свързани с отдалечен медицински център чрез уеб интерфейс [26].
- **ALIAS** - The Adaptable Ambient Living Assistant: Целта на проекта ALIAS е разработването на мобилна роботизирана система, която взаимодейства с възрастните потребители, осигурява помощ в ежедневието и насърчава социалното приобщаване чрез създаване на връзки с хора и събития в света. ALIAS представлява мобилна роботизирана платформа с възможност за наблюдение, взаимодействие и достъп до информация от онлайн услуги без манипулационни възможности. Освен интегрирането на съществуващите решения, ще бъдат въведени две новости: а) въведена е нова концепция за когнитивен потребителски интерфейс, за да се осигури добра използваемост и да се избегнат страховете на хората да навредят на робота. Б) проактивното поведение на роботизираната платформа ще гарантира, че потребителят ще остане в контакт със заобикалящата го среда и ще бъде стимулиран психически. В) Третата уникална точка за продажба е Brain-Computer-Interface (BCI), която ще бъде включена, за да се обучават и запазват умствените функции на потребителя [27].
- **SILVER** - Supporting Independent LiVing for the Elderly through Robotics: търси и проучва нови технологии базирани на роботиката, за да подпомагат възрастните хора в тяхното ежедневиe. Целта е да се осигури независим начин на живот на възрастните хора в техните домове, дори те да имат физически или когнитивни проблеми. Уникалното в този проект, е че се използва Pre-Commercial Procurement (PCP) процес, за да се намерят и изберат нови технологии и решения. В Европа PCP е недостатъчно използван инструмент за насърчаване на иновациите. Една от целите на този проект е да демонстрира ефективността на подхода на PCP за справяне с обществените и правителствените нужди [28].

Както се вижда от кратките описания на тези проекти, основно се набляга на подпомагане на възрастни хора и инвалиди. Това е така поради големите нужди от подпомагане на тези хора. Създаването на роботи с различни възможности ще допринесе за създаването на универсален робот, който да може да извършва всички или повечето от необходимите задачи. Увеличаването на изследванията ще доведе и до намаляване на

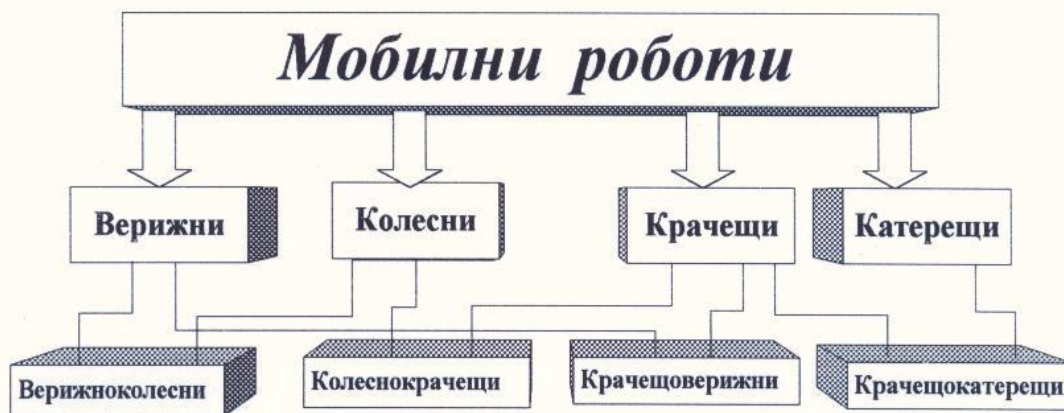
цените на самите работи. Взимайки в предвид актуалната информация относно развитието на сервизните работи, основните направления, които са предмет на изследвания в световен мащаб са следните:

- Мобилност и автономност.
- Управление и комуникация.
- Сензорни системи и компютърно зрение.
- Изкуствен интелект.
- Цена и възможности за приложение и разпространение.

1.5. Мобилност на сервизните работи.

Мобилните работи са разпространени навсякъде в нашето ежедневие. Като тенденцията за в бъдеще е те да станат дори част от живота на хората. Мобилните работи се използват във всички сфери на индустрията, медицината, образованието, военните технологии, космически експедиции и други. Новостите дори предвиждат внедряването на мобилни работи в домовете на хората. Развитието и използването на мобилни работи е сериозно и ще продължава да нараства с големи темпове.

Мобилната роботика е интересна област за изследвания на видове движения и възможностите им за използване при машини, които трябва да се придвижват автономно, при това често пъти без пътища, специално подготвен терен, при наличие на препятствия, тесни места и др. Един от възможните подходи за класификация на видовете мобилни работи е според начина на придвижване. В този смисъл те са колесни, верижни и крачещи. Движението може да бъде непрекъснато или прекъснато. При роботите с прекъснато движение имаме и катерещ се тип. В практиката освен това се срещат специфични мобилни работи, напр. летищи, както и такива от смесен тип. Една примерна класификация е дадена на фигура 1.8.

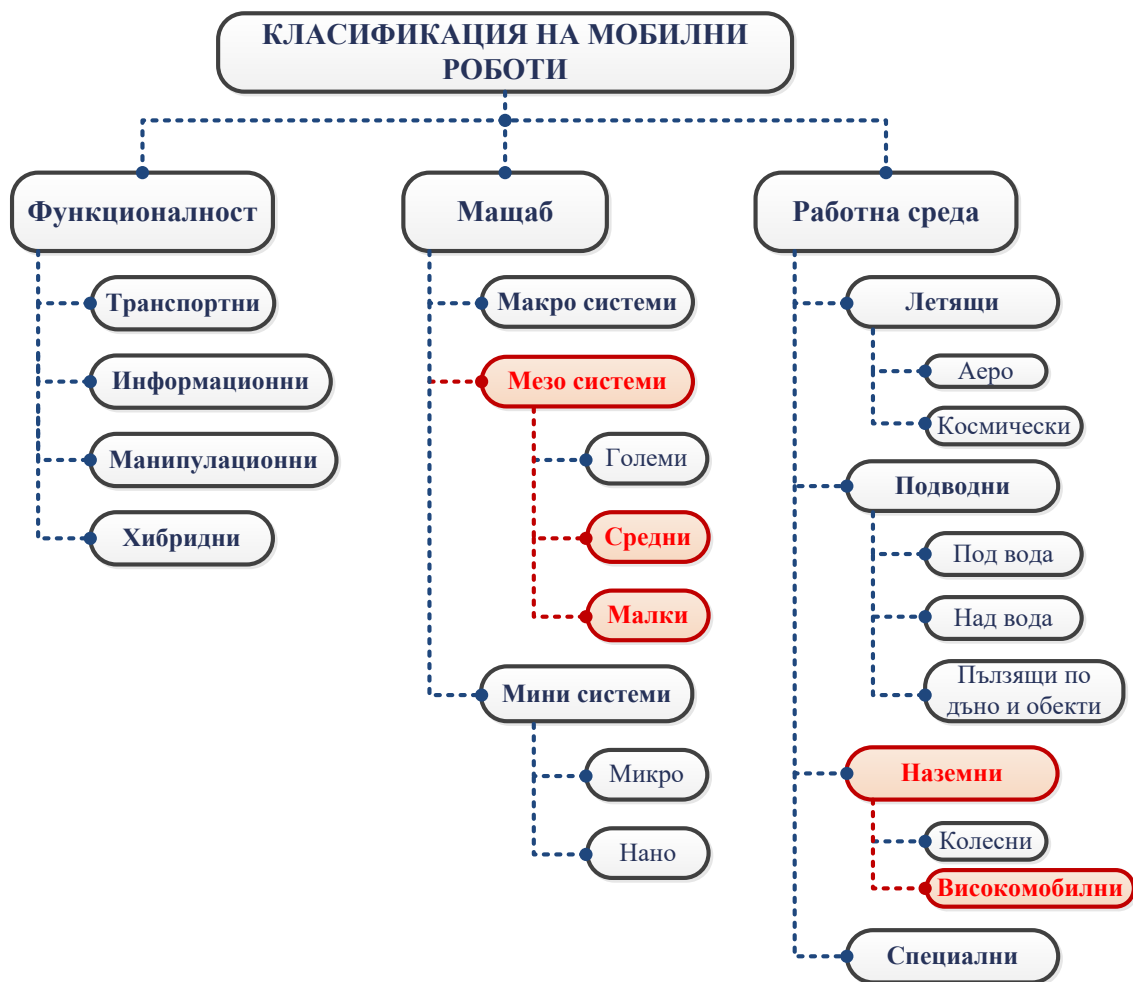


Фигура 1.8. Класификация според начина на придвижване

1.5.1. Видове мобилни платформи.

Различните области на приложение и много нужди на хората са допринесли за наличието на голямо разнообразие от мобилни работи, имащи най-различни

възможности и функции. Днес разнообразието на мобилни работи е много голямо. Те представляват технически системи с широко разнообразие от възможности и приложения. Определящо значение за техните технически характеристики имат вида на работната среда, мащаба на локомотивните механизми и движения, както и поставените цели и задачи. Обща класификация на мобилните работи е показана на фиг. 1.9.



Фигура 1.9. Класификация на мобилните работи

Както се вижда на фигурата според функционалните си възможности мобилните работи се определят като:

- Транспортни
- Информационни
- Манипулационни
- Хибридни

Към групата на транспортните МР спадат автоматичните мобилни платформи (МП), използвани за транспортиране на обекти от най-различно естество. Те намират приложение в болнични заведения за транспортиране на храна, медицински и хирургически консумативи, чисто или замърсено бельо, медицински отпадъци, в

складове и цехове за пренасяне и повдигане на материали , за теглене на композиции и тежкотоварни превози, (фиг. 1.10).

Информационни са тези работи, които се използват за наблюдение и измерване параметрите на средата, патрулиране и охрана, автоматизирано обслужване на клиенти и мобилни автоматизирани гидове.

За да се определи един робот като манипулационен, той трябва да има способността активно да въздейства върху средата и/или обектите в нея [29]. Най-често това са функции, като взимане на проби и извършване на технологични операции, асистирание на човек при пренасяне на тежки материали, прецизиране на движенията и т.н. Към тази група се причисляват не само мобилните роботи, които са с изграден многофункционален манипулатор или имат възможност за допълнително вграждане на такъв, но и роботизираните протези, чието развитие през последните години значително се усъвършенства (фиг. 1.11).

Последната група МР по показателя функционалност са хибридните. Както става ясно от наименованието им, то това са работи, които по функционалност комбинират две или повече от изброените функции.

Другият показател по който могат да се класифицират МР е техният мащаб. Размерите на МР, както и на ходовете на възможните локомоционни движения естествено дефинират системите като: макро, мезо и мини.



Фигура 1.10. Тежкотоварен робот



Фигура 1.11. Роботизирана протеза

Развитието на МЕМС технологиите създаде предпоставки за изграждане на микророботи, а в перспектива и на наносистеми. Многообразието на мезосистемите (с антропоморфни габарити, ходове и обекти) изисква допълнително структуриране на малки, средни и големи МР.

Спецификата на работната среда е формирала следните основни структурни групи МР:

- летящи системи с 6 степени на свобода в атмосферата (многовитлови платформи (фиг. 1.12), аеростати, махащи (фиг. 1.13) и др.) и в космоса (вкл.

Многоръки с възможности за придвижване чрез прехващане – пълзене) (фиг. 1.14);

- подводни – с 6 степени на свобода и специфични възможности за движение по дъно, по обекти и по водната повърхност (фиг. 1.15);
- наземни МР за придвижване по повърхност (до 3 степени на свобода) в условия на гравитация и триене.

Те използват естествени терени, пътища, улици, площадки, вкл. Тунели под земята, както и трасета в сгради и инженерни системи;

- специални – за движение в тръби, по електропроводи, придвижване чрез издърпване, прилепване към стени и тавани, ровене в рохкави структури, движение в организми (вкл. Човешки).



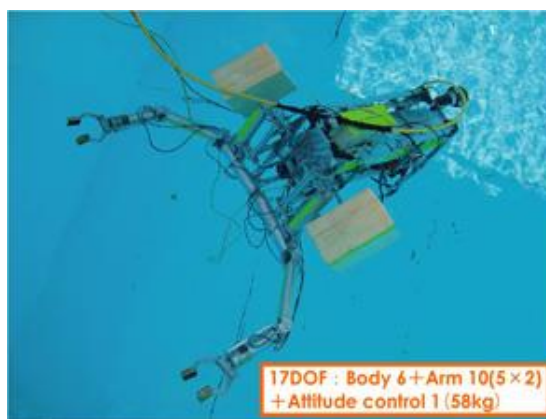
Фигура 1.12. Многовитлова платформа



Фигура 1.13. Аеростат



Фигура 1.14. Многорък робот



Фигура 1.15. Подводен робот

Съвременните наземни мобилни роботи се делят на конвенционални и високомобилни и са обособен клас универсални системи с приложение в редица дейности:

- междуцехов и вътрецехов и складов транспорт, информационни и транспортни задачи и чистене в производството и обслужването
- анализ и работа в опасни среди, при бедствия и аварии, при военни и охранителни дейности

- лабораторни изследвания и обучение

Докато първата група включва специализирани и относително евтини автоматични устройства, втората се състои от многофункционални, високомобилни системи с дистанционно управление. В този случай сложността, разнообразието и динамиката на средата и задачите изискват участие на специализиран оператор или екип за наблюдение, анализ, вземане на решения и водене на МР.

Най-често работната сцена на апарата включва:

- сградни помещения и технически съоръжения (индустриални и транспортни)
- урбанизирани райони (сгради, дворове, канали, тунели, пътища, площадки и ж.п. линии)
- естествена среда при различен релеф и повърхност

Широкото разпространение на изкуствени, плоски, гладки и твърди основи (подове, улици и площадки), позволява бързо и ефективно придвижване чрез колесни МП с конвенционални конструкции:

- триколесни – диференциални (фиг. 1.16)
- класически четириколесни (фиг. 1.17)
- двуколесни балансиращи (фиг. 1.18)
- едноколесни балансиращи (фиг. 1.19)



Фигура 1.16 Триколесен робот



Фигура 1.17 Четириколесен робот



Фигура 1.18 Двуколесен робот



Фигура 1.19 Едноколесен робот

Системите са прости, но позволяват само заобикаляне на препятствия в равнина. Разширяването на задачите, профила и вида на основата, както и многообразието от препятствия в пространството налага използването на висококомбилни роботи.

Висококомбилните роботи могат да се класифицират на следните конструктивни групи (фиг. 1.20):



Фигура 1.20 Висококомбилни платформи

Високата мобилност на многоколесните роботи (Фиг. 1.21, 1.22, 1.23) се дължи на индивидуалното управление на всяко колело и геометрично близкото разположение на осите на колелата, като по този начин всяка група колела може да се разглежда, като едно единствено с елипсовидна форма или верига.

Роботите използващи омниколела (Фиг. 1.24) имат подобрена подвижност и маневреност, дължаща се на свойствата на омниколелата. Характерното за тях е, че имат едновременно активна и пасивна посока на въртене. Посоката на движение се управлява чрез посоката на въртене на отделните колела, като при еднаква скорост се наблюдава праволинейно движение. При променлива скорост на кое да е от колелата движението става вече по крива.



Фигура 1.21 Четириколесен робот



Фигура 1.22 Многоколесен робот



Фигура 1.23 Многоколесен робот



Фигура 1.24 Робот с омни колела

Крачещите мобилни роботи (Фиг. 1.25) се характеризират с голяма мобилност дължаща се най-вече на малкия брой контактни точки между робота и повърхността на движение. Те имат голяма адаптивност, маневреност и способност да преодоляват препятствия, каквито другите видове наземни роботи не могат. Основният недостатък на тези роботи са сложността на хранящите и механични конструкции на краката. Един крак трябва да може да поддържа самостоятелно голяма част от общото тегло на робота, а и в повечето случаи да го повдига или спуска.

Пионер при създаването на змиевидни роботи е професор Shigeo Hirose от Токийският технологичен институт. Високата мобилност на тези роботи се дължи на хипер-редундантни локомотори, които използват от една страна, и от друга на факта, че те имат подобно на змия тяло с функции на „крака“, когато се движат, с функции на „ръце“, когато се изкачват и на „пръсти“, когато носят нещо.

Локомоционната система на високомобилни хибридни роботи е съвкупност от два или повече начина за придвижване. На практика се срещат почти всички възможни конфигурации като: крачещо-колесни, крачещо-верижни, колесно-верижни и др.. Подобни роботи имат относително високо ниво на сложност, тъй като е необходимо да се използват специални окачващи механизми, изместване център на тежестта, асиметричен дизайн (фиг. 1.26, 1.27).



Фигура 1.25 Крачещ мобилен робот



Фигура 1.26 Хобриден робот



Фигура 1.27 Хобриден робот



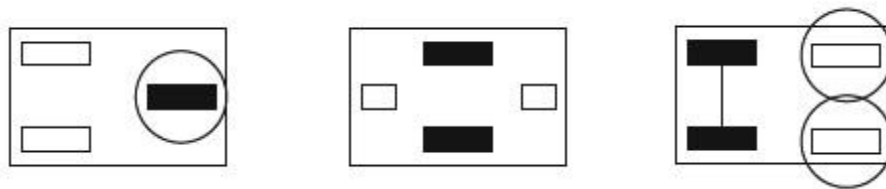
Фигура 1.28 Хобриден робот

Последната категория високомобилни роботи са верижните. Задвижването с вериги на практика прави задвижването на робота „некоректна“ диференциална система от най-прост тип. Тази категория се разделя на две суб-категории – верижни мобилни платформи и верижни мобилни роботи (фиг. 1.28). Последните се изграждат на базата на платформите, но с тази разлика, че могат и активно да въздействат върху средата.

1.5.2. Характеристики на мобилните роботи.

Мобилните платформи са много видове, в зависимост от нуждите за преодоляване на различни препятствия. Най-разпространените и използвани са платформи на колела, омни-колела, верижни и крачещи.

Най-простия случай за мобилен робот е колесния робот, показан на фиг. 1.29. Колесните роботи включват едно или повече задвижващи колела и имат няколко пасивни или евентуално за завиване. Повечето такива платформи изискват два двигатели за задвижване и ако е необходимо един за завиване.



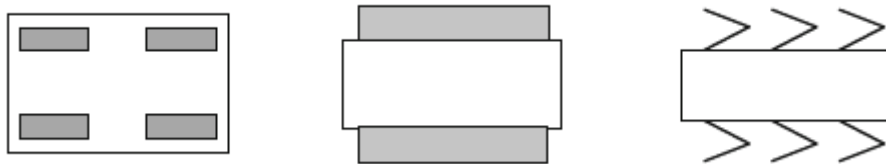
Фигура 1.29 Колесен робот

Дизайнът на платформата вляво на фиг. 1.30 има единично задвижващо колело, което е и завиващо. Изискват се два двигателя, един за задвижване и един за завиване. Предимството на тази конструкция е, че изпълнението на двете функции движение и завиване са напълно разделени, чрез използването на два двигателя. Следователно управляващия софтуер за задвижване и завиване ще е много лесен. Като недостатък е, че роботът не може да се върти на място, тъй като задвижващото колело не е разположено в центъра на платформата.

Дизайнът на платформата в средата на фиг. 1.30 се нарича „диференциално задвижване“ и е един от най-често използваните дизайни за мобилни платформи. Комбинацията от две задвижващи колела позволява робота да се движи напред, да завива, и да се върти на място. Връзката между командите за управление, например крива на даден радиус и съответните скорости на колелата трябва да се осъществи посредством софтуер. Друго предимство на тази конструкция е, че двигателите и колелата са с фиксирани позиции и не трябва да се въртят както при предния дизайн. Това значително опростява дизайна на механиката на робота.

И накрая в дясно на фиг. 1.30 е платформа с така нареченото „акерман управление“, която е стандартно задвижвана и завиваща система на кола със задно задвижване. Имаме един двигател, който задвижва и двете задни колела с помощта на диференциална кутия и един двигател за управление на завиването на двете предни колела.

Важно е тук да се отбележи, че и трите типа дизайн на мобилните платформи изисква два двигателя, които са достатъчни за задвижване и завиване.



Фигура 1.30 Колесен, верижен и крачец робот

Специален случай на колесни роботи са омни-колесните на фиг. 1.30. При тях се задвижват четирите колела, които са със специализиран дизайн.

Един много голям недостатък на колесните роботи е, че те изискват сравнително гладки повърхности, по които да се движат. Верижните роботи (фиг. 1.30, в средата) са

по-гъвкави и могат да се движат по неравен терен. Имат голяма проходимост като движение нагоре и надолу или преодоляване на препятствия и са лесни за управление. Недостатък е, че нямат точността на колесен робот, но с необходимите системи и сензори са надеждни. Верижните платформи се нуждаят от две двигателя, по един за всяка верига.

Крачещите роботи (фиг. 1.30, в дясно) са последната категория наземни мобилни роботи. Подобно на верижните роботи, те могат да се движат по неравен терен и да преодоляват препятствия. Главното правило е: колкото повече крака, толкова по лесен баланс. Например шест-кракия робот от фигурата може да се управлява по такъв начин, че три от краката винаги да са на земята, докато другите три са във въздуха. Крачещите роботи обикновено изискват по два или повече двигатели на крак, така че на шест-крак робот са необходими 12 двигателя. Използването на повече двигатели позволява по-голяма свобода на движенията, но за сметка на тежестта на платформата и цената на частите [30].

1.5.3. Локализация и навигация.

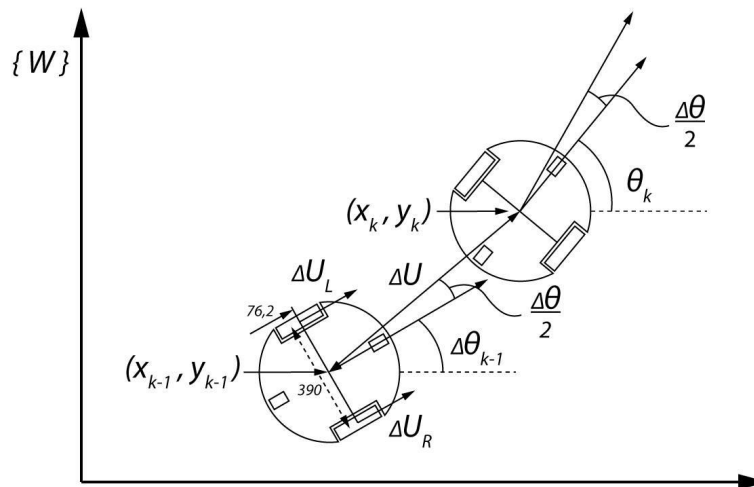
При мобилните роботи основен проблем е тяхното придвижване от точка до точка, намиране на текущото местоположение и откриване, избягване или преодоляване на препятствия. Тези проблеми са свързани с навигация и локализация. Локализацията е проблем свързан с намиране на позицията на робота спрямо други обекти около него в статична среда. За да се осъществи локализирането на робота трябва да има създадена карта или да се създава карта по време на работа за да може да се създадат ориентир за намиране на позицията. При локализацията се прилагат различни подходи за намиране на позицията на робота, например: Локализация Монте Карло, филтри на Калман, относителна и абсолютна позиция и други. Тези подходи се прилагат за да се намали грешката при намиране на местоположението като се комбинират данните от сензорите с несигурните данни и се оптимизират.

Системата за навигация при мобилните роботи изисква точна оценка на текущото положение на робота и точна карта на помещението. Тя се разделя на две нива на управление – глобално планиране на път и локално управление на движението. Планирането на път изисква модел или карта на средата, за да може да определи геометричните точки на пътя, по който да се движи робота от начална позиция до целта. Локалното управление използва данните от сензорите и определя движението, което ще избегне сблъсъци с непознати обекти или препятствия, чиято позиция се е променила. SLAM процесът (Simultaneous Localization and Mapping) е основен метод, използван при локализация и навигация. Основната цел на този процес е да следи за текущата позиция на робота като непрекъснато я опреснява. За прецизно определяне на тази позиция се използва лазерен скенер. Това се постига чрез извличане на данни от околната среда и повторно наблюдаване, когато робота се движи, т.е. непрекъснато сканиране на средата по време на движение на робота. Тези методи позволяват постигане на автономност при движение на роботите.

1.5.4. Одометрия.

Използвайки прости геометрични уравнения, е лесно да се изчисли моментното положение на робота спрямо известната стартова позиция. Това изчисление се нарича одометрия. Одометрията изчислява относителното хоризонтално преместване на робота и промяна в ориентацията, като функция на нарастващото хоризонтално преместване на задвижващите колела (Фиг.1.31).

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. Това представяне има ограничения. Пример за тези ограничения е приплъзване на колело: ако колелото се плъзне по мокър под, тогава енкодера ще регистрира завъртанията на колелото, но тези обороти няма да съответстват на реално изминатото разстояние от колелото.



Фигура 1.31. Параметри на одометрията.

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:

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

Одометричните грешки се делят на две категории:

Систематични грешки: разлика в диаметъра на колелата, неточности в механиката на платформата;

Несистематични грешки: боксуване, приплъзване, изпускане на обороти и др.;

Систематичните грешки са тези грешки, които се дължат на дефекти и разлики в самата мобилна платформа на робота. Тези грешки обикновено са константни и могат да се коригират, като се приложат математически модели, без да е необходимо да се правят допълнителни измервания. Примери за систематични грешки са:

Нееднакви диаметри на задвижващите колела;

Несъосност на колелата;

Промени в ефективното сцепление на платформата;

Ограничение в резолюцията на енкодерите;

Ограничена честота на дискретизация на енкодерите.

Несистематичните грешки са тези грешки, които са причинени от взаимодействието на робота с непредвидими особености на околната среда. Например нарушения по повърхността на пода, като подутини, пукнатини или остатъци от боклук, може да са причина за повече завъртания на колелото, отколкото е предвидено, защото колелото върви нагоре или надолу по неравността. Примери за несистематични грешки:

Движение по неравен под;

Преминаване през някакъв предмет на пода;

Боксуване на колелото поради: хлъзгав под, голямо ускорение, рязко завиване, загуба на контактна точка с пода.

Систематичните грешки се проявяват най-често при движение по равни и чисти настилки на закрито. При движение по такива повърхности тези грешки оказват по-голямо влияние на одометрията отколкото несистематичните. Въпреки това при движение по неравни повърхности несистематичните грешки може да бъдат повече. Нашият робот е предназначен да се движи на закрито, като коригираме и двата типа одометрични грешки за постигане на добра точност. Измерването на одометричните грешки е процес, който включва множество експерименти и анализ на техните резултати [31].

1.6. Манипулатори при сервизните работи.

1.6.1. Популярни роботизирани манипулатори.

- PR-2: Този робот разполага с две антропоморфни ръце. Всяка ръка притежава по шест степени на свобода. Те могат да работят в една координатна система. Предназначени са за извършване на сложни движения като сгъване на дрехи, хващане на неудобни предмети, готвене и други.

- Care-O-Bot: този робот е оборудван с роботизирана ръка, която притежава седем степени на свобода. Той е предвиден за домашен помощник на хората и основните задачи за ръката са да хваща и поставя различни предмети.
- SRA Service robot arm: това е прецизна ръка с шест степени за свобода. Тя е пригодена специално за сервизната роботика. Притежава голяма точност при позициониране. Може да манипулира с стандартни обекти, които са част от домашната среда.
- Mover series, robot arms: роботите от серията Mover са три на брой, като разликата е в степените на свобода, съответно Mover4 Mover5 Mover6 са с четири, пет и шест степени. Те са предназначени за обучение, но предлагат добри възможности за приложение в научните изследвания, както и в сервизната роботика.

1.6.2. Управление.

Управлението на роботизирани манипулатори е сложен процес, при който се използват различни методи и принципи. Последователността от действия за управлението на един манипулатор е следната: управление на задвижващите елементи на ставите, координиране на движенията на ставите, планиране на действията на манипулатора, задаване на крайна цел, до която да се достигне. Някои ръце се управляват посредством джойстик или бутони от потребителски интерфейс. Други използват сложни методи като разпознаване движенията на човешката ръка, след което манипулатора изпълнява същото движение [32].

За да се планира управление на манипулатор по зададена траектория трябва да се решат правата и обратната задачи на кинематика, както и да се реши матрицата на Якоби за да се планират силите и ускоренията. За обратна връзка при управлението се използват енкодери, разположени във всяка от ставите. Друг метод за обратна връзка е използване на външен сензор, като камера или сензор за измерване на разстояние. Важен принцип при управлението на такива работи е да няма хлабини и луфтове, както и моторите да притежават достатъчно точност и мощност, а енкодерите да имат голяма резолюция.

1.7. Управление на сервизните работи.

1.7.1. Работа в реално време.

При сервизните работи работата в реално време е задължителна. Предвид тяхното предназначение и те трябва да имат способността да се адаптират и да реагират на динамични промени в средата. Особеното при системите, работещи в реално време е, че обработката на информацията и управлението се извършват в зависимост от външни събития, настъпващи в предварително неизвестни моменти от време. Реакцията на всяко събитие включва последователност от процедури, които се изпълняват в строги интервали от време. Системите по принцип могат да работят дълго време автономно без външна намеса (за приемане на данни или подаване на задание). При работа в реално време навременната реакция е по-важна от ефективното натоварване на системата.

Много от процесите, представени чрез потоци от сензорни данни и управляващи въздействия, текат едновременно. Следователно и програмите, които отговарят за тях, трябва да работят паралелно.

Най-общо системите за реално време биват:

- периодични – базирани на работа с твърда тактова честота,
- аperiodични – базирани на сигнал от външно събитие,
- интерактивни – отговарят на външно събитие (например заявка от клиент в банка), но забавянето на отговора не довежда непременно до разстройство на системата (може да зависи от претовареност на линиите за комуникация).

В този смисъл можем да разделим системите за реално време на:

- твърди реално-временни системи – при които времевите изисквания се определят само от обекта и нарушаването им води да грешки и разстройство на системата,
- гъвкави реално-временни системи – при които времевите изисквания се определят и от възможностите на компютърната система и нарушаването им не води непременно до грешки и разстройство.

Аналогично, програмите в една компютърна система, представящи протичащи процеси, могат да се разделят най-общо на следните видове:

- последователни – поведението на програмата зависи само от операциите в нея и от техния ред,
- многозадачни - някои от операциите могат да протичат паралелно, те се конкурират и могат да си разменят данни,
- реално-временни – освен възможностите за паралелно действие и обмен на данни, има зависимост и от външни събития, данни.

За предоставяне на хардуерни ресурси, управление на програми и трансфер на данни в компютърните системи се грижат специални програми, наречени операционни системи. Освен тривиалния случай, когато една операционна система работи с една потребителска програма, най-общо имаме видове многопрограмни (multi-programming) системи, съответно операционни системи, а именно:

- многопотребителски (multi-user) - осигуряват на всеки потребител стартирането и управлението на собствена единична програма. Тя по определен ред и за определено време получава достъп до ресурсите на системата, но не може да взаимодейства с програмите на другите потребители,
- многозадачни (multi-tasking) – тук се предполага наличието на един потребител, който работи с много процеси (задачи - tasks), представени чрез съответни програмни процеси. Отделните програми се конкурират, взаимодействат си, комуникират и обменят данни помежду си.

Основното изискване към операционната система е да разпределя ресурсите между множество програмни процеси. При системите за реално време е характерно, че някои от процесите са по-важни от други, както и че времето за реакция е критично и често твърде кратко.

1.7.2. Операционни системи и софтуер за роботи.

Сред популярните операционни системи за роботи са Robot Operating System (ROS) и Microsoft Robotic Developer Studio (RDS 4). ROS е гъвкава рамка, за писане на софтуер за роботи. Представя колекция от библиотеки, инструменти и конвенции, които имат за цел да улеснят процеса на създаване на сложно и робастно поведение на множество роботи и платформи. RDS 4 е Windows базирана среда за учени, любители и разработчици, която им дава възможност да създават приложения за управление на различни хардуерни платформи. RDS притежава симулационни инструменти, библиотеки и функции за работа в реално време.

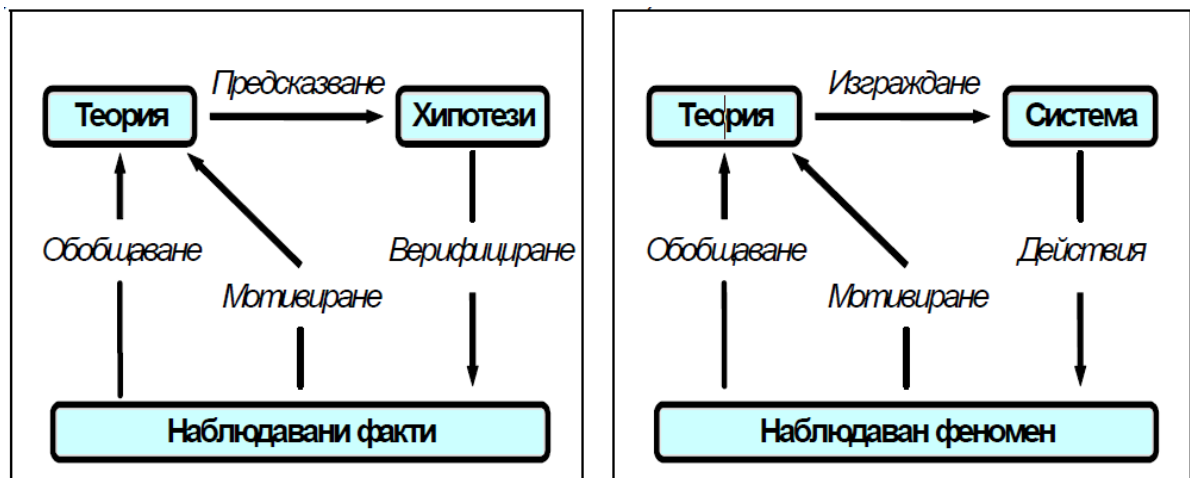
Open Source Computer Vision Library (OpenCV) е пакет от библиотеки с алгоритми за анализ, обработка и работа с изображения. Предоставя работа с три различни програмни езика: C++, Python и Java. Поддържа се от много операционни системи като Windows, Linux, Mac OS, iOS и Android. Може да разпределя изчисленията на ниво мулти-процесор и хардуерно ускорение, което ускорява значително работата с изображения.

Ардуино е популярна и често използвана среда за програмиране и управление на роботи. Програмният език е близък до C++ езика. Поддържа се за Windows, Linux и Mac OS. В софтуера има вградени много готови програми и алгоритми за управление на сензори, електромотори, модули за безжична комуникация и други.

1.7.3. Изкуствен интелект.

Изкуственият интелект (ИИ) е познавателна и инженерна научна област, която се занимава с разбиране и синтезиране на порожданото от интелекта интелигентно поведение. Интелигентното поведение е взаимодействието между система и среда, характеризиращо се с възможност да се знае, разсъждава и действа целенасочено. Интелигентното поведение включва: перцепция, логически разсъждения, обучение, вземане на решения и действия.

Всяка система, която интегрира процесите на перцепция, разсъждение и действия и е способна да води интелигентно поведение, може да се определи като интелигентен агент. Основна научна цел на изкуствения интелект е разбиране на принципите, които правят интелигентното поведение. Основна инженерна цел на изкуствения интелект е специфициране и развитие на методи за синтез и изграждане на интелигентни агенти.



Фигура 1. 32. Индуктивен и инженерен методи в ИИ

Целта на ИИ е разбирането на интелигентното поведение, чрез синтез на реални системи, които го осъществяват. Трябва да се има предвид, че компютърът е физически знакова система и може да осъществява реално интелигентно поведение на системи с перцепция и действия в еднородна, чисто компютърна (знакова) среда. В този смисъл, компютърното симулиране, само по себе си е резултат на реално осъществено поведение.

От друга страна, компютърното симулиране на движението на манипулатора на един робот, например, поражда законови описания на поведение, което не е реално осъществено. Аналогичен е и случаят при симулиране на някакъв вид перцепция. За осъществяване на реално поведение в такива случаи, освен компютъра са необходими още и нетривиални интерфейси (сензори и изпълнителни механизми), които изискват среда за реализация, различна от компютърната.

Роботиката е научна област, същността на която е интеграцията на перцепция, разсъждения и действия, осъществявани в разнородни физически среди. Роботите са вид (интелигентни) агенти и чрез тяхното изграждане се адресират не само въпросите за ролята на вътрешните механизми, които определят връзката му с разнородните физически среди. Ето защо, ИИ и роботиката са концептуално и методологически свързани в един процес на взаимно проникване и обогатяване.

1.8. Цел и задачи.

Целта на дисертацията е изследване, подобряване и създаване на методи, алгоритми и системи за управление на сервизни работи, предназначени за подпомагане на човека. Провеждането на тези изследвания трябва да допринесе за развитието на сервизните работи, като подобри и разшири техните възможности за изпълнение на различни по сложност задачи. Сервизните работи трябва да могат да се движат автономно, да разпознават основни предмети, да взимат и носят 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 работи: локализация и навигация, компютърно зрение, операционната система за работи (ROS), интернет интерфейс, сензорни системи,

интелигентни системи за управление, роботи 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.

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

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

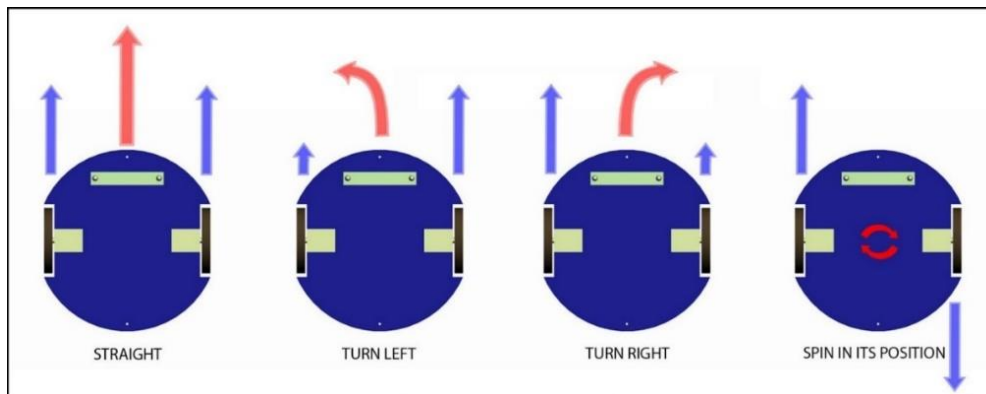
Робота върху, който се провеждат всички изследвания и експерименти е специално създаден за учебни и изследователски цели (фигура 2.1). Съставните модули на робота са мобилна платформа, антропоморфна роботизирана ръка РОВКО 01, два контролера за управление на електромоторите на платформата и ръката, Кинект сензор, инфрачервени и ултразвукови сензори и батерии.



Фигура 2.1 Сервизен мобилен робот

Мобилната платформа на робота е четириколесна. Страничните колела са активните, те са диференциално разположени и тяхното управление е сравнително лесно. Колелата разположени отпред и отзад са пасивни и служат за осигуряване на опора. За задвижване на активните колела се използват постоянно токови мотори с редуктори, които разполагат с еднокодери за измерване на оборотите на въртене. 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

Самата мобилна платформа е на три нива, където се разполагат останалите компоненти на робота. Най-долу са разположени моторите, главния контролер на робота, ултразвукови и инфрачервени сензори и батериите. На следващото ниво е предвидено място за компютъра на робота и на последното ниво са разположени ръката на робота, контролера и драйверите за управление на ръката и сензора Кинект. Параметрите на мобилната платформа необходими за нейното управление са:

- Радиус на колелата – 76,2 мм;
- Разстояние между колелата – 390 мм;
- Резолюция на енкодерите – 114 точки за едно завъртане.

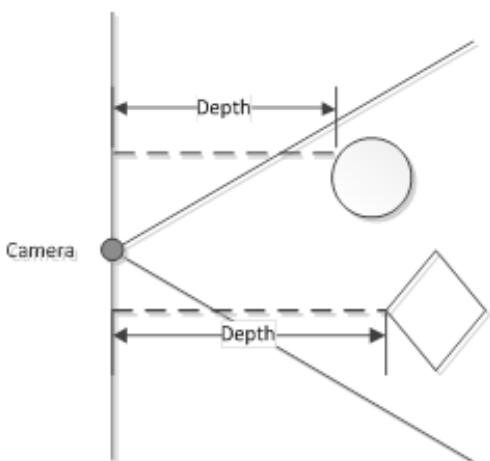
Главния контролер на робота е специално предназначен за управление на различни мобилни платформи с две активни колела. В основата на контролера е микро-контролерът Propeller P8X32A. Той разполага с осем 32 битови ядра, които предоставят голяма мощност и възможности. Контролерът включва два високо-токови драйвери за управление на мотори, осем канален 10 битов АЦП и множество цифрови входове и изходи. Няколко регулируеми захранващи напрежения и три превключващи се спомагателни захранващи порта, могат да захранват допълнителните модули на робота. Драйверите за управление на моторите са с H-мостове. Според документацията могат да издържат на максимален ток до 30А. При нормално и продължително натоварване е препоръчително токът да е до 10А, което е напълно достатъчно за задвижване на подобна мобилна платформа.

Сензорът Kinect се състои от следните компоненти [33]: Цветна камера, инфрачервен (IR) излъчвател и инфрачервен дълбочинен сензор и микрофон.

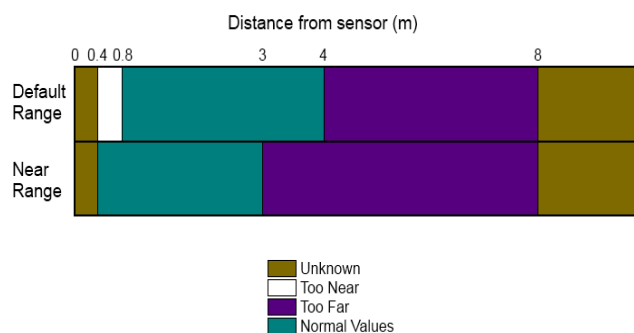
Във всеки кадър сензора за дълбочина заснема изображение с нива на сиво, на всичко попадащо в обхвата на сензора. Кадърът се формира от пиксели, чиито размери се специфицират от NUI_IMAGE_RESOLUTION номерацията. Всеки пиксел съдържа декартовото разстояние в милиметри, от сензора на камерата до най-близкия обект в двумерна координатна система (x, y). Тези координати не представят физическия обект, а показват локацията на дадените пиксели от заснетия кадър (Фиг. 2.3).

Когато се отвори потока на данните за измерванията на дълбочинния сензор NUI_IMAGE_STREAM_FLAG_DISTINCT_OVERFLOW_VALUES, можем да прочетем три вида стойности, които показват, че дълбочината не може да се измери коректно от тази локация. Променливата „too near“ показва, че е разпознат обект, но се намира твърде близо

за да се измери точната дистанция. Променливата „too far“ показва, че обекта е прекалено далеч за да се определи разстоянието до него. Променливата „unknown“ показва, че не е засечен обект (Фиг. 2.4).



Фигура 2.3. Обхват на дълбочинния сензор



Фигура 2.4. Допустими разстояния при измерване

Инфрочервен сензор - Sharp GP2Y0A21YK0F е инфрочервен сензор за измерване на разстояние, които използва лъч инфрочервена светлина да се отрази от обект, за да измери на разстоянието до обекта (Фиг. 2.5). Тъй като сензора използва триангулация на лъчева светлина, за да се изчисли разстоянието, сензорът е в състояние да осигури последователни и достоверни показания, които са по-малко чувствителни от изменението на температурата или отразяваща способност на обекта. Сензорът извежда аналогово напрежение, съответстващо на разстоянието на обекта, и може лесно да се чете чрез използване на евтина аналогово-цифров преобразувател (ADC) чип.

Ултразвуков сензор - осигурява лесен метод за измерване на разстояние. Този сензор е идеален за всякакъв брой приложения, които изискват от вас да се извършат измервания между движещи се или неподвижни обекти (Фиг. 2.6).

Един I / O пин се използва за излъчване на ултразвуков сигнал и след това засича върнатия импулс. Сензорът измерва времето, необходимо за връщане на импулса и връща стойност към микроконтролера като променлива ширина на импулса по същия I / O пин.



Фигура 2.5 Инфрочервен сензор



Фигура 2.6 Ултразвуков сензор

Робко 01 е учебен робот разработен през 80-те [34]. Той представлява антропоморфна ръка с шест степени на свобода. Съставните части на ръката са основа, три стави и хващач. Задвижващите механизми са изпълнени чрез зъбни колела и предавки, а използваните мотори са стъпкови. За управлението на робота е разработен контролер състоящ се от драйвери за управление на стъпковите мотори A4988 и контролер Ардуино за управление на драйверите.

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. Тази дистрибуция има голямо бъдеще, свързано с пренасянето и интегрирането в света. Това дава на ROS една отлична перспектива за развитието и използването на приложения в областта на роботиката, напълно различни по мащаб и сложност.

ROS има голям набор от драйвери за различни видове сензори и за управление на различни механизми, каквито са пакетите за зрението на робота. Също така има и библиотеки, които прилагат алгоритми за решаване на проблеми в областта на роботиката, като правата и обратна кинематична задача, навигация, разпознаване на обекти и т.н. Използвайки всички тези пакети, се намалява драстично времето за разработка и се подобрява степента на успех на все

Главната цел на ROS е да предостави код, който да се използва отново и отново в областта на изследването и развитието на Роботиката. ROS е рамка от процеси (известни още като нодове), която позволява на изпълнимите файлове да бъдат индивидуално проектирани и свободно съчетани по време на тяхното изпълнение. Тези процеси могат да бъдат групирани в пакети и стекове, които могат да бъдат лесно споделяни и разпределени. ROS също така предоставя централизирана система на кода, наречена хранилище, която позволява разпределение на сътрудничеството. Този дизайн, от файлово ниво до ниво общност, дава възможност за независими решения относно разработването и изпълнението на приложения. ки проект.

В подкрепа на тази основна цел за споделяне и сътрудничество, има още няколко други цели на рамката ROS:

Peer-to-peer - система, построена с помощта на ROS, състояща се от редица процеси, извършвани на различни хостове, свързани по време на работа в peer-to-peer топология.

Многоезичен – ROS е проектиран да бъде езиково-неутрален. В момента поддържа четири езика: Python, C++, Lisp и Octave. Има и няколко експериментални библиотеки за Java и Lua.

Инструментно-базиран – ROS съдържа голям брой малки инструменти, които се използват за изграждане и стартиране на различни компоненти в ROS. Тези инструменти изпълняват различни задачи, като например задаване параметрите на конфигурацията,

визуализация, peer-to-peer топологията, автоматично генериране на документацията и много други.

Лесно приложим - ROS е проектиран да бъде толкова лесен, колкото е възможно, така че кодът написан на ROS да бъде използван за други роботизирани софтуерни рамки. За сега ROS е интегрирана с OpenRAVE, Orocos и Player.

Достъпен и с отворен код - Пълният сорс код на ROS е публично достъпен, което допринася за улесняване отстраняването на грешки на всички нива на стека. ROS се разпространява под условията на BSD лиценз, което позволява развитието както на нетърговски, така и на търговски проекти.

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

2.2.1. Файлово ниво

Идеите за файловото ниво са ROS средства, които могат да бъдат срещнати на всеки диск, като:

Пакети – те са основната единица за организиране на софтуера в ROS. Те могат да съдържат ROS процеси (нодове), ROS-зависими библиотеки, набор от данни, конфигурационни файлове или всякакви други неща, за които е удобно да бъдат организирани заедно;

Манифести (manifest.xml) – предоставят метаданните на пакет, включително и информация за неговия лиценз и зависимости, както и специфична информация за езика, каквито са компилиращите флагове;

Стекове – са колекция от пакети с обща функционалност, като например “стека за навигация”;

Стек манифести (stack.xml) – предоставят информация за стека, включително и информация за неговия лиценз и неговите зависимости върху други стекове;

Съобщения (MSG) – описанието на съобщението, съхранено в my_package/MSG/MyMessageType.msg, определя структурата от данните за съобщенията, изпратени в ROS;

Сървизи (SRV) – описанието на сървиза, съхранено в my_package/SRV/MyServiceType.srv, определя структурата от данните на заявката и отговора за сървиза в ROS.

2.2.2. Изчислително ниво

Това ниво осигурява peer-to-peer мрежа от ROS процеси, които обработват данните заедно. Основните концепции в това ниво са нодове, master, parameter server, съобщения, сървизи, топици и bags, като всичките те предоставят данни по различен начин.

Нод – нодовете са процеси за изпълнение на дадена функция. Те са основната структурна и функционална единица за изпълнение в ROS. Всички функции и цялата логика в ROS приложението се изпълняват чрез нодове. Ето например една система за управление на робот се състои от много нодове – един нод управлява лазерния скенер, друг – двигателите,

трети – осигурява графичния изглед на системата и т.н. ROS нодовете се пишат, използвайки клиентските библиотеки като `goscrr` или `gospy`.

Master – той осигурява регистрацията на имената и проверява вече наличните. Без него, нодовете не биха били в състояние да се открият един друг, да разменят съобщения помежду си или да извикват сървизи.

Parameter server – той позволява съхранението на данните по ключ на централно място. В момента той е част от Master-а.

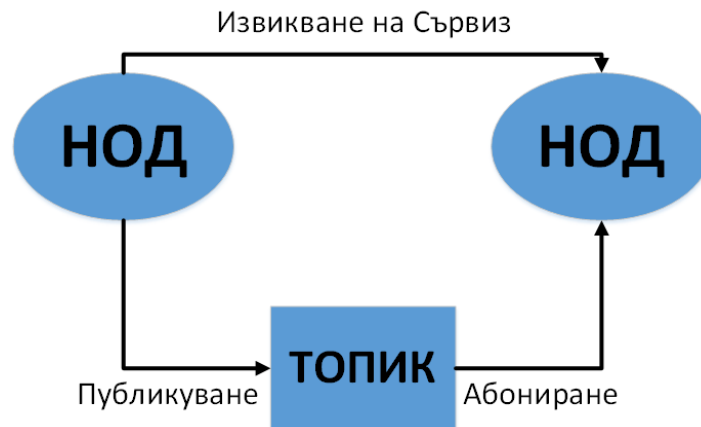
Съобщения – посредством тях нодовете комуникират помежду. Съобщението е проста структура от данни, включваща текстови полета. Те поддържат стандартните примитивни типове (цяло число, число с плаваща запетая, булев израз, и др.), както и масиви от примитивни типове. Съобщенията могат да съдържат и произволно вложени структури и масиви (подобно на `structs` в C).

Топици – те са средство за IPC (Inter-Process Communication), начин да се обменят съобщения с определен формат. Един нод изпраща съобщение чрез публикуването в даден топик. Топикът всъщност представлява име, което се използва за идентифициране на съдържанието на съобщението. Нодът, който се интересува от определен вид данни, ще се свърже към определения топик. Така топиците позволяват да се обособят съобщенията между изпращача и получателя, като и двете страни „знаят” само името на топика, но не един за друг. Това позволява прилагането на изключително мащабируем и гъвкав софтуер и позволява на разработчика да се фокусира върху самата логика и креативността на приложението.

Сървизи – те дават много удобни средства за двупосочна комуникация на съобщенията и лесно изпълнение на клиент-сървър отношенията между нодовете. Сървизите се определят от чифт съобщения – едно за заявката и едно за отговора. Осигуреният нод предлага сървиз под дадено име и клиентът използва сървиза като изпраща съобщение заявка и чака за отговор.

Bags – те са формат за съхранение и възпроизвеждане на данните в съобщенията в ROS. Те са важен механизъм за съхраняване на данни, като например данните от сензорите, чието събиране може да се окаже трудно, но необходимо за разработването и тестването на алгоритми.

Нодовете се свързват с другите нодове директно, като Master-а осигурява проверка на информацията, точно както DNS сървър [36]. Нодовете, които се прикрепят към даден топик, ще изискат връзки от други нодове, които публикуват този топик и ще установят връзката през съгласуван протокол за връзка (Фигура 2.7). Най-често използвания протокол в ROS, се нарича TCPROS, който използва стандартни TCP/IP sockets.



Фигура 2.7 Основна концепция в ROS.

Комуникацията между различните нодове може да се изпълнява чрез TCP/IP [37] връзка по мрежата, която прави възможно взаимодействието между нодове, стартирането на различни компютри през Интернет и логическа/пълна дистрибуция между няколко машини.

ROS изпълнява много удобен формат за определяне на сървизите и съобщенията, изпратени между нодовете, и този формат е езиково-независим. По този начин може да има множество нодове, написани на различни езици за програмиране (Python [38], C++ [39]) и комуникация един с друг за прилагане на цялостна и интегрирана система.

2.2.3. Ниво общност

Това ниво предоставя ресурси, които позволяват на отделни общности да обменят знания и софтуер. Тези ресурси включват:

Дистрибуции – това са колекции от различни версии на стекове на ROS, които могат да се инсталират. Те имат същата роля като дистрибуции на Linux: правят по-лесно инсталирането на целия софтуер и поддържат всички последователни версии на софтуера;

Хранилище – ROS разчита на централизирани хранилища на код, където различните институции могат да развият и да предоставят своите собствени софтуерни компоненти за работи;

ROS Wiki - е основният форум за документирание на информацията за ROS. Всеки може да си направи профил и да допринесе със своята документация, като прави и корекции или актуализации, пише уроци и други;

Пощенски списъци - е основният канал за комуникация, както и форум, където може да се задават въпроси за ROS;

Отговори - сайт, където може да се получат отговори на въпросите, свързани с ROS;

Блог - осигурява редовни актуализации, снимки и видео.

2.3. Навигация и локализация

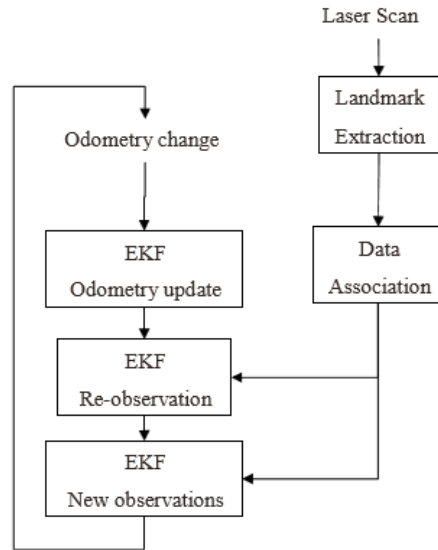
Навигационните системи за мобилни роботи се нуждаят от достатъчно надеждно измерване на текущото местоположение на робота и прецизна карта на зоната за навигация. Тези системи се разделят на две нива на управление: глобално планиране на път и

управление на локалното движение. Планирането на път се нуждае от модел или карта на средата, за да определи и изчисли геометричните точки на пътя, така че мобилния робот да се придвижи от стартова позиция до целевата позиция. Докато локалното движение обикновено използва информация от сензори за да определи придвижване, което ще осигури избягване на сблъсъци с непознати препятствия или такива препятствия, които са се променили локацията.

Една напълно функционираща навигационна система трябва да включва в себе си локална и глобална навигационна система: глобалната система преизчислява целия път и постоянно търси нови пътища, когато се пови несъответствие с картата. Локалната навигация използва сензорите на робота за да разпознае и избегне непредвидени препятствия. Мобилния робот изпълнява алгоритъм, който го кара да следва оптималния път, като следи геометричните точки на пътя. Ако се появи препятствие на пътя на робота, се активира друг алгоритъм (алгоритъм за избягване на препятствия), който дава възможност за заобикаляне на препятствието по най-краткия възможен път ако това е възможно или преизчислява друг оптимален път за да достигне целта.

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 трансляционното и ротационното ускорение директно върху пространството.

За реализиране на навигационната система на робота използваме метода SLAM (Simultaneous Localization and Mapping) – едновременна 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. Понеже одометрията на робота (която ни дава позицията на робота) в повечето случаи отчита множество грешки, не можем да разчитаме изцяло на нея. Вместо това можем да използваме сканиране на средата с лазер или друг вид сензор за да коригираме правилно позицията на робота. Това е изпълнено като се извличат особеностите на средата и се преповтарят сканиранията, докато робота се придвижва. В основат на процеса е разширен филтър на Калман (ЕКФ). Той е отговорен за опресняването на позицията, на която робота си мисли, че се намира, използвайки за ориентир характеристиките на средата. Филтъра на Калман наблюдава измерванията на промяната в позицията на робота, също така и промяната на тези маркери, които вижда в средата.



Фигура 2.8. Принципно схема на SLAM процесът

Когато одометрията се променя при движението на робота, промените отнасящи се за новата позиция на робота се обновяват във филтъра на Калман, чрез блока опресняване на одометрията. Разположението и данните за маркерите в средата се извличат на база на новата позиция на робота. След това роботът се опитва да направи сравнение между тези маркери и маркерите от предходното сканиране. Тогава ново определените маркери се използват за да се презапише новата позиция на робота в EKF.

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. Ако навигационния пакет не получава данни от сензорите на робота, тогава той ще се движи на сляпо и има голяма опасност да се удари в нещо или някой. 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. Въпреки голямото разнообразие от сензори, навигационния пакет приема само определени типове данни публикувани от `sensor_msgs/LaserScan Message` или `sensor_msgs/PointCloud Message`. Всеки сензор може да се конфигурира, така че информацията от него да се преобразува в някой от двата типа съобщения.

За работи с лазерни скенери, ROS предоставя специален тип съобщение от пакет със съобщенията, наречен `LaserScan`. В това съобщение се съдържа информацията от сканиранията на сензора. Този тип съобщения позволяват лесно програмиране на системата, като може да се работи виртуално с всеки лазер, докато постъпващата информация от скенера може да се форматира, така че да се събира в съобщението. Ние използваме сензора Kinect като лазерен скенер. Пакетът който предоставя възможността на ROS да се свързва и обменя данни с дълбочинни сензори се нарича `orenni_camera`. Топиците, от които се нуждаем са за дълбочинния сензор на Kinect-a:

Дълбочинна камера

- Публикува данни само, когато `depth_registration` е `false`-истина (OpenNI регистъра е забранен) и съдържа два топики: `depth/camera_info` (`sensor_msgs/CameraInfo`) – калибриране на камерата и метаданни.

`depth/image_raw` (`sensor_msgs/Image`) – извлича първично изображение от устройството. Изображението се записва в променлива от тип `uint16` и представлява измерените стойности на дълбочината в милиметри – регистрираната дълбочинна камера (заедно с RGB камерата).

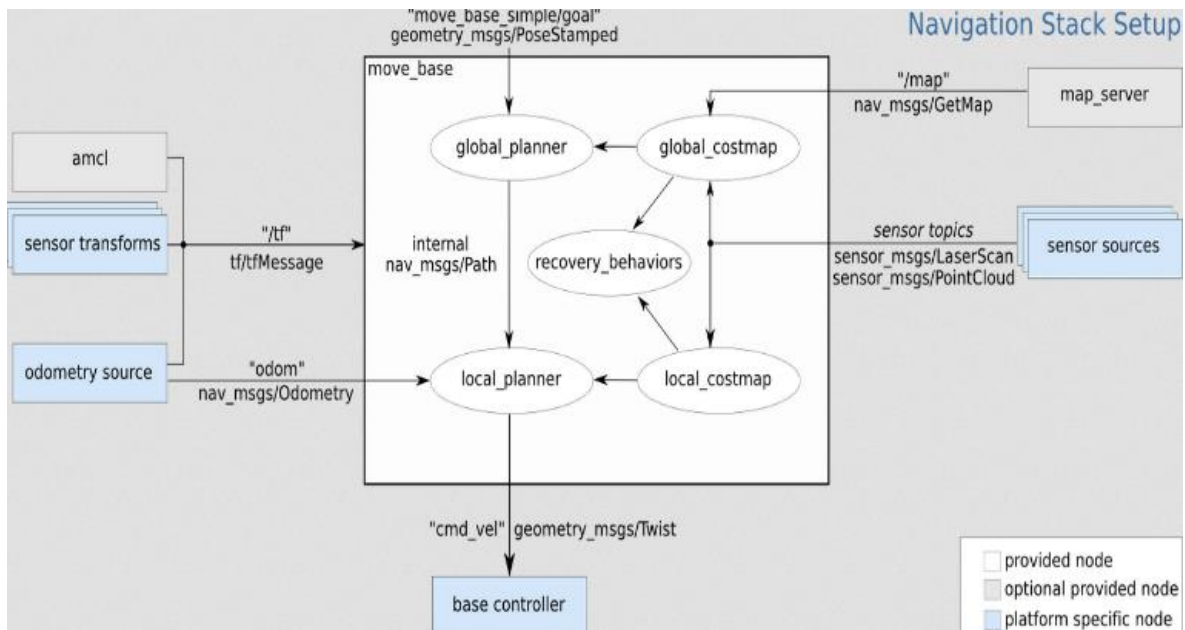
Публикува се директно от драйвера, само когато `depth_registration` е `true`-истина (OpenNI регистъра е разрешен) и съдържа следните топици:

`depth_registered/camera_info` (`sensor_msgs/CameraInfo`) – калибриране на камерата и метаданни. Същите като `rgb/camera_info`, но синхронизирани със `depth_registered/image_raw`.

`depth_registered/image_raw` (`sensor_msgs/Image`) – извлича първично изображение от устройството. Изображението се записва в променлива от тип `uint16` и представя дълбочината в милиметри.

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. Като необходими предусловия за изпълнение на навигационния пакет, робота трябва да изпълнява следните условия: да има инсталиран ROS, да притежава „tf“ трансформиращо дърво и да се четат данните от сензорите, като се използват правилните видове съобщения в ROS. Също така навигационния пакет трябва да се конфигурира според специфичните особености на мобилната платформа на робота – тип платформа и динамични характеристики.



Фигура 2.9. Принципна схема на работата на Навигационния пакет

На фигура 2.9 е показана блоковата схема, която илюстрира принципа на работа и конфигурацията на навигационния пакет. Белите компоненти са задължителни компоненти и са имплементирани по подразбиране, сивите компоненти са препоръчително, но незадължителни и сините компоненти се конфигурират и настройват според специфичните характеристики на мобилната платформа на всеки робот.

Навигационния пакет изисква от робота да публикува информацията за връзките между координатните рамки като се използва библиотеката „tf“ [50]. Тази библиотека позволява данните от сензорите или каквито и да се други данни с печат, да се транспортират в тяхната оригинална рамка посредством мрежата, както и да се запишат в оригинален вид. Когато алгоритъм иска да използва данни в координатната рамка, която е най-подходяща за изчислението, може да потърси в библиотеката tf за трансформацията от координатната рамка на щампованите данни до желаната координатна рамка. Използването на получената трансформация за трансформиране на данните по време на нужда предотвратява ненужните междинни трансформации, спестявайки както изчислителното време, така и деградацията на данните, дължащи се на многократна обработка с потенциални закръглени проблеми.

За удобство, за да получите най-новите налични данни, заявката в момент нула ще върне най-новото общо време в заявените стойности. Ако такова време не съществува, процесът който прочита данните „Listener“ ще направи изключение, по същия начин, както ако не е налице време, което не е било зададено извън кешираната история.

2.3.3. Информация от одомерията

Одомерията е много важен метод, използван за изчисляване на текущата позиция на мобилни системи. Колкото по точни са одомеричните данни толкова по добри са точността и прецизността на позициониране. Одомеричните данни се използват за да се определи позицията на робота спрямо стартовата локация. За да измерваме изминатото разстояние от робота използваме енкодери монтирани на електромоторите на робота. Те броят направените обороти и по този начин може да се определи изминатото разстояние на всяко от активните колела на робота. Тези сензори са в основата на изчисляване на одомерията, като трябва да се добавят и някои допълнителни параметри от мобилната платформа като диаметър на колелата, междуосие и резолюция на енкодерите. Реализиран е алгоритъм за изчисляване на одомеричните 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:

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

- Определяне на сегашната позиция

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

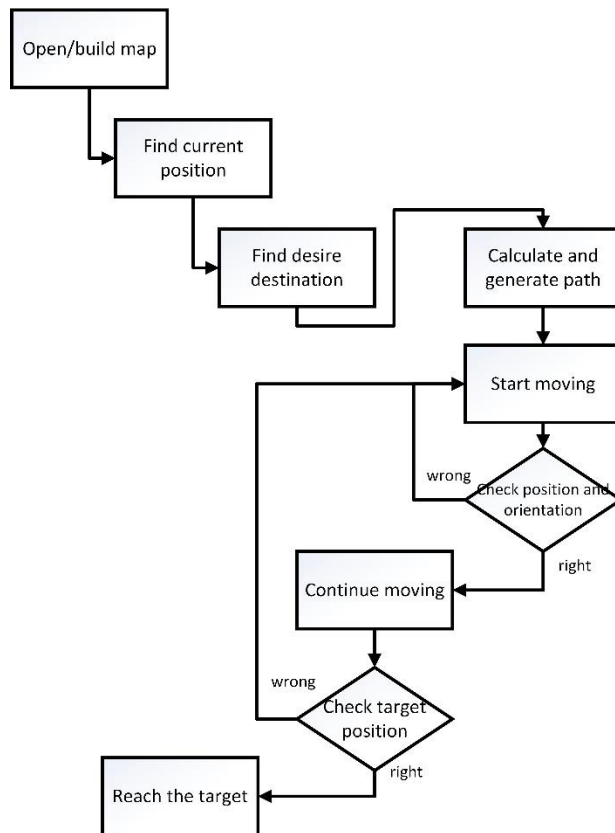
$y := y + \text{distance} * \sin(\theta)$

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 (`vx`, `vy`, `vtheta`) \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.



Фигура 2.10. Алгоритъм за оптимизиране на навигацията

Този алгоритъм показва начина на работа на навигацията на робота. В началото на алгоритъма е зареждане на картата на средата или създаване на нова карта ако средата е непозната. След като има карта на помещението се активира процес на намиране на локализацията на робота. Тук позицията на робота се задава от оператор за да постигне по-добра работа. След като робота знае своята позиция можем да му задаваме желани цели или позиции, които да достигне. Тези цели се задават директно върху използваната карта. След

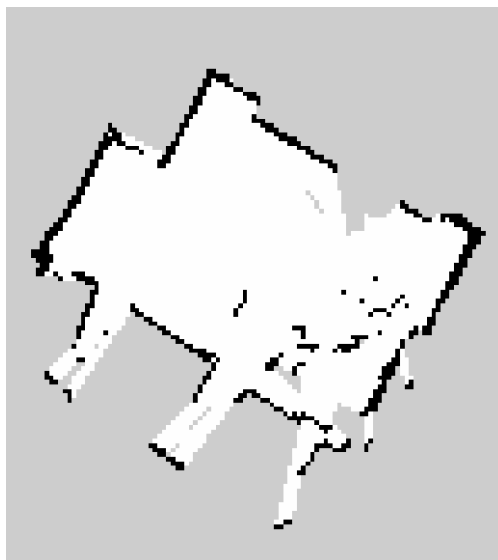
като има зададена дестинация навигацията изчислява пътя до там. Алгоритъма за изчисляване на път е използвания в изкуствения интелект алгоритъм A*, който винаги намира най-кратък път до целта.

Когато роботът започне да се движи по зададената траектория, навигацията проверява неговата позиция и ориентация в всеки един момент. Това е необходимо поради вероятността от отклонение от траекторията на движение, което може да се дължи на грешки в одометрията, неравности по терена или промяна в средата на робота, например нещо застава пред робота. Тази проверка в реално време се повтаря докато робота не достигне до крайната си цел. При наличие на препятствия или отклонение от правилния път навигацията преизчислява пътя за придвижване. Когато робота достигне крайната цел той спира и чака докато не се зададе следваща.

Веднъж след като е определена локацията на робота, навигацията знае във всеки следващ момент къде се намира робота и каква е неговата ориентация. Този алгоритъм позволява избягване на препятствия, защото непрестанно сканира средата. Много важна роля за движението на платформата изпълнява одометрията. Ако нейните параметри не се изчислят правилно роботът много трудно, неточно и бавно ще се придвижва до целта. Използвайки данните от одометрията навигацията може да управлява моторите мобилната платформа. Навигацията изчислява и праща команди за подаваната мощност към всеки мотор, така че да се завърти на определена позиция. Така навигацията работи точно и с относително малка грешка.

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 показан карта на едно от сканираните помещения, направена в ръчен резим на сканиране – управляваме робота с джойстик.



Фигура 2.11. Карта направена от робота.

Картата се генерира като роботът обхожда и сканира желаната среда. За създаване на самоата карта се грижи нода „slam_gmapping”. След като имаме карта на определено помещение или пространство можем да я използваме по всяко време, когато роботът е на това местоположение. Без наличие на такава карта, навигационният пакет не може да локализира местоположението на робота. Неудобството тук е необходимостта всеки път да се избира карта от базата данни на робота и ръчно да се посочва началната позиция на локацията му. Поради това се налага разработване на алгоритъм за автономно разпознаване на помещенията и респективно локализиране. Този алгоритъм е описан в следващата точка от тази глава.

2.3.6. Локализация и разпознаване на помещения и сгради

Понеже търсим приложение на мобилен робот в помещения на закрито, където разстоянията са сравнително малки, има стени, които пречат на сензорите за глобална локализация и се търси относително голяма точност на позициониране използваме методите за релативна локализация, като добавяме сензора Kinect, който със своя инфрачервен сензор за дълбочина ще намали грешките от одомерията и ще прави измервания в реално време.

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. Ако намери съвпадения той ще знае стаята и сградата, където се намира, ако не намери съвпадение ще се активира алгоритъм, за запис на нова карта на помещението и роботът ще започне да сканира.

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 работи, чрез който можем да създаваме двумерни картни на помещенията. За създаване на тези карти се използват данните от лазерния скенер и позицията на робота. Поради това мобилния робот трябва да разполага с хоризонтално разположен лазерен скенер и добре изчислена одомерия. „slam_gmapping“ възела ще преобразува всяко сканиране в одометрична рамка на „tf“ библиотеката. ROS предоставя специален тип съобщение в пакета sensor_msgs, наречен LaserScan.

Друг пакет, който играе важна роля в управлението на робота е навигационния пакет на ROS [52]. Този пакет използва същите данни както пакета gmapping – данните от одомерията и данните от лазерния скенер. Работата на навигационния пакет обаче е да генерира управляващи команди за движение на робота. За да работи правилно навигационния пакет, той трябва да се конфигурира според параметрите на робота: одометрични характеристики, инициализиране на данните от сензорите, конфигуриране на „tf“ библиотеката и описание на управляващия контролер.

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.

Когато няма намерени съвпадения, се активира процедура на сканиране на новото помещение. Тази процедура стартира автоматично след изчакване от 10 секунди, при които потребителя може да я отмени.

2.3.7.2. Режим на запис на нова карта.

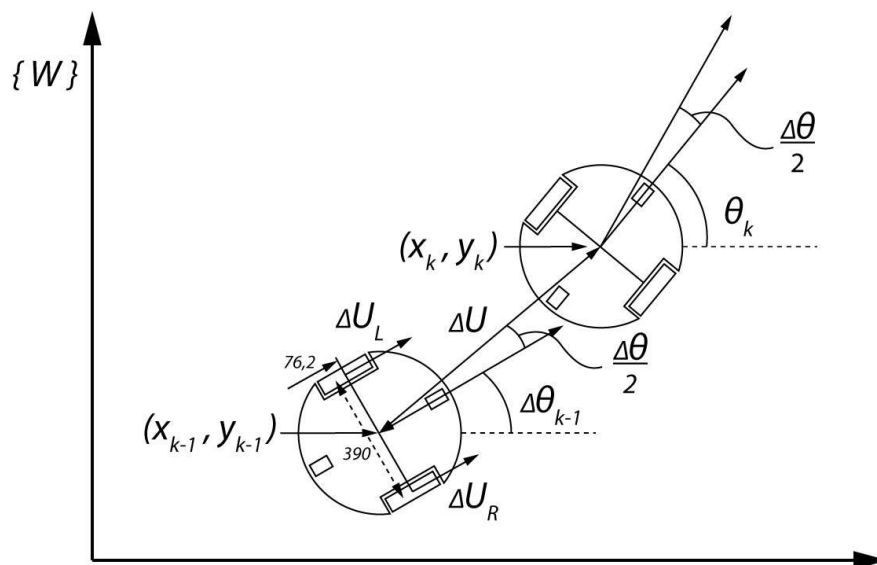
Когато този режим се активира робота обхожда помещението, в което се намира като в това време е активиран пакета gmapping заедно с навигационния пакет. Робота се движи докато не получи пълна картина на помещението, след което се връща в изходната си позиция. Ново-сканираната карта се добавя към базата 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

Използвайки прости геометрични уравнения, е лесно да се изчисли моментното положение на робота спрямо известната стартова позиция. Това изчисление се нарича одометрия. Одометрията изчислява относителното хоризонтално преместване на робота и промяна в ориентацията, като функция на нарастващото хоризонтално преместване на задвижващите колела (Фиг. 2.12).

Одометрията се основава на уравнения, които се изчисляват сравнително лесно и използват данни от енкoderите на всяко от колелата. Все пак одометрията разчита на представянето, че оборотите на колелата могат да се преобразуват в линейно преместване спрямо пода. Това представяне има ограничения. Пример за тези ограничения е приплъзване на колело: ако колелото се плъзне по мокър под, тогава енкoдера ще регистрира

завъртанията на колелото, но тези обороти няма да съответстват на реално изминатото разстояние от колелото.



Фигура 2.12. Параметри на одомерията.

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:

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

Одометричните грешки се делят на две категории:

- Систематични грешки: разлика в диаметъра на колелата, неточности в механиката на платформата;
- Несистематични грешки: боксуване, приплъзване, изпускане на обороти и др.;

2.4.1. Описание на систематичните и несистематичните грешки

Систематичните грешки са тези грешки, които се дължат на дефекти и разлики в самата мобилна платформа на работа. Тези грешки обикновено са константни и могат да се коригират, като се приложат математически модели, без да е необходимо да се правят допълнителни измервания. Примери за систематични грешки са:

- Нееднакви диаметри на задвижващите колела;
- Несъосност на колелата;
- Промени в ефективното сцепление на платформата;
- Ограничение в резолюцията на енкодерите;

- Ограничена честота на дискретизация на енкодерите.

Несистематичните грешки са тези грешки, които са причинени от взаимодействието на робота с непредвидими особености на околната среда. Например нарушения по повърхността на пода, като подутини, пукнатини или остатъци от боклук, може да са причина за повече завъртания на колелото, отколкото е предвидено, защото колелото върви нагоре или надолу по неравността. Примери за несистематични грешки:

- Движение по неравен под;
- Преминаване през някакъв предмет на пода;
- Боксуване на колелото поради: хлъзгав под, голямо ускорение, рязко завиване, загуба на контактна точка с пода.

Систематичните грешки се проявяват най-често при движение по равни и чисти настилки на закрито. При движение по такива повърхности тези грешки оказват по-голямо влияние на одометрията отколкото несистематичните. Въпреки това при движение по неравни повърхности несистематичните грешки може да бъдат повече. Нашият робот е предназначен да се движи на закрито, като коригираме и двата типа одометрични грешки за постигане на добра точност. Измерването на одометричните грешки е процес, който включва множество експерименти и анализ на техните резултати [53].

2.4.2. Коригиране на систематичните грешки.

Систематичните грешки са специфични за всяко колело и обикновено не се променят по време на движение. Поради това одометрията може да се подобри като цяло, чрез измерване на индивидуалния принос на най-влиятелните източници на грешки, след което да се противодейства на тяхното въздействие с помощта на софтуер.

Прилагайки SLAM процеса при управление на робота, ние използваме одометричните данни за да се изчислява траекторията на движение на робота. Намалването на систематичните грешки води до по-добро изчисляване на траекторията на движение и намаляване на отклоненията, като се подават управляващи сигнали с коригирани параметри. 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 на право. 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 колело. Също така можем да дефинираме:

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

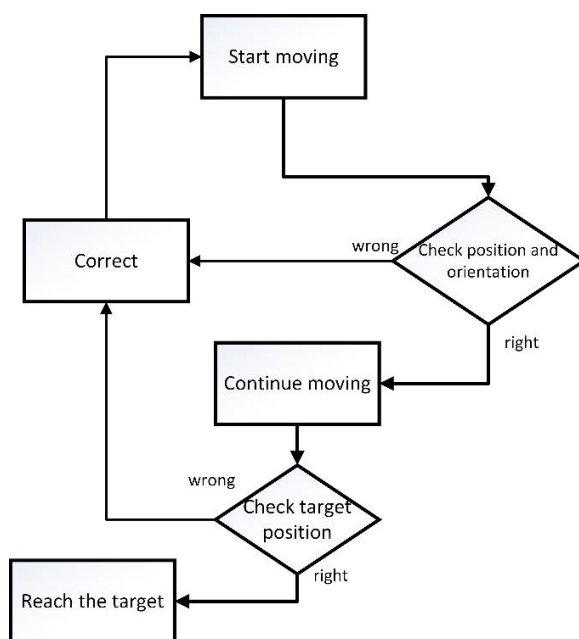
Където b е междуосието на мобилната платформа.

Коригирането на систематичните грешки може да се раздели на два основни етапа: измерване на грешките, чрез експерименти и анализиране на получените данни с цел коригиране. Измерването на отклоненията при движение е реализирано като се програмира роботът да се движи в квадрат с дължина 3м. След определен брой обиколки на квадрата се отчитат отклоненията при движение по правите и при завиване при ъглите. Наблюдавайки отклоненията на робота при преминаване от ъгъл до ъгъл можем да определим отклонението в абсолютната позиция. Така можем да сравним желаната позиция и ориентация с реалната.

2.4.3. Коригиране на несистематичните грешки.

Несистематичните грешки са трудни за корекция, понеже са не повтарящи се и могат да бъдат предизвикани от множество различни фактори. За да коригираме отклонението на робота при наличие на несистематична грешка използваме данните от сензорите на робота. Основният сензор, който следи за отклонение от зададена траектория на движение е сензора за измерване на дълбочина, вграден в Kinect сензора. Този сензор измерва в реално време разстоянията до ориентируващи точки от средата, в която се движи робота и при наличие на отклонения изпраща коригиращи сигнали към управлението на моторите.

Понеже следено за отклонение се случва в реално 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. Това коригиране е активно докато не се постигне възстановяване на правилната ориентация и се навлезе в зоната на траекторията на навигацията.

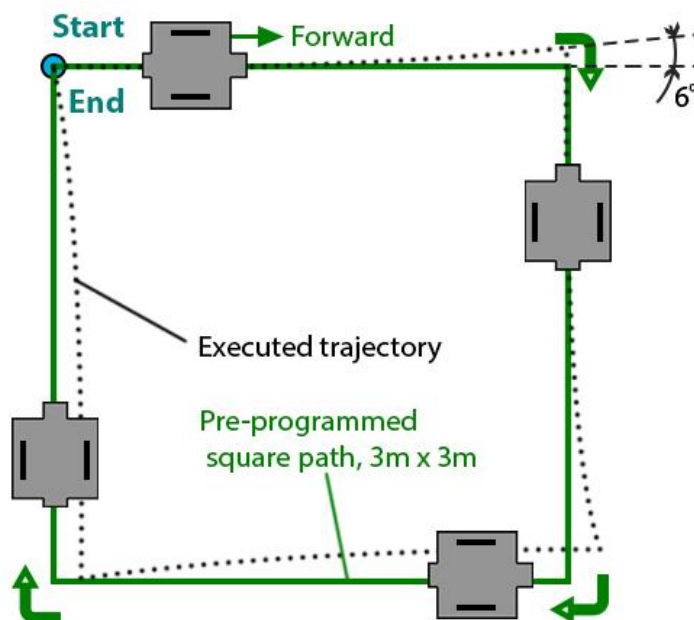


Фигура 2.13. Алгоритъм за коригиране на несистематичните грешки в реално време.

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

За да подобрим допълнително изчисленията от одометрията трябва да намалим още систематичната грешка. Понеже робота е оборудван с гумени колела, които се пълнят с въздух има голяма вероятност систематичната грешка да се променя според налягането на въздуха в колелата или от различие в самите гуми на колата. Междуосието се дефинира като разстоянието между контактните точки на две колела от диференциално задвижван робот. Това разстояние трябва да бъде известно за да се изчисли броят на тиковете на енкодерите, които представляват завъртането на колелото. Изменението на активното междуосие е причинено от факта, че гумените колела взаимодействат с пода не в една точка, а в контактна зона.

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



Фигура 2.14. 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. Този комбиниран сензор може да ни дава данни за ориентацията на робота, ускорението при движение и спиране. Използването на този сензор ни дава завиването и ориентацията на робота без да се налага да ги изчисляваме от данните на енкодерите. Така можем да сравняваме данните от IMU сензор и тези от енкодерите и по този начин да намалим грешката на робота при движение и да я коригираме в реално време.

Също така можем да използваме факта, че знаем подадените команди към робота. По този начин във всеки един момент ще знаем как роботът ще се премести или поне ще

очакваме да се премести по този начин. Знаем какви трябва да се стойностите на линейните и ъгловите скорости и ускорения. Благодарение на IMU и енкoderите ще знаем какви са тези стойности в реално време. С някои техники за машинно обучение ние адаптираме робота към текущата земна повърхност. Това води до стабилно и предвидимо движение на нашата интелигентна система. Като се има предвид всичко това, в повечето случаи можем да открием аномалии в движението на робота - колебания на колелата. Това ни позволява не само да реагираме и да регулираме силата на засегнатото колело, но и с помощта на някои евристики, да коригираме одометрията.

2.5. Компютърно зрение

2.5.1. Цветни пространства в OpenCV.

Сиво: Цветно пространство, което ефективно премахва цветната информация, превеждаща на нюанси на сивото: това цветово пространство е изключително полезно за междинна обработка; Като разпознаване на лица.

BGR: Цветното синьо-зелено пространство, в което всеки пиксел е три елементна матрица, всяка стойност представляваща синьо, зелено и червено.

HSV: hue-оттенькът е цветен тон, saturation-наситеността е интензивността на цвета и value-стойността представлява неговата тъмнина (или яркост в противоположния край на спектъра).

2.5.2. Трансформация на Фурие.

Всички вълни са само сумата от обикновените синусоиди на различни честоти. С други думи, вълните, които наблюдавате навсякъде около вас, са сумата от други вълнови формати. Тази концепция е невероятно полезна при манипулиране на изображения, защото ни позволява да идентифицираме региони в изображения, където сигналът (като например пикселите на изображението) се променя много, а регионите, където промяната е по-малко драматична. Концепцията за трансформация на Фурие е в основата на много алгоритми, използвани за общи операции за обработка на изображения, като например откриване на ръба или откриване на линия и форма.

2.5.3. Високочестотен и нискочестотен филтър.

Високочестотен филтър (HPF) е филтър, който изследва регион на изображение и увеличава интензивността на определени пиксели въз основа на разликата в интензивността с околните пиксели.

Ако HPF увеличава интензитета на пиксела, предвид разликата му със съседите си, нискочестотният филтър (LPF) ще изгладя пиксела, ако разликата с околните пиксели е по-ниска от определения праг. Това се използва при отнемане и замъгляване. Например, един от най-популярните филтри за размиване / омекотяване, Гаусово замъгляване, е нискочестотен филтър, който намалява интензитета на високочестотните сигнали.

2.5.4. Намиране на градиент на интензитета на изображението

Изгладеното изображение се филтрира със Sobel ядро в хоризонтална и вертикална посока, за да получи първата производна в хоризонтална посока (G_x) и вертикална посока (G_y). От тези две изображения можем да намерим градиент на ръба и посока за всеки пиксел, както следва:

$$Edge_Gradient(G) = \sqrt{G_x^2 + G_y^2};$$
$$Angle(\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right);$$

Направлението на градиента винаги е перпендикулярно на ръбовете. Тя е закръглена до един от четирите ъгъла, представляващи вертикални, хоризонтални и две диагонални посоки. OpenCV поставя всичко по-горе в една функция, `cv2.Canny()` [56].

2.5.5. Хистерезис: Последната стъпка. `cv2.Canny` използва два прага (горен и долен):

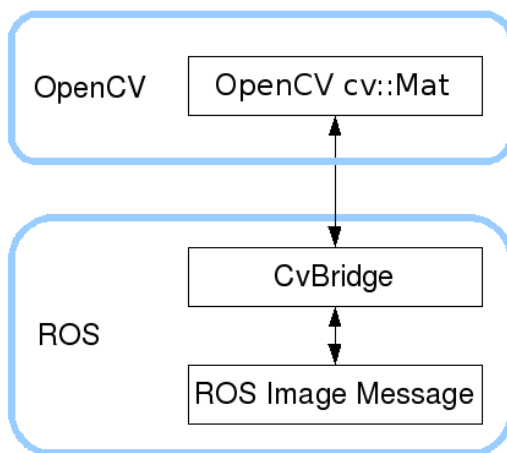
- Ако пикселният градиент е по-висок от горния праг, пикселът се приема като край
- Ако стойността на пикселния градиент е под долния праг, то тя се отхвърля.
- Ако пикселният градиент е между двата прага, то той ще бъде приет само ако е свързан с пиксел, който е над горния праг.

2.5.6. 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.

ROS преминава около изображения в собствения си `sensor_msgs` / формат на съобщението за изображения, но много потребители ще искат да използват изображения във връзка с OpenCV. `CvBridge` е библиотека на ROS, която осигурява интерфейс между ROS и OpenCV. `CvBridge` може да се намери в пакета `cv_bridge` в стека `vision_opencv` [57].



Фигура 2.15. 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. Въпреки че CameraInfo съдържа цялата информация, необходима за отстраняване на необработено изображение и точки на проекта върху него, препоръчително е да използвате тази библиотека, тъй като извършването на тези операции правилно през пространството на всички опции на камерата може да не е тривиално.

2.5.7. 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_DISPARIITY_MAP - несъответствие в пикселите (CV_8UC1);
 - o CV_CAP_OPENNI_DISPARIITY_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 получим карта на дълбочината от сензора за дълбочина, използваме метода VideoCapture::operator. За получаване на няколко карти с данни, използваме методите VideoCapture::grab and VideoCapture::retrieve.

За настройка и получаване на данни генерирани от сензорите използваме съответно VideoCapture::set and VideoCapture::get методите.

Тъй като се поддържат два вида генератори на данни на сензора (генератор на изображения и генератор на дълбочина), има два флага, които трябва да се използват за настройка / получаване на собственост на необходимия генератор:

- CV_CAP_OPENNI_IMAGE_GENERATOR – знак за достъп до свойствата на генератора на изображения.
- CV_CAP_OPENNI_DEPTH_GENERATOR – знак за достъп до свойствата на дълбочинния генератор. Тази стойност на флага се приема по подразбиране, ако нито една от двете възможни стойности на собствеността не е зададена.

Както Kinect, така и стерео камерата са свързани към USB портовете на компютъра на работа. Докато Kinect търси обект, стерео камерата е в режим на готовност. Когато обектът се разпознае и роботът се придвижи до него на определено разстояние, компютърът включва стерео камерата и започва да управлява роботизирания манипулатор. Процесът на управление на ръката е описан в следващата глава.

За да разпознаем обектите, получаваме краищата на сцената. След това системата търси конкретни форми. Ако има подобни форми, системата проверява формата на същия обект по цвят. Така че за разпознаване на ръба използваме детекция на ръба на Canny с два прага. За преобразуването на цветовете преобразуваме RGB видео сигнала в HSL изображение и извличаме само един канал, L ще бъде обработен. Предимствата от използването на един канал вместо три канала са намаляването на времето за обработка и сложността. 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.

2.5.8. 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. Те включват: Придобиване: откриване и проследяване на лицеви образни пластири на изображения в динамична сцена, Нормализация: сегментиране, привеждане в съответствие и нормализиране на изображенията на лице и разпознаване: представяне и моделиране на лицевите изображения като идентичности и асоцииране на нови изображения с известни модели [59].

Как анализираме изображение и как го кодира мозъкът? 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. Тъй като не виждаме света като разпръснати части, нашата визуална кора трябва по някакъв начин да комбинира различните източници на информация в полезни модели. Автоматичното разпознаване на лица е свързано с извличането на онези смислени елементи от изображението, поставянето им в полезно представяне и извършването на някаква класификация върху тях.

Разпознаването на лица въз основа на геометричните особености на лицето е може би най-интуитивният подход към разпознаването на лица. 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,). Разпознаването е извършено чрез изчисляване на евклидовото разстояние между характеристичните вектори на дадена сонда и еталонното изображение. Този метод е силен срещу промените в осветлението по своята същност, но има огромен недостатък: точната регистрация на маркерите е сложна, дори при най-съвременните алгоритми. Някои от най-новите разработки на геометричното разпознаване на лица са извършени в [62]. Използван е вектор с 22 размери и експерименти върху големи набори от данни показват, че геометричните характеристики сами по себе си не носят достатъчно информация за разпознаване на лица.

2.5.8.1. 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 зареждаме 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()
```

2.5.8.2. 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 възнамерявате да добавите свои собствени алгоритми, добре е да добавите уникален префикс към алгоритмите си, за да ги различите от другите алгоритми..
- Настройване / извличане на параметри на алгоритъм по име. Ако сте използвали функцията за заснемане на видео от модул `OpenCV highgui`, вероятно сте запознати с `cvSetCaptureProperty ()`, `cvGetCaptureProperty ()`, `VideoCapture :: set ()` и `VideoCapture :: get ()`. Алгоритъмът предлага подобен метод, където вместо идентификационния

номер на цялото име вие определяте имената на параметрите като текстови низове. Вижте Алгоритъм :: комплект () и Алгоритъм :: get () за подробности.

- Четене и записване на параметри от / към XML или YAML файлове. Всеки производствен алгоритъм може да съхрани всички свои параметри и да ги прочете обратно. Не е необходимо да го въвеждате отново всеки път.

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 ())

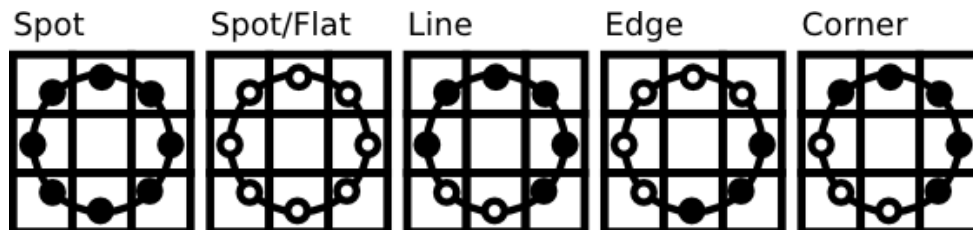
По-формално описание на оператора за локални бинарни шаблони (LBP) може да бъде дадено като:

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

, $c(x_c, y_c)$ 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. Всъщност, авторите са успели да се конкурират с най-съвременните резултати за класификацията на текстурата. Скоро след публикуването на оператора бе отбелязано, че фиксираната съседна знакова функция не успява да кодира подробности, които се различават по мащаб. 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 улавят следните квартали:



Фигура 2.16. 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:

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

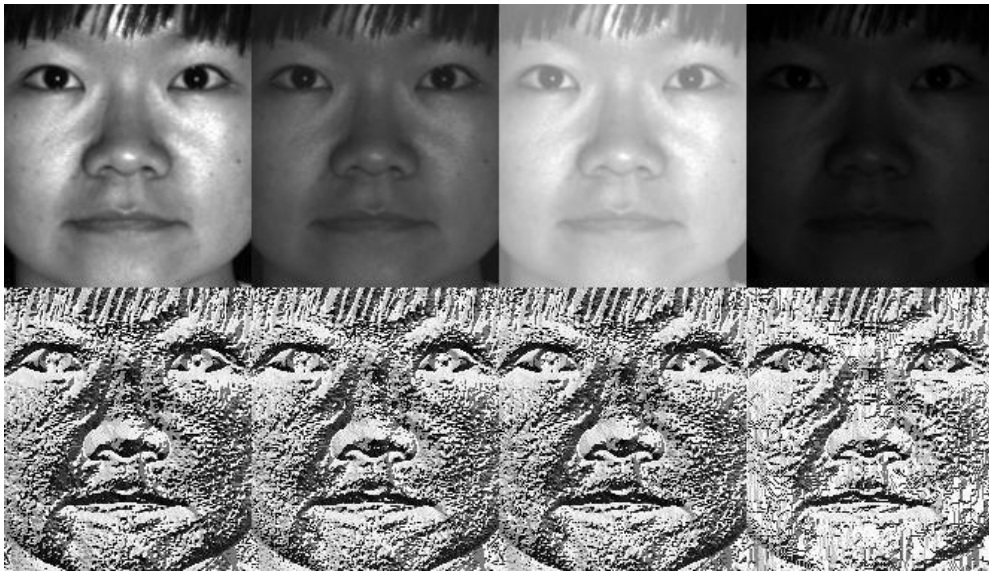
$$y_p = y_c - R \sin\left(\frac{2\pi p}{P}\right)$$

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 (за да видите как изглежда LBP изображение!):



Фигура 2.17. LBP изображение след обработка.

Следващите редове създават модел LBPН за разпознаване на лица и го обучават с изображенията и етикетите, четени от дадения CSV файл. LBPНFaceRecognizer използва разширени локални двоични шаблони (вероятно е конфигурируем с други оператори на по-късен етап) и има следните стандартни стойности: радиус = 1, съседи = 8, grid_x = 8, grid_y = 8. Така че ако искаме LBPН FaceRecognizer използвайки радиус от 2 и 16 съседи, извикваме фабричния метод с: `cv::createLBPНFaceRecognizer(2, 16)`. И ако искаме праг (напр. 123.0) го извикваме с неговите стойности по подразбиране:

```
cv::createLBPНFaceRecognizer(1,8,8,8,123.0).
```

```
Ptr<FaceRecognizer> model = createLBPHFaceRecognizer();
```

```
model->train(images, labels);
```

Следният ред предсказва етикета на дадено изображение на тест:

```
int predictedLabel = model->predict(testSample);
```

За да се получи сигурност на предсказването извикваме модела с:

```
int predictedLabel = -1;
```

```
double confidence = 0.0;
```

```
model->predict(testSample, predictedLabel, confidence);
```

```
string result_message = format("Predicted class = %d / Actual class = %d.", predictedLabel,  
testLabel);
```

```
cout << result_message << endl;
```

ГЛАВА 3. Сервизни работи за специфични цели.

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

Дистанционното управление в нашия слушай е разделено на два типа. Единия тип е управление чрез използване на джойстик, като за обратна връзка се използва видеото от камерата на робота. Другия вариант е полу-автономно управление, като тук се използва системата за навигация и оператора задава желаните места за инспекция.

При първия тип управление трябва да се реализира система за директна комуникация оператор-робот. Важно при това управление е комуникацията да е бърза и да няма забавяне при изпращане/приемане на пакети с данни. Командите от джойстика трябва да са с голяма разделителна способност за да се получи равномерно и сигурно управление на мобилната платформа. Видео сигналът от камерата трябва да е с необходимата разделителна способност, така че оператора да може да се ориентира добре в средата около робота. Този тип управление е подходящ, както за работа на закрито, така и за работа на открито. Понеже се работи от определена дистанция трябва да се предвидят комуникационни модули, които да имат необходимия обхват.

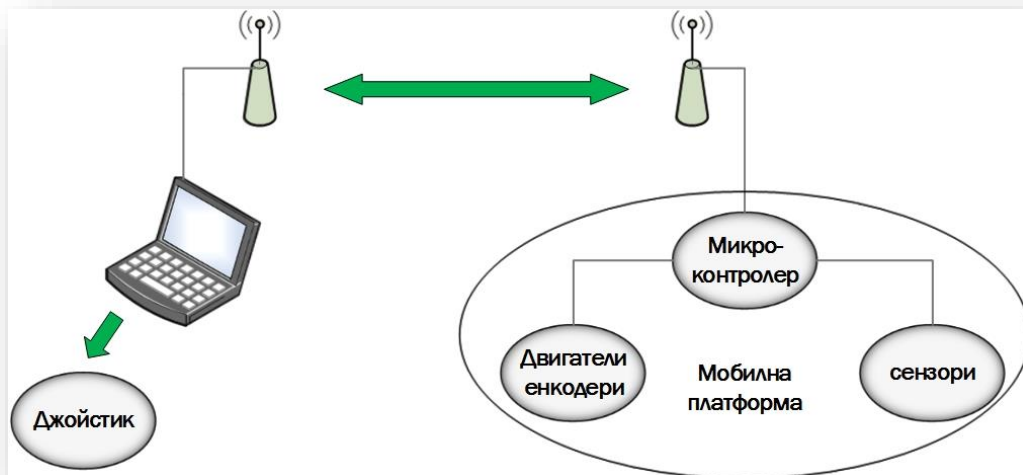
Втория тип управление – полу-автономно изисква значително по-голям ресурс. Трябва да се използват допълнителни сензори, както и да се обработват по-голям обем данни. Тук обаче се намалява човешката грешка, понеже робота сам се движи до зададените цели. Реализираната навигационна система за нашия робот, може да се използва за реализира на този тип управление. Тя може да работи на закрито и на открито. Като недостатък при работа на открито е малкия обхват на Kinect сензора и неспособността му да работи коректно при наличие на голяма осветеност от слънчева светлина. Поради това е препоръчително да се замени с лазерен скенер пригоден за работа на закрито. Така значително ще се подобри работата на системата за сметка на скъпата цена на лазерния скенер.

3.1.1. Управление на мобилната платформа.

Управлението на мобилни работи е сложен процес, при който се включват всички части на мобилната платформа – двигатели, сензори, комуникационни модули и други. За всеки отделен елемент или блок от робота се прави управление в зависимост от това как искаме да управляваме системата, какви са нейните механични характеристики и хардуерни възможности.

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

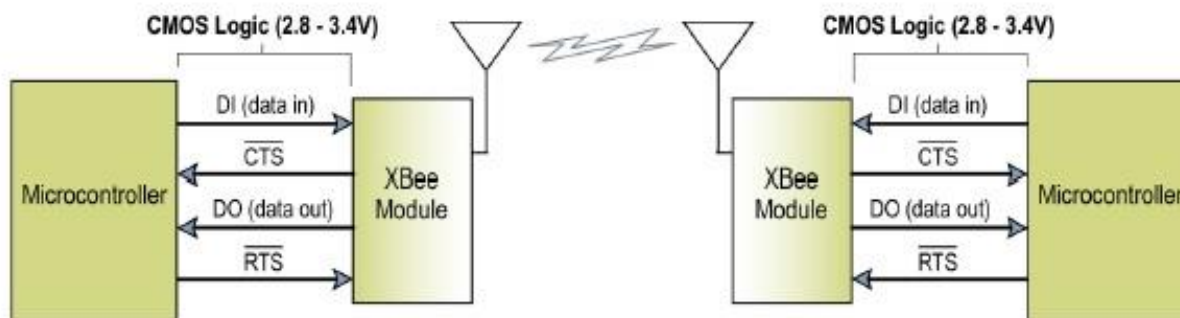
Схемата на фигура 3.1 показва как ще се осъществява управлението на робота. От едната страна на схемата е управляващия компютър свързан към джойстик и модул за безжична комуникация, а от другата е роботът, който също е свързан към модул за безжична комуникация. Така реализирана схемата позволява да се изпращат управляващи сигнали от компютъра към робота в едната посока, а в обратната посока да се изпращат сигнали от сензорите на робота към компютъра(обратна връзка).



Фигура 3.1. Управление на робота за инспекция.

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. Натоварвайки компютъра от друга страна редуцираме работата на контролера на робота, като той само изпълнява получените команди и изпраща данните от сензорите. Това допринася за бързодействието на робота, снижава критерия от необходимостта за използване на скъпи и мощни контролери, намалява риска от забиване на системата и намалява консумираната енергия от батерията, [64].

Използваният модул за безжична комуникация е X-bee модул. Той е проектиран да отговаря на IEEE 802.15.4 стандартите и поддържа уникалните нужди от ниска цена и безжичен мрежов сензор с ниска консумация. Принципа на работа на модулите и свързването им е показан на фигура 3.2. XБee модула се свързва с множеството устройства, през логическо ниво на асинхронен сериен порт. През своя сериен порт модула може да комуникира с всяко логическо или напреженово съвместими UART, или с всяко серийно устройство с трансмитер на нива, [65].

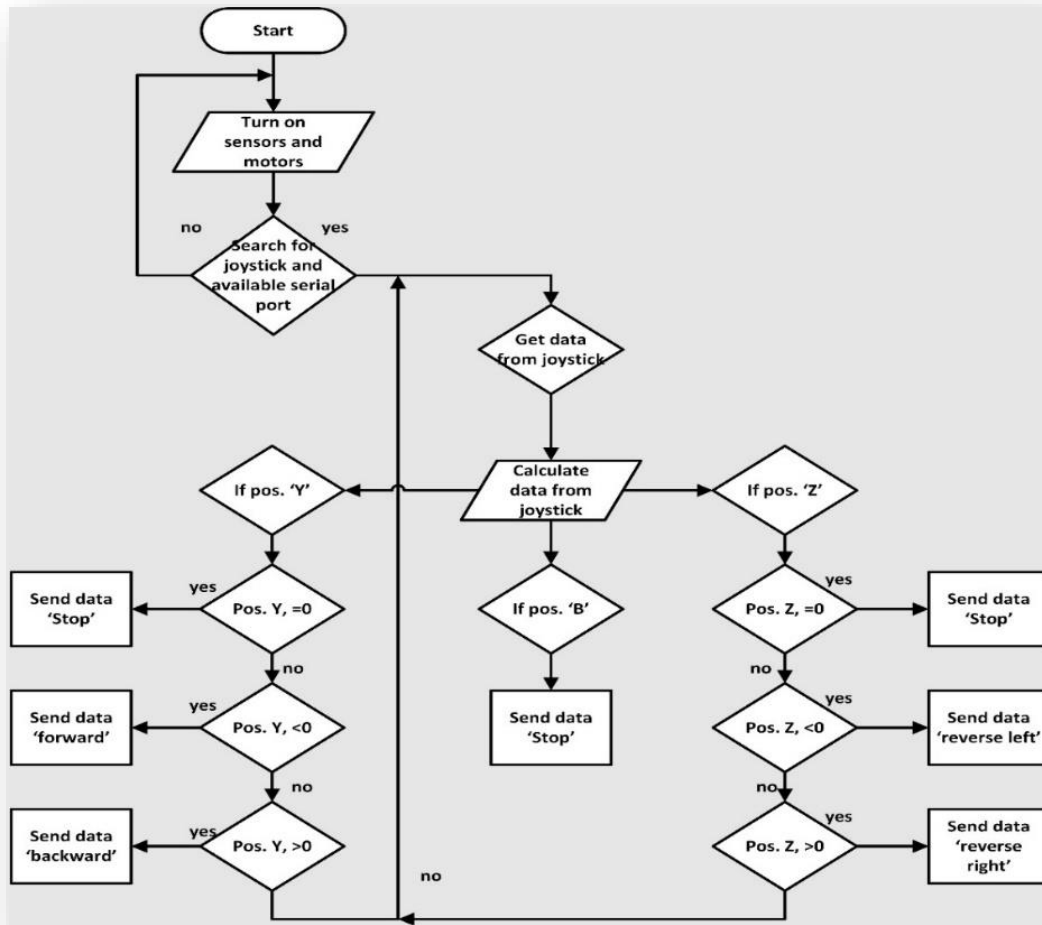


Фигура 3.2. Принципна схема на комуникацията.

За реализиране на управлението на работа ще се разработи приложение написано на програмният език C#, върху платформата .NET, разработени от Майкрософт. В днешно време C# е един от най-популярните езици за програмиране. На него пишат милиони разработчици по цял свят. Тъй като C# е разработен от Майкрософт като част от тяхната съвременна платформа за разработка и изпълнение на приложения .NET Framework, езикът е силно разпространен сред Microsoft-ориентираните фирми, организации и индивидуални разработчици. Най-често C# програмите се изпълняват върху MS Windows, но .NET Framework и CLR се поддържа и за мобилни телефони и други преносими устройства базирани на Windows Mobile. Под Linux, FreeBSD, MacOS X и други операционни системи C# програмите могат да се изпълняват върху свободната .NET Framework имплементация Mono, която обаче не се поддържа официално от Microsoft.

C# е съвременен обектно-ориентиран език за програмиране с общо предназначение, създаден и развиван от Microsoft редом с .NET платформата. На езика C# и върху .NET платформата се разработва изключително разнообразен софтуер: офис приложения, уеб приложения и уеб сайтове. Въведение в програмирането със C# настолни приложения, богати на функционалност мултимедийни Интернет приложения (RIA), приложения за мобилни телефони, игри и много други. C# е език от високо ниво, който прилича на Java и C++ и донякъде на езици като Delphi, VB.NET и C. Всички C# програми са обектно-ориентирани. Те представляват съвкупност от дефиниции на класове, които съдържат в себе си методи, а в методите е разположена програмната логика – инструкциите, които компютърът изпълнява [66].

Формирането на пакетите от данни става, чрез написания софтуер на база на алгоритъма за управление даден на фигура 3.3. Написано е конзолно приложение, което представлява софтуера за управление на работа. Тази програма чете данните от джойстика, преобразува ги в данни, които да се четат директно от контролера като команди за изпълнение, формира пакетите за предаване и ги изпраща към работа. За да се покажат стойностите на изпратените параметри се въведени оператори за отпечатване в конзолата. Това е реализирано в три функции като всяка изпълнява една задача.



Фигура 3.3. Блок схема на алгоритъма за управление на работа.

„static void Main()“ е главната функция, която изпраща данните от другите функции към работа. 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()“:

- проверява се дали е включен джойстик към компютъра, ако няма такъв програмата изчаква докато не бъде включен.

```

if (joystickGuid == Guid.Empty)
{
    Console.WriteLine("No joystick/Gamepad found.");
    Console.ReadKey();
    Environment.Exit(1);
}

```

```
}
```

- проверява дали е свързан комуникационният модул към компютъра, ако не е отново се изчаква докато не се включи такъв.

```
static SerialPort port = new SerialPort("COM2", 9600, Parity.None, 8, StopBits.One);
```

„static void Control(string position, int speed)“ – преобразува получените данни от втората функция и ги изпраща към главната функция. Получените параметри се проверяват и се разпределят според стойностите им чрез „if else if“ оператора, където се формират и пакетите за изпращане.

Когато всичко е включено програмата започва да извлича данните от джойстика. За всяка промяна тя получава нови параметри. Това е реализирано чрез прилагане на безкраен цикъл (while), които извежда новите стойности генерирани от бутоните на джойстика. Тези параметри се преобразуват, като се формират два параметъра за управлението на платформата на робота: единия е за посоката на движение (string position), а другия е за скоростта на въртене на електромоторите (int speed). Параметъра за посока е във формат string и представлява низ от символи, които дават желаната посока. Параметъра за скоростта е във формат integer, който се променя от 0 до 128. Това е резолюцията за промяна на скоростта, което позволява да се постига плавно потегляне и спиране на робота.

Тези параметри се добавят към пакетите за управление. Командата, с която се изпращат пакетите е: „port.Write(new byte[] { Convert.ToByte('w'), sendSpeed }, 0, 2);“. Тук „port.Write“ записва данните към серийния порт за да се изпрати пакета, а „new byte[]“ е самия пакет, като в скобите са двата параметъра за посока и скорост. Последните цифри са за означаване на начало на пакета и броя на променливите, които съдържа. Така контролера получава на готово параметрите и не се налага да преобразува или изчислява допълнително. Следва част от програмния код за формиране на пакетите:

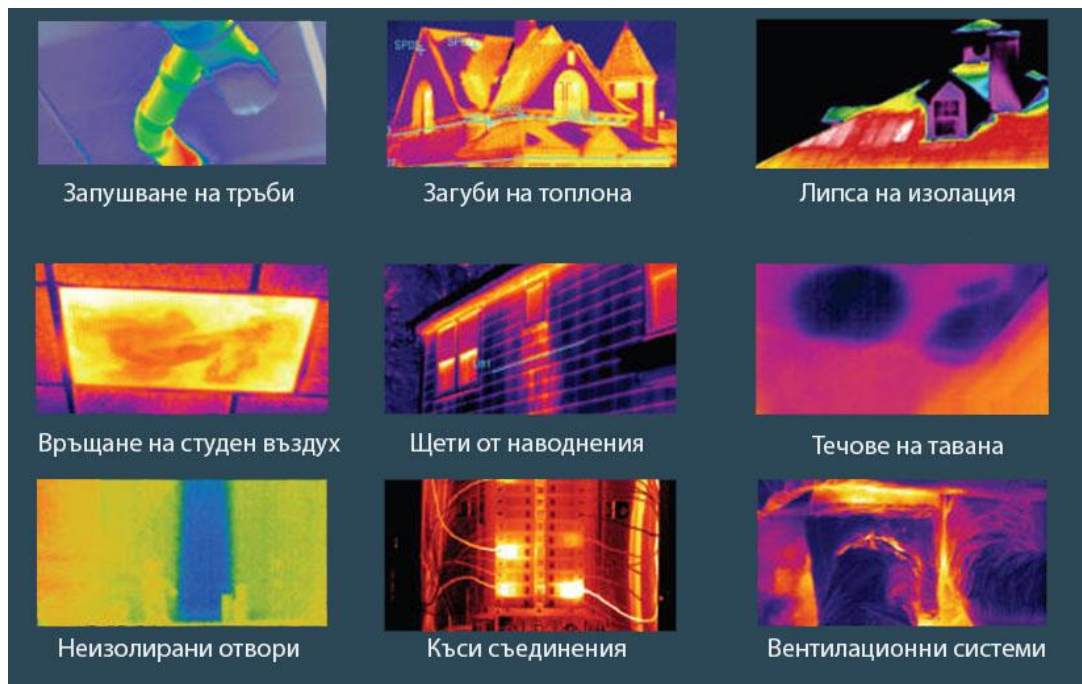
```
if (speed < 32511)
{
    sendSpeed = Convert.ToByte(-(speed - 32511) / 127); ;
    port.Write(new byte[] { Convert.ToByte('w'), sendSpeed }, 0, 2);
    Console.WriteLine(sendSpeed);
}
```

Предвиден е бутон за принудително спиране на робота при настъпване на някаква грешка или повреда, за да се избегнат нежелани удари или пропадане. При натискане на този бутон към робота се изпраща пакет, който с нулеви стойности на параметрите за управление:

```
if (position[0] == 'B')
{
    port.Write(new byte[] { Convert.ToByte(position[0]), 0 }, 0, 2);
}
```


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. Ясно могат да се разграничат различните температури и да се анализират.



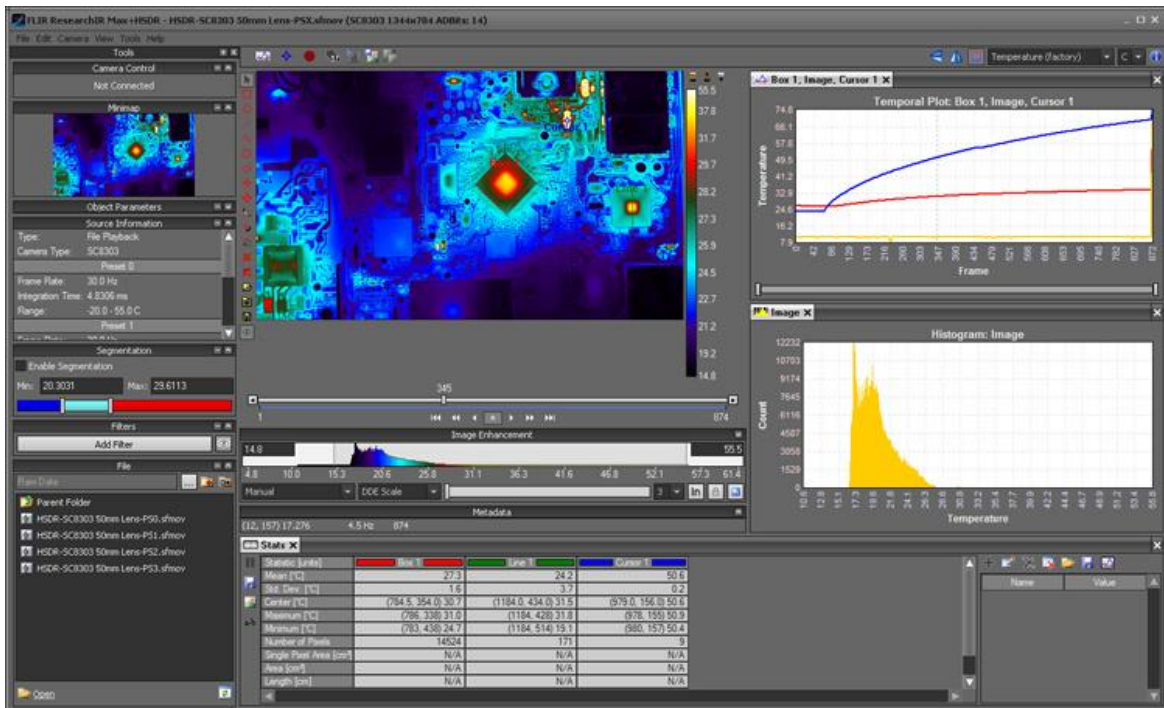
Фигура 3.4. Приложения на термо камерата.

Ще се използва и помощен софтуер ThermoVision SDK, който е софтуер за управление на достъпа до термо камерата и данните от изображенията. Може да се разработват програми за използване на камерата с различно приложение. Този продукт позволява да се възползваме напълно от възможностите на съвременните инфрачервени термо камери. Приложенията на ThermoVision SDK са следните:

- Създаване на комуникация между приложенията и камерата.
- Заснемане и събиране на изображения чрез FireWire™ и Ethernet интерфейси.
- Регулиране на параметрите за конфигурация на камерата и фокусиране.
- Управление на калибрирането на камерата.
- Изпращане на други команди към камерата.
- Генериране на реални температурни изображения от 16-битово изображение.
- Затваряне на комуникацията към термо камерата.

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

(фиг. 3.5). Така използването на термо камерата е по-ефективно и позволява задълбочени анализи за да се предотвратят бъдещи аварии или проблеми или да се овладеят и отстранят при настъпване.



Фигура 3.5. Специализиран софтуер FLIR Reporter.

3.2. Роботи за възрастни и инвалиди, ценово ориентирани решения

Нашата мобилна платформа трябва да бъде ефективна от гледна точка на разходите и удобна за ползване от хората с увреждания. Най-важното, което нашата платформа трябва да направи, е да се движи. Освен това, потребителският интерфейс трябва да се занимава с разнообразието от увреждания. Ясно е, че един метод не може да се използва във всяка ситуация. Така че, опитахме и оценихме няколко начина да контролираме движението на тази платформа. Друго важно нещо е, че нашият робот трябва да бъде програмиран по такъв начин, че да е безопасен за оператора и околностите му. Това означава, че дори в режим на ръчна работа, платформата трябва да вземе предпазни мерки и да предотврати всякакви действия, които ще доведат до щети не само на нея, но и на нейната среда. Сега ще обясним как решихме тези проблеми.

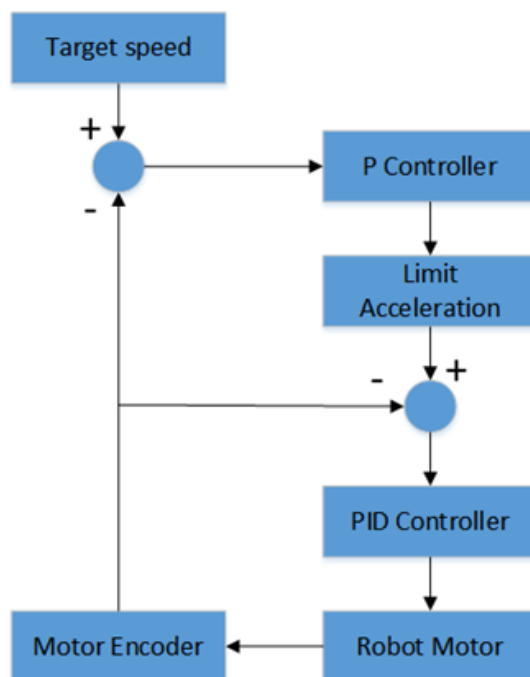
3.2.1. Управление на задвижването.

Искахме нашата платформа да има надеждно и сигурно движение. За да постигнем това, измерихме максималното ускорение за безопасно движение на нашия робот. Височината му е около 1 метър. Поставихме върху него нехлъзгава подложка и чаши пълни с течности. Резултатът е максимално ускорение от 0.4m/сек^2 . Така че това бяха границите на нашата платформа, в които чашите с течностите не падат и самите течности не се разливат.

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. Използвахме го, за да регулираме скоростта на всяко задвижващо колело. Обратната връзка беше дадена от енкодерите, монтирани върху осите на всяко от колелата. На всяко колело имаше два отделни контролера, работещи по следния принцип: (показано на фигура 3.6).

Тиковете от енкодерите измерват изминатото разстояние от робота. Чрез разграничаването му може да се изчисли текущата скорост на колелото. След това се изчислява разликата между желаната скорост и текущата. Тази разлика е дадена на входа на обикновен P контролер, чиято задача е да превърне желаната скорост в желаното ускорение. Това ускорение след това се ограничава и се подава в PID контролер. На изхода на контролера е необходимата мощност на двигателя за постигане на нашата целева скорост.



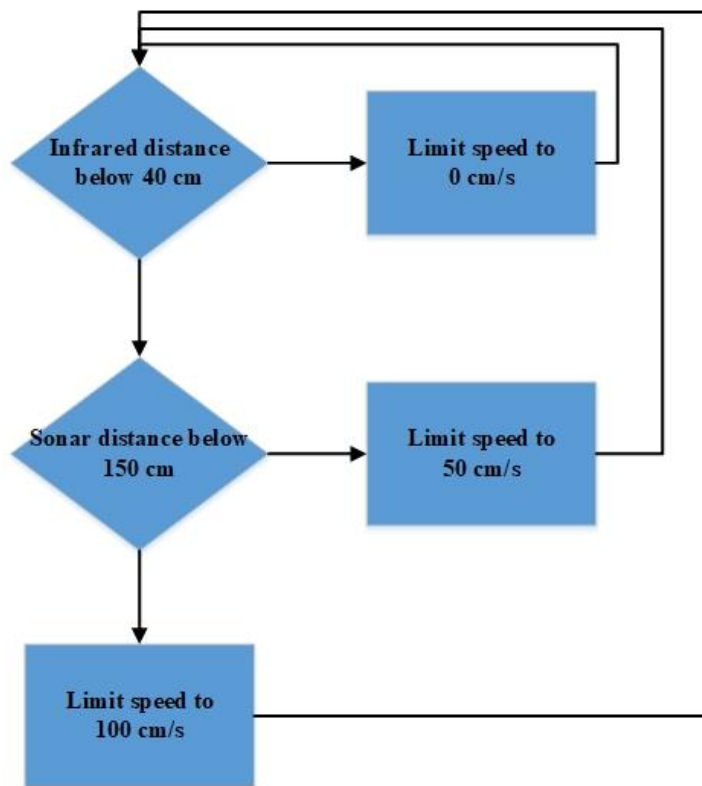
Фигура 3.6. Схема на управлението на двигателите.

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 0.8м; Сонарният сензор имаше по-голямо разстояние - 3м от мобилната платформа.

Нашите тестове показаха, че много по-близкият сензор за разстояние - инфрачервеният, не е достатъчен, когато искаме да избегнем сблъсъци. Когато този сензор усети обекта, вече е твърде късно, за да реагира роботът поради скоростта и масовите му свойства. И ограничаването на скоростта просто не е приложимо. Поради този проблем използваме сонарния сензор, за да ограничим скоростта до ниво, което е достатъчно, за да може роботът да спре, когато инфрачервените сензори усетят препятствието.

Също така, защото нашата целева група са хора с увреждания, можем да предположим, че живеят сами или се движат бавно, така че средата, в която работи роботът, е почти статична. Така че хората или предметите няма да се появят внезапно пред платформата и тези сензори, като се има предвид предложената логика (фигура 3.6), ще бъдат достатъчни, за да избегнат сблъсъци.



Фигура 3.7. Логика за ограничение на скоростите.

За нашия робот ние изчислихме, че когато е на около 1.5м от препятствието, скоростта трябва да бъде ограничена до 0.5м в секунда и когато препятствието е на 0.4м, роботът трябва да спре. Тези измервания са направени в съответствие с тези граници: максимална линейна скорост от 1м/сек и максимално ускорение от 0.4м/сек². Подробности за усъвършенства на системата за избягване на препятствия са представени в следваща точка от тази глава.

Отбягването на сблъсък е важно, защото при теле операция винаги има забавяне на видео потока и системата е много вероятно да се срине, защото операторът не може да предвиди какво ще се покаже пред платформата. Също така, когато става въпрос за хора в старческа възраст или хора с увреждания, е много възможно да допуснат грешка и да предизвикат нежелан сблъсък.

3.2.3. Изследване на методите за управление на сервизния робот.

Изпробвахме няколко метода за управление на движението на робота асистент. Ние експериментирахме с много традиционния метод като физическия джойстик, както и с иновативния контрол на движението на очите. Някои подробности за всички от тях са обяснени в следващите параграфи:

- **Физически джойстик:**

Стандартният метод за управление на всяко движещо се превозно средство е с джойстик. Опитаме и двете аналогови стикове и бутони. С PID контрола и алгоритмите за избягване на сблъсъка, които управляват робота чрез този метод, е много приятна и лесна работа. Въпреки че джойстика не може лесно да взаимодейства с уеб интерфейса, така че няма възможност за дистанционно управление на робота чрез него. Освен това хората с увреждания в ръцете не могат да го използват. По-възрастните хора не могат да взаимодействат добре с един аналогов стик. Така че, ние препоръчваме или разделянето на линейното и ротационното движение на системата на две отделни аналогови пръчки, или чрез бутони за отделно движение в някоя от посоките.

- **Виртуален джойстик:**

Следващият метод, който опитахме, и най-любимият ни е виртуалният джойстик. Тя работи като един аналогов стик на физически джойстик. Той е интегриран в уеб потребителския интерфейс и позволява лесно дистанционно управление на системата чрез всеки уеб браузър. Но отново, тестовете показват, че по-възрастните и хората с увреждания на ръце не могат да я използват добре. Така че, тук можем да сменим аналоговия виртуален стик с аналогови виртуални бутони. Натискането на някой от бутоните ще доведе до завъртане или движение в желаната посока.

- **Метод за управление чрез проследяване на главата:**

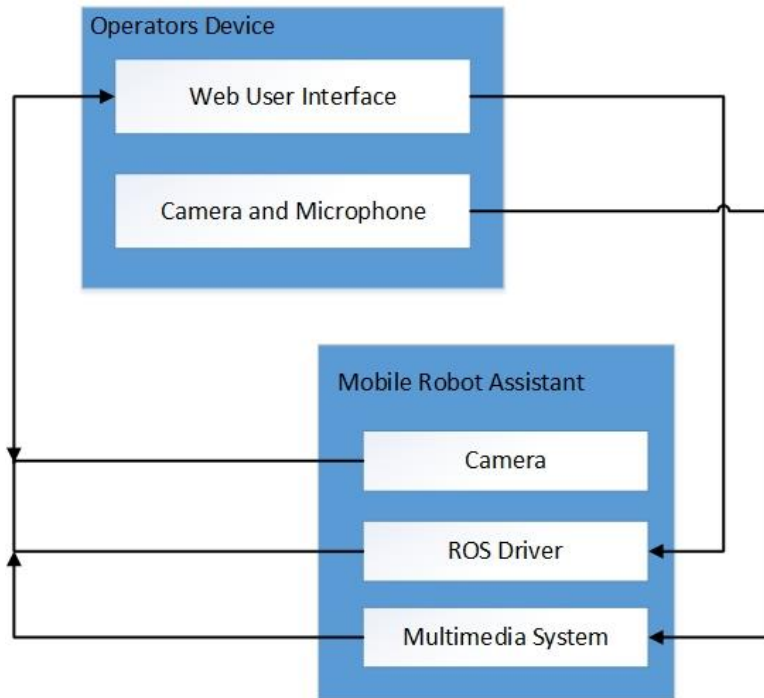
Оценяваме не толкова стандартния метод за контрол, който следи движението на главата на оператора. За да направите това, контролиращото устройство - лаптоп, трябва да е снабдено с уеб камера и трябва да работи с допълнителен софтуер, който анализира видео потока на камерата и изпраща команди към уеб интерфейса или директно към робота чрез `roslibjs` и `rosbridge`. Софтуерът, използван от нас, се основава на проекта `Enable Viacam`, отворен код за управление на мишката. Този метод изисква сравнително малко време за обучение и общите резултати са много впечатляващи, така че това е задължително условие за такива системи. Но трябва да споменем, че устройството за управление трябва да бъде статично по време на операцията по управление, така че ако се използва смартфон и таблет, те трябва да бъдат поставени на стойка. Също така, този метод не може да бъде директно внедрен в софтуера на робота, защото има закъснения или кратки паузи по време на предаването на видео на живо.

- **Контрол чрез проследяване на очите:**

Последният изследван метод е движението на робота чрез проследяване движението само на очите на оператора. Това изисква не само статично устройство, но и хубава уеб камера. За това изследване опитахме да преместите виртуалния джойстик в уеб интерфейса, като преместите мишката с очите си. Използвахме безплатната програма за контрол на мишката - `Camera Mouse`. Този метод изисква по-дълго време за обучение. Също така, придвижването на очите около екрана прави много трудно да се наблюдава какво се случва във видео потока. Това беше най-трудният начин за контролиране на системата, но не можем да я пропуснем, защото може би е последната възможност за някои от хората с увреждания.

3.3. Потребителски интерфейс за управление на сервизни работи

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. С този интерфейс контролът, настройката и комуникацията с робота могат да се извършват чрез всяко прилично устройство с уеб браузър - смартфон, таблет, лаптоп или персонален компютър. Този тип интерфейс ни позволи да намалим разходите за хардуера до минимум, защото този интерфейс бе хостван на вече съществуващия компютър в робота.



Фигура 3.8. Принцип на работа на интерфейса на робота.

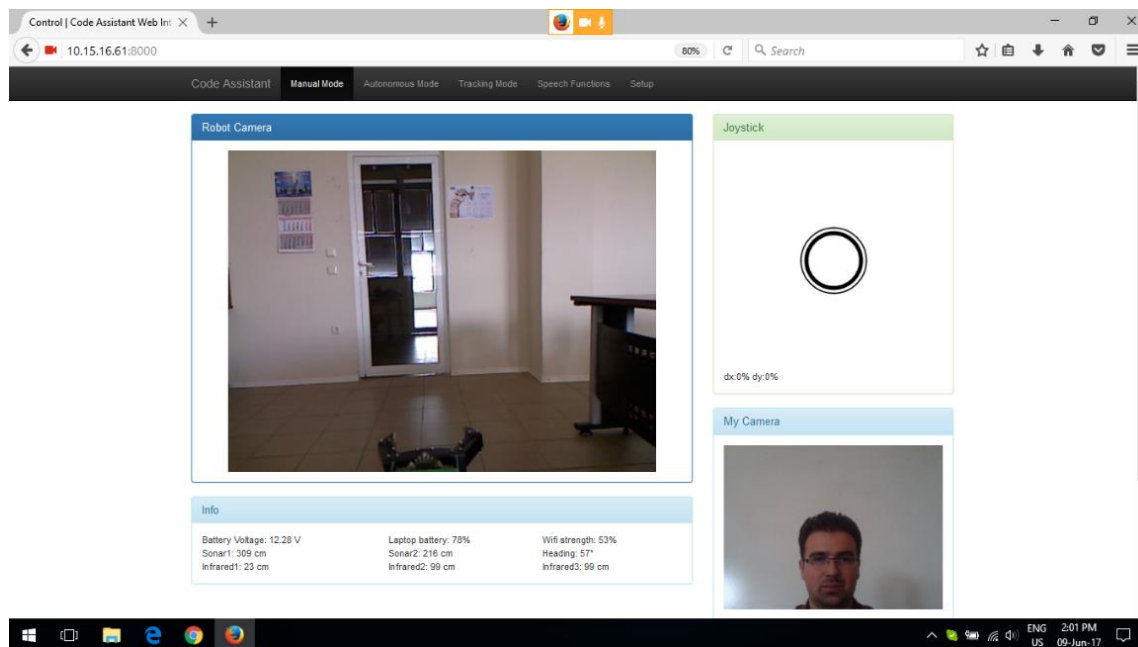
Фигура 3.8 показва как интерфейсът взаимодейства със системата ROS. Чрез ROSBridge мрежата и JavaScript интерфейсът асинхронно получава текущото състояние на мобилната система и изпраща заявки за движение към нея. Освен това потребителският интерфейс получава потока от камери чрез HTTP връзка благодарение на ROS уеб пакета за видео. Аудио потокът от робот микрофона и видео / аудио потока от устройството към екрана на робота и високоговорителя се прехвърлят чрез WebRTC протокола. WebRTC се поддържа в повечето браузъри за Windows, Linux и Android.

За интерфейса използвахме Python базираната уеб рамка - Django [67]. Въведохме двустранна видео и аудио комуникация в реално време чрез WebRTC библиотеката, която вече се поддържа в повечето съвременни уеб браузъри. Интерфейсът осигурява достъп до всички параметри на нашата система на робота асистент и спестява всякакви модификации, които правят персонализирането и настройването на робота много приятна и удобна за потребителя задача. Също така има виртуален джойстик за управление на позицията на робота и видео на живо от неговата уеб камера. Също така, текущото състояние на

платформата се показва на интерфейса. Направихме специални бутони за гласово синтезиране на важна информация за хората, като времето или текущото време.

3.3.1. Меню за управление в ръчен режим.

В ръчен режим потребителят вижда четири прозореца. Камера на робота, камера на устройството на потребителя, виртуален джойстик и информация за системите на робота. В този режим потребителят контролира всяко движение на робота самостоятелно (фиг.3.9).



Фигура 3.9. Меню за управление в ръчен режим

Тези прозорци на достатъчни за да се управлява робота, както потребителя пожелае. Видеоот от камерата на робота ни дава ясна картина на средата пред робота.

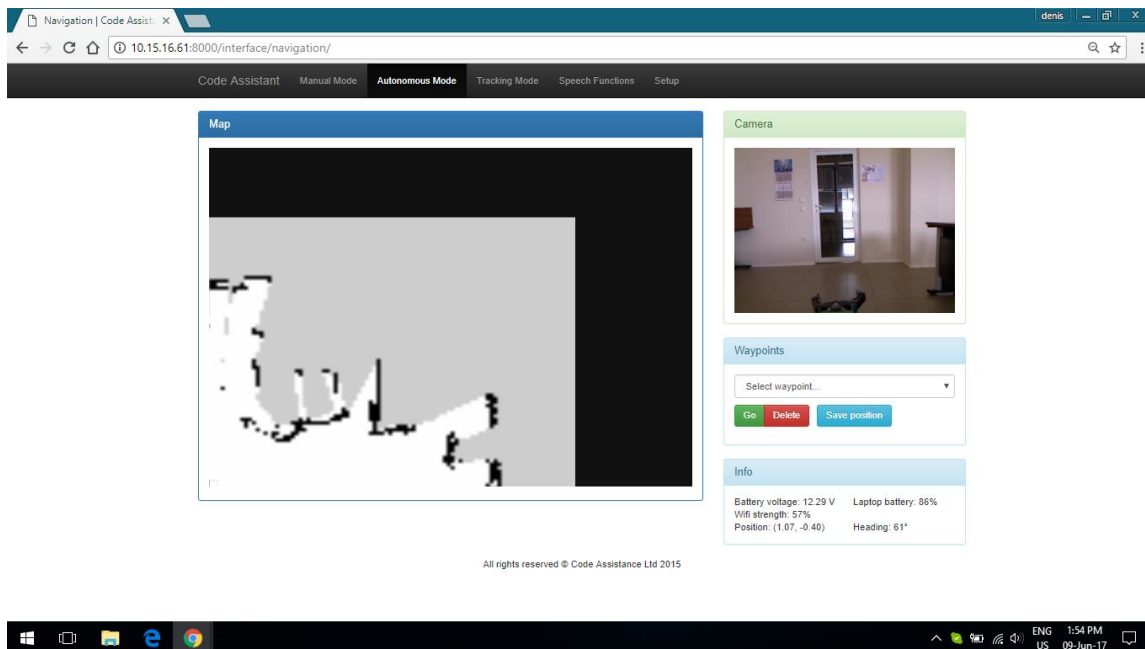
Прозорецът за виртуалния джойстик позволява прецизно подаване на сигналите за движение на робота. Виртуалния джойстик работи като един аналогов стик на физически джойстик. Той е интегриран в уеб потребителския интерфейс и позволява лесно дистанционно управление на системата чрез всеки уеб браузър. Но отново, тестовете показват, че по-възрастните и хората с увреждания на ръце не могат да я използват добре. Така че, тук можем да сменим аналоговия виртуален стик с аналогови виртуални бутони. Натискането на някой от бутоните ще доведе до завъртане или движение в желаната посока.

Прозорецът за информация ни дава процентите на заряда на батериите на робота и на лаптопа, силата на сигнала на Wi-Fi модула, разстоянията, които измерват сензорите за дистанция и ориентацията на робота спрямо стартовата му позиция. Този режим изисква предварително обучение на оператора или възрастния човек, за да управлява робота.

3.3.2. Меню за управление в полу-автономен режим.

Полу-автономен режим: в това меню добавихме системата за автономна навигацията на робота. В това меню потребителят вижда картата на сградата и може да управлява робота, като избира желаната дестинация. Роботът ще отиде там автономно, благодарение на

навигационния стек на ROS. Потребителят може да види къде се намира роботът върху картата и какво вижда камерата на робота (фигура 3.10).



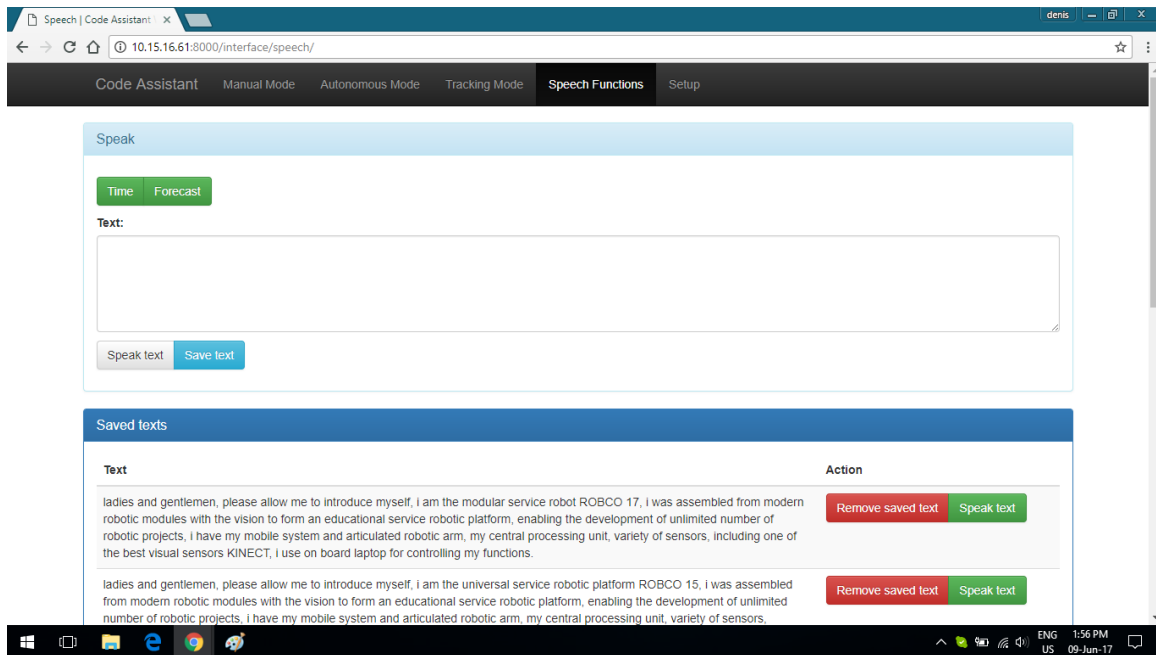
Фигура 3.10. Меню за полу-автономно управление.

Добавен е допълнителен прозорец, от който можем да задаваме запаметени позиции, без да е необходимо да ги избираме всеки път от картата. Това се изпълнява като се натискат бутоните от прозореца. Има бутони Go – тръгни, Stop - спри, Delete – изтрий позицията, Save Position – запамети позицията. Така правим работата с робота по-лесна.

3.3.3. Меню за управление на гласовите команди на робота.

В това меню са добавени функциите за говор на робота (фигура 3.11). Има два бутона Time – робота казва на потребителя моментното часово време, Data – робота казва датата и Forecast – робота казва на потребителя, какво е времето навън. За източник на часово време и датата използваме часа и датата от компютъра на робота, които се настройват автоматично през интернет. За източник на прогнозата, използваме препратка към сайт за прогнозиране на времето.

В това меню има прозорци за набиране на текст и за пускане на говор на запаметени предварително текстове. Това има приложение, както при възрастни хора, така и при инвалиди. Хората със затруднение в говора могат да набират текст, който искат да произнесат. След което робота ще говори вместо тях.



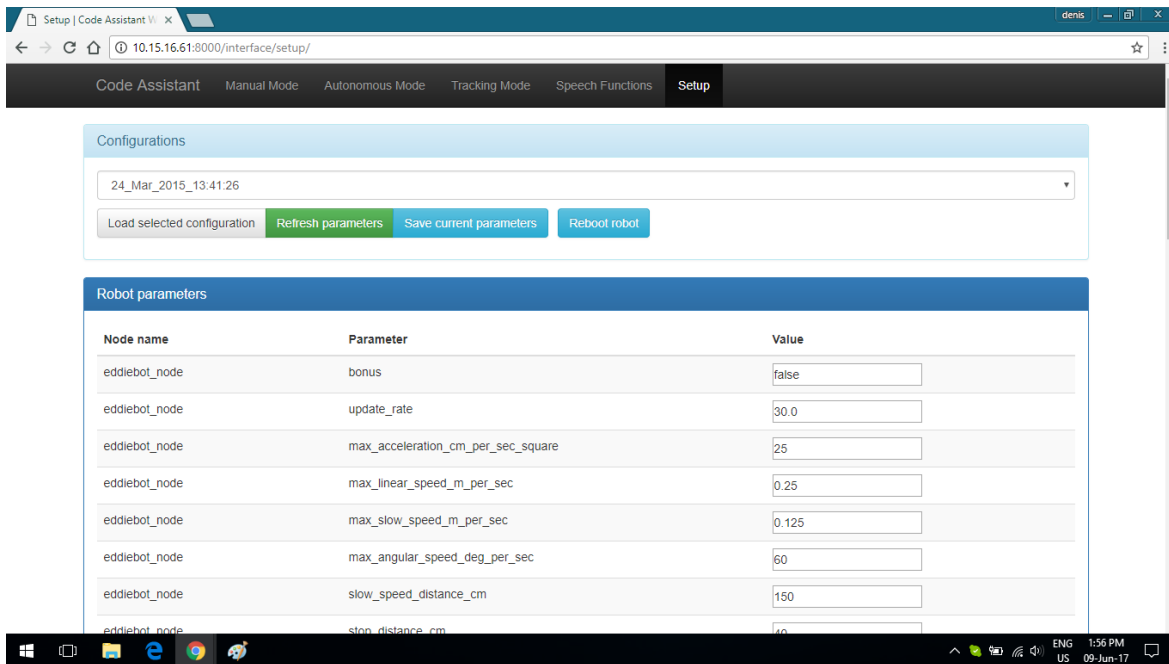
Фигура 3.11. Меню за гласови команди.

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

Това меню е предназначено главно за обучените оператори и разработчиците. То позволява настройки на почти всички параметри от различните системи на робота. Като се започне с тези за подаване на команди за задвижванията – мощност, ускорение, скорост, след това настройване на данните от сензорите и се достигне до опциите за аудио и видео сигналите.

В горния прозорец се задава определена конфигурация на параметрите. Тази конфигурация е направена след множество тестове на робота, докато не се достигне до стабилност при управление. Бутоните под този прозорец са за зареждане на друга конфигурация, рестартиране на параметрите от текущата конфигурация, запаметяване на параметрите след промяна и бутон за рестартиране на ROS. Последният бутон е важен, понеже при проблеми или промени в системите се налага те да бъдат рестартирани за да заработят правилно.

Втория прозорец е с имената на параметрите, нодовете, в които се намират и стойностите, които притежават. Има възможност да се променят само стойностите на тези параметри, без да се променят техните имена или нодове. Понеже може да се променят, така че робота да стане опасен, е препоръчително да се променят само от разработчиците.



Фигура 3.12. Меню за настройки на робота.

3.4. Управление на роботизирани манипулатори.

След като вече имаме разработени системи за автономна навигация и разпознаване на обекти, трябва да научим робота да манипулира с тези обекти. За постигане на целта е разработен метод за управление на роботизирана ръка с три степени на свобода.

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

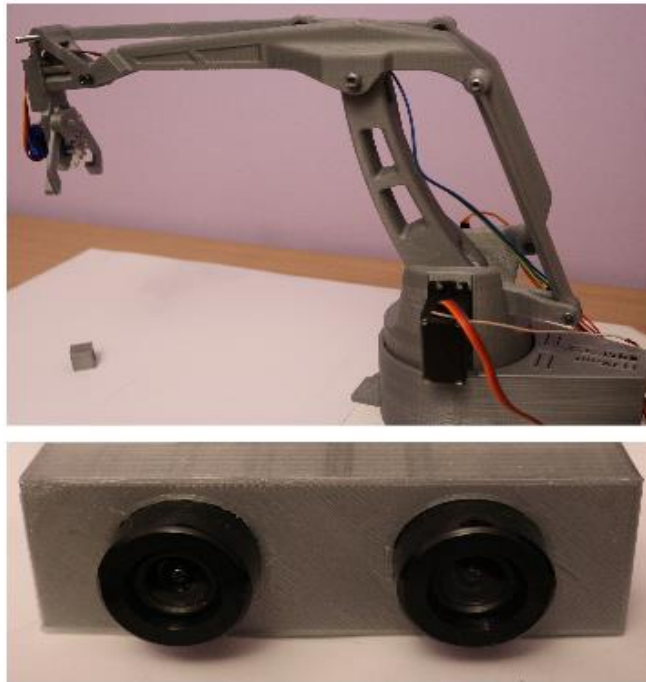
Стъпка 1: разпознаване на определен обект и придвижване на мобилната платформа възможно най-близо до него, така че обекта да застане в зоната на достижимост на ръката на робота. Освен това, докато се движите по картата, роботът трябва да разпознава обектите, които ги интересуват, и да ги постави на картата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

В тази точка разглеждаме проблема на позициониране на роботизиран манипулатор до определен обект, като крайната цел е хващане на този обект. Проблемите като определяне на позицията и изчисляването на точните динамични и кинематични модели се решават чрез

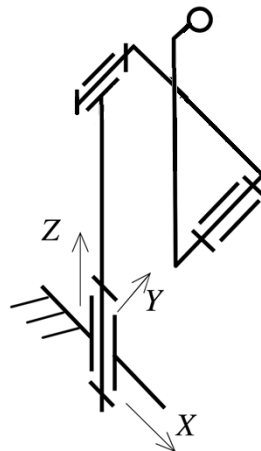
използване на стерео компютърно зрение, което осигурява визуална обратна връзка за компенсации на всякакви неточности чрез процедура за Iterative Learning Control (ILC).



Фигура 3.13. 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 3.14. Кинематична схема на манипулатора.

3.4.2. Предварителни условия.

Следните предварителни изисквания трябва да бъдат изпълнени:

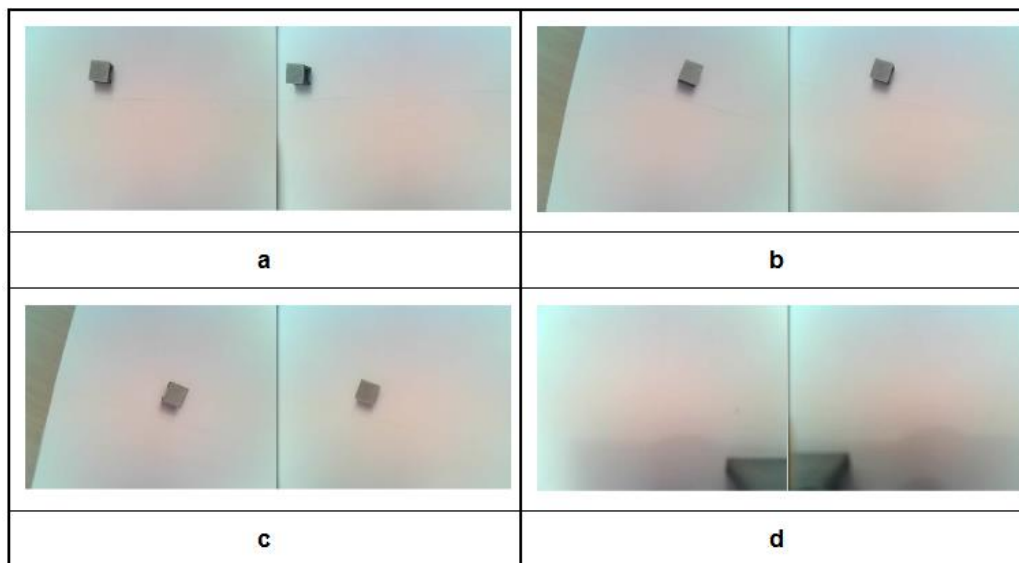
- (P1) Стерео камерата трябва да бъде монтиране възможно най-близо до хващача и да се движи заедно с него.
- (P2) Идеалната позиция на камерата е тази, при която зоната за хващане се намира в долните ляв и десен ъгли съответно на дясната и лявата камери.
- (P3) По време на цялото движение хващача трябва да е позициониран перпендикулярно на двумерната координатно пространство на робота.
- (P4) Моторите на робота трябва да могат да запазват позицията на ставите на манипулатора.
- (P5) Роботът трябва да измерва детерминистично текущата си позиция.

3.4.3. Алгоритъм.

Следващият алгоритъм разчита на визуални данни от стерео камерата в реално време за да може да позиционира хващача на идеалната позиция за хващане на всеки предварителни избран обект:

- (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 работи, 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 3.15. Фази на изпълнение на алгоритъма за позициониране.

ILC главно се използва за управление на индустриални роботи, които изпълняват повтарящи се задачи. В нашия случай, повтарящата се задача е движението на хващача до достигане на обекта.

3.4.4. Приложение на ВЕА 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 ВЕА.
- (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 ВЕА и ILC и обнови управляващите сигнали и премини на стъпка S5.2.

3.4.5. 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 работи

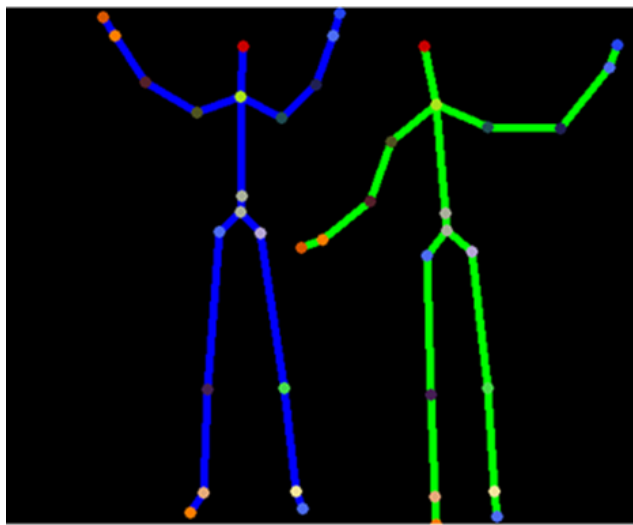
3.5.1. Алгоритъм за следене на човек и разпознаване на падане.

Сервизните работи предназначени за работа с хора трябва да разполагат със системи за осъществяване на взаимодействието човек – робот. Такива системи са тези, които разпознават определени действия и/или предприемат действия като отговор на вече разпознатите.

Роботите, които ще се помещават в домовете на хората или в обществени сгради трябва преди всичко да се грижат за безопасността на хората. Поради това се налага приложението на алгоритми за следене на хората и съответно за предприемане на определени действия при различни ситуации.

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. След обединяване на всички тези системи можем да реализираме алгоритъм за следене на човека, т.е. робота да се превърне в сянка на човека за който се грижи и във всеки един момент да наблюдава неговото състояние. В допълнение на това робота е добре да разпознава някои опасни ситуации като припадък, прилошаване и други.

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. По този начин ние можем да разпознаваме някои характерни движения или някои необичайни такива (фигура 3.16). Използвайки тези познания за тези движения, можем да зададем на робота да предприема определени действия при разпознаване на конкретно движение.

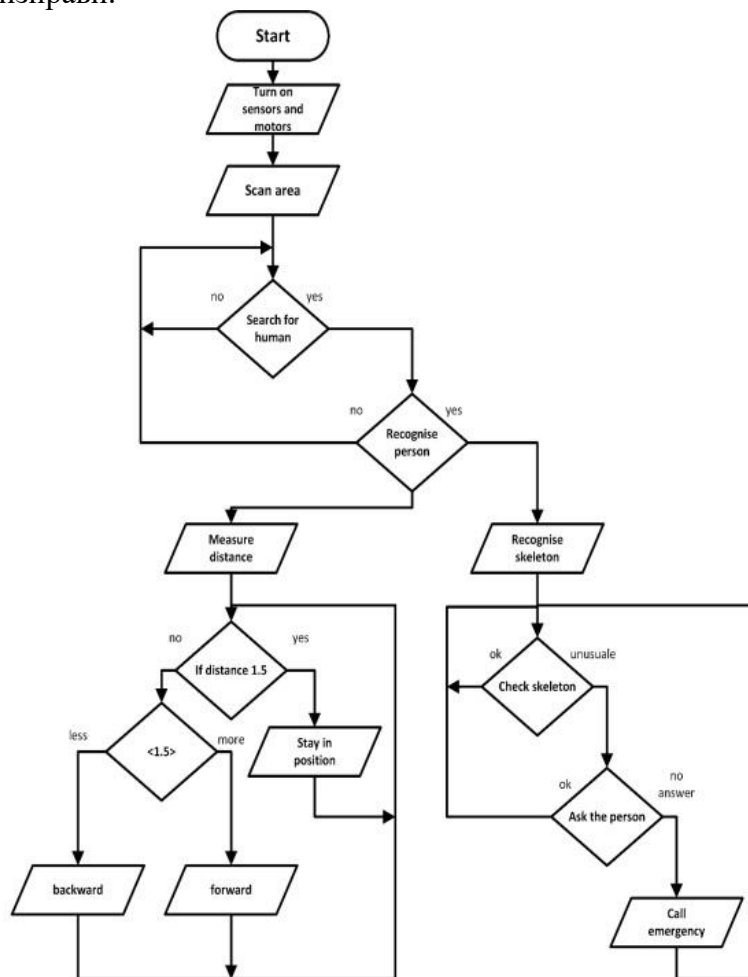


Фигура 3.16. Създаване на модел на скелета на човек.

Алгоритъмът за следене и разпознаване на падане има за цел да осигури постоянно наблюдение на човека и при установяване на падане да алармира близките на човека и бърза помощ за случил се инцидент. Описанието на алгоритъма е следното (фиг. 3.17):

- Робота се включва и активира всичките си сензори, системи и механизми. След това 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 Кинект-а да измерва разстоянието до обекти. Граничното разстояние, което робота трябва да поддържа е между 1.5 – 2м. Ако разстоянието е по-малко от 1.5м робота се отдалечава от човека на където има възможен път. Ако това разстояние е по-голямо от 2м, тогава робота се доближава. По този начин робота спазва определена дистанция от човека и така не му пречи.
- Разпознаването на падане е базирано на следене на точките на скелета и по конкретно тези над кръста на човека. Ако се случи, така че тези точки се позиционират под кръста, тогава можем да предполагаме, че човека е паднал. 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.

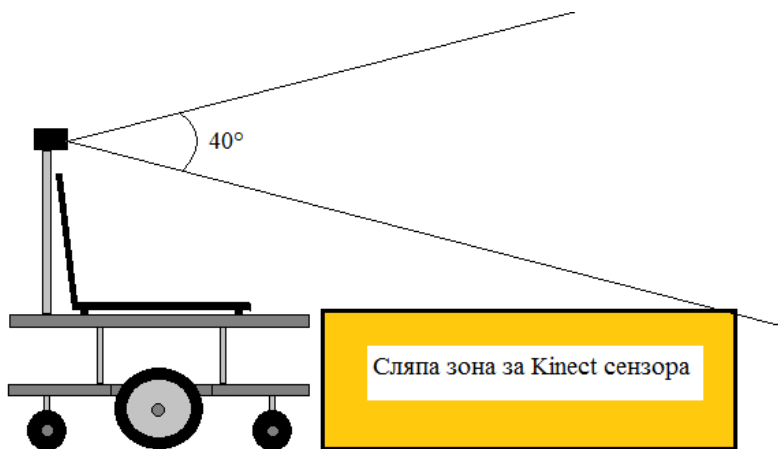


Фигура 3.17. Алгоритъм за следене и разпознаване на падане.

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). Тази зона е много рискова понеже така роботът може да се сблъска с различни малки предмети, животни или дори хора, и дори няма да регистрира наличието им. Поради това трябва да се добавят допълнителни сензори, които да покрият тази зона и да регистрират появата на различни обекти непосредствено пред робота. Тези обекти обикновено са динамични, което налага нуждата от голяма честота на измерванията.



Фигура 3.18. Проблемна зона.

Зоната, която трябва да се обхване е с формата на паралелепипед с размери височина 0.25 м, дължина 0.6м и ширина 0.45м. Така характеризирана тази зона изисква сензори с максимален обхват на измерване поне 0.7м, за да се постигне точност при измерванията. Сензорите трябва да се разположат, така че оптимално да покриват празната зона.

При регистриране на обект пред робота, от страна на допълнителните сензори, роботът трябва да спре движението си и да потърси друг път, по който да продължи да се движи безопасно, заобикаляйки обекта. Ако не може да се намери друг път, тогава робота трябва да изчака докато засечения обект на се премести.

- **Решение на проблема.**

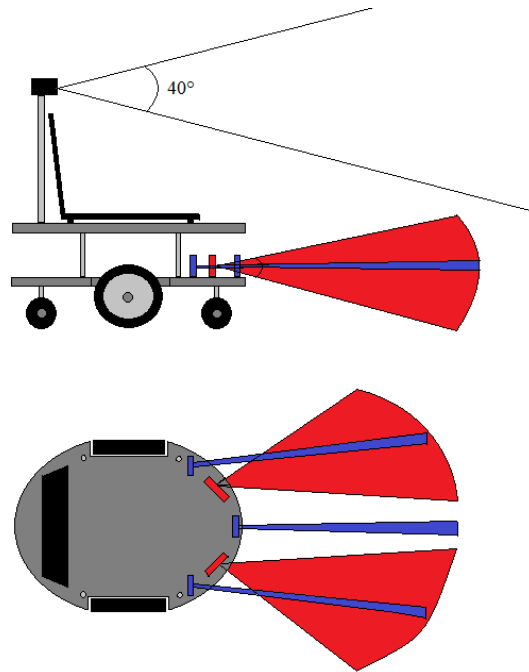
За управлението на двигателите е добавен ПИД регулатор към системата за навигация. Този регулатор се грижи за точното подаване на мощността. Така се премахват проблеми свързани с отклонения по време на движение. Също така навигационната система е програмирана да работи на принципа на алгоритъма А*, като се стареа да намира най-кратък път до дадена цел.

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 0.15м, 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 0.7м, понеже след това се припокриват с Kinect сензора. Също така като се използва по-малък диапазон на обхват на сензорите се увеличава тяхната точност и бързина на измерване.

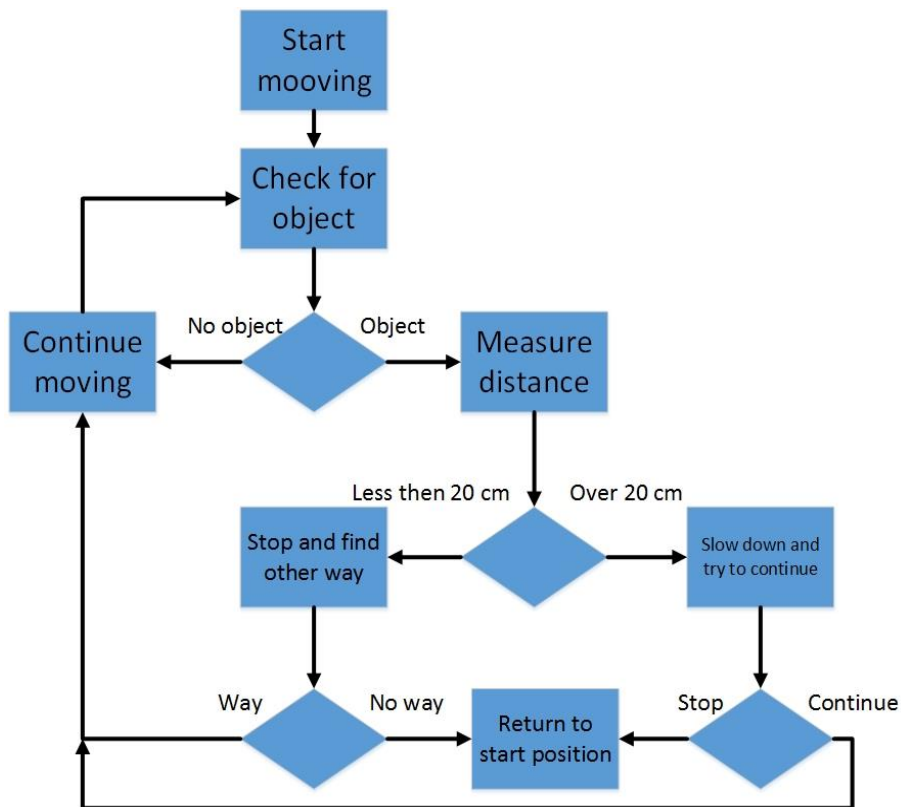


Фигура 3.19. Обхват на сензорите.

Сензорите са свързани към основния контролер на робота. След това данните от измерванията им се предават към компютъра на робота. Компютъра пресмята алгоритмите за навигацията и другите системи на робота и изпраща управляващи сигнали към контролера.

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. Целта на този алгоритъм е да добави допълнителни условия, свързани с измерванията на инфрачервените и ултразвуковите сензори. Блокова схема на алгоритъма е показана на фигура 3.20.



Фигура 3.20. 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. При разстояние по-малко от 0.2м робота трябва да спре и да потърси алтернативен път ако има такъв. При разстояние повече от 0.2м робота трябва да намали с готовност за спиране и отново да потърси алтернативен път до зададената цел. Всички допълнителни команди преминават праз ПИД регулатора в пакета за навигацията.

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 0.1м;
- 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.

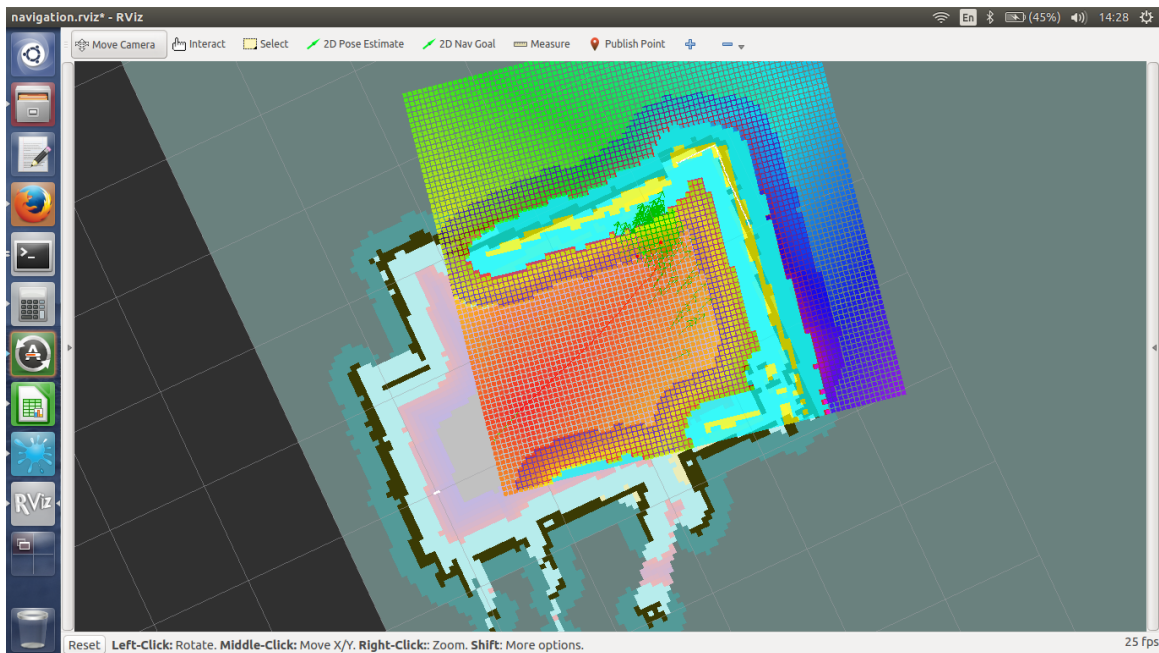
На фигура 4.1 в цветното поле е показана картата на помещението, местоположението на робота и изчислената траектория до другия ъгъл на стаята. Това цветно поле представлява обхвата на дълбочинния сензор на Кинект-а. оцветените в жълто стени се използват като маркери за ориентиране. И понеже няма препятствия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

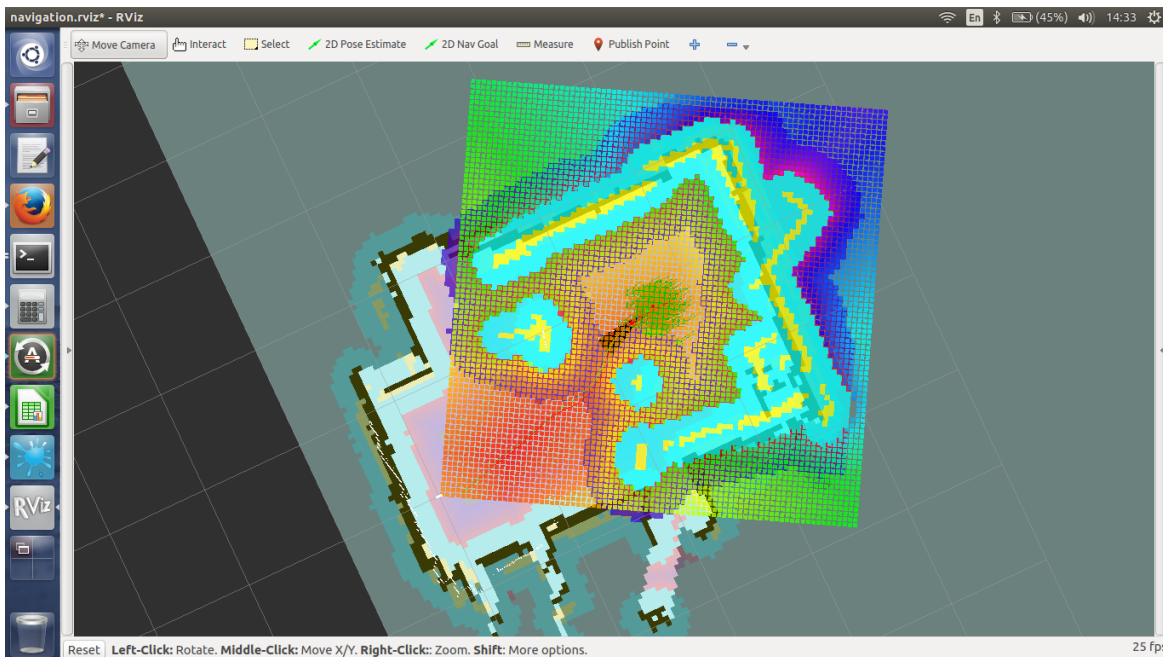
Резултати от експеримент 2. На робота му бе зададена точка, до която да достигне. Навигационната система успешно контролира робота до достигане на целта, като отклонението между центъра на мобилната платформа на робота и зададената точка е около 0.15м.

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 4.1. 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 4.2. Разпознаване на допълнително поставените обекти и изчисляване на нов път.

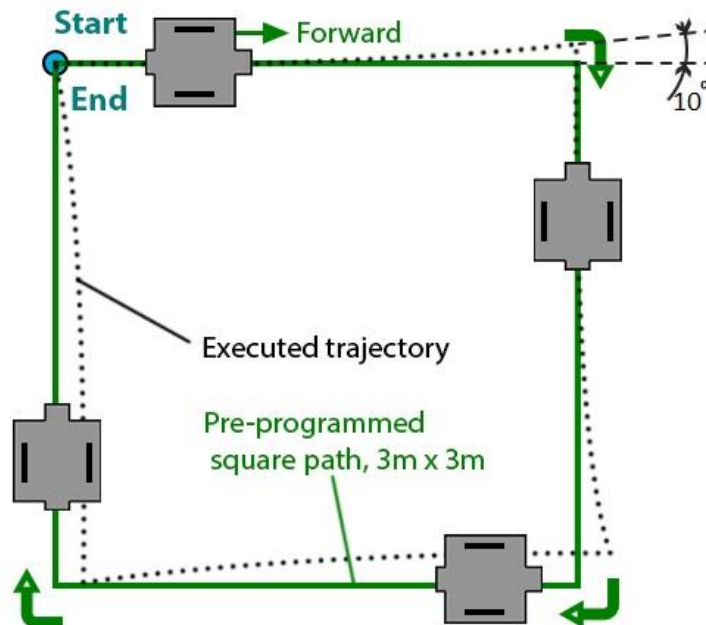
4.2. Експерименти и резултати след коригиране на грешката на одомерията

Грешката на одомерията е от голямо значение за правилната работа на навигацията. Поради това проведехме 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 одомеричната грешка чрез метода на изпълнение на квадратна траектория;
- Измерване на одомеричната грешка след въвеждане на коригирано управление на моторите, което компенсира грешката;
- Измерване на одомеричната грешка след добавяне на IMU сензор.

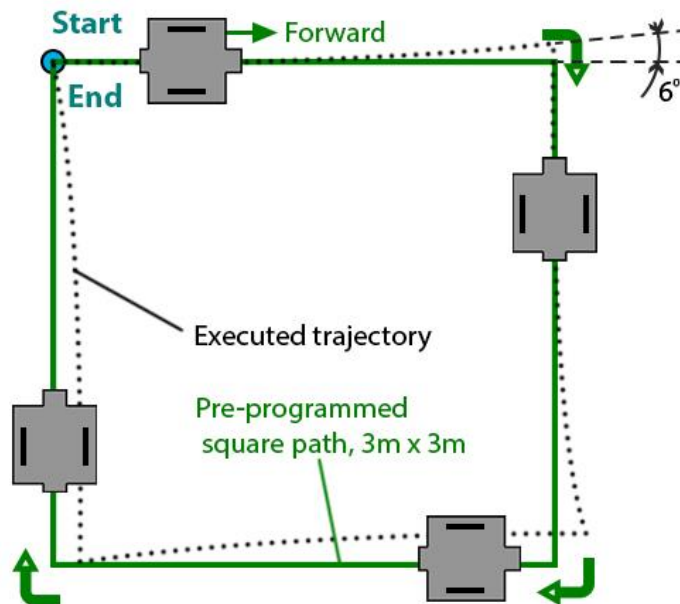
Всички изследвания са проведени като се използва принципа на двупосочното изпълнение на квадратна траектория. Робота се програмира предварително да изпълнява такава траектория, без да се използват никакви допълнителни сензори освен енкoderите на моторите.

Резултати от експеримент 1. Одомеричната 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.



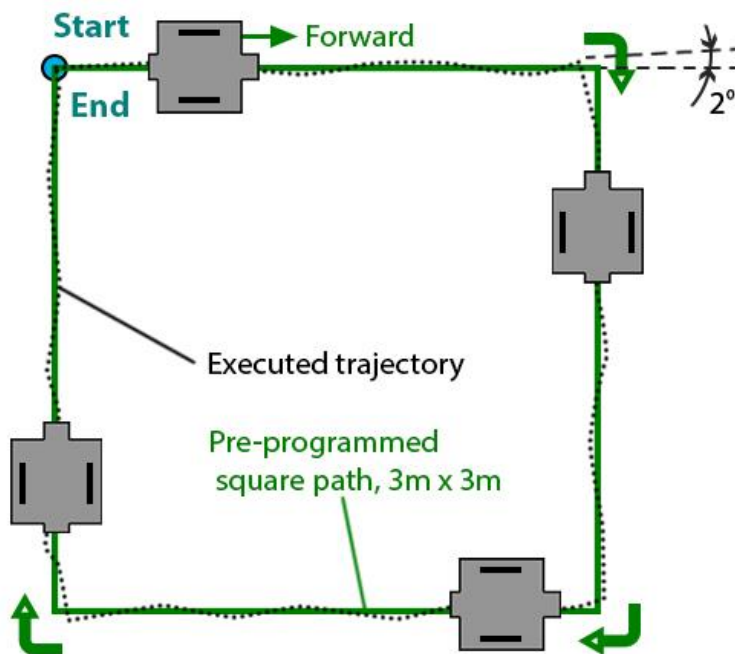
Фигура 4.3. Одомерична грешка.

Резултати от експеримент 2. При този експеримент повтаряме предходния, като измерената одомерична 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 3м до следващ завой, при изминаване на повече разстояние това отклонение отново не е достатъчно.



Фигура 4.4. 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.

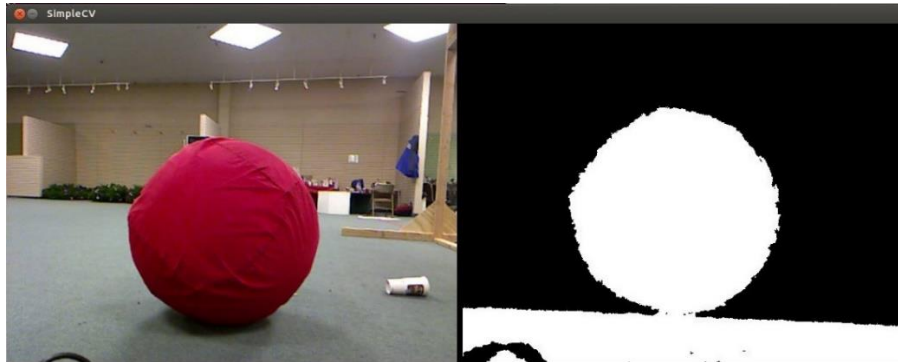


Фигура 4.5. Оптимизиране на движението на робота с IMU сензор.

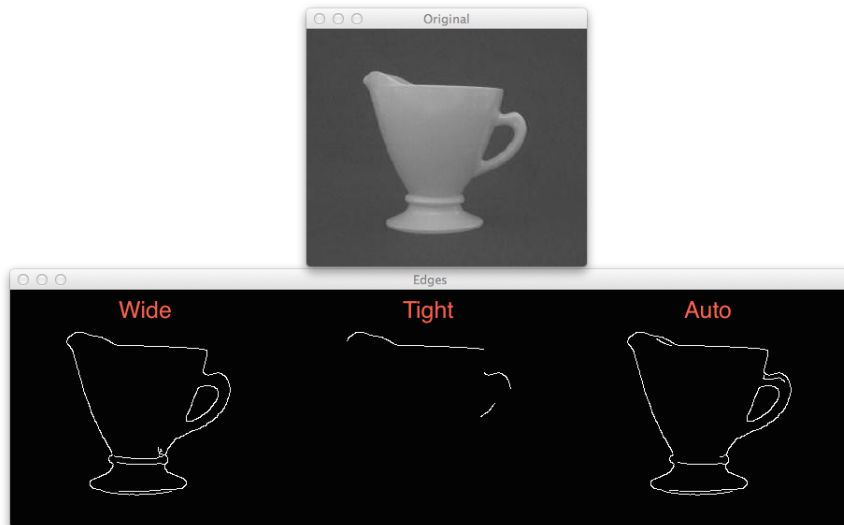
4.3. Експерименти и резултати на системата за компютърно зрение

Получени резултати от експерименти за разпознаване на предмети по форма и цвят:

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 4.6. Разпознаване на червен цвят.



Фигура 4.7. Разпознаване на предмет според детекция на ръбовете.



Фигура 4.8. 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 – 2.0м. 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 – 2.5м. 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 ИЛС 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 ИЛС. 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, ИЛС 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. Бутон за аварийно спиране работи надеждно.

Експерименти и резултати от управлението на работа посредством уеб базиран 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 провеждане на експерименти, постигнатите резултати показват добри постижения в разгледаните проблемни области. Подобрена точност и скорост на движение при навигацията, коригиране на одомеричната 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.

НАУЧНО-ПРИЛОЖНИ ПРИНОСИ

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;

БИБЛИОГРАФИЯ

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 работи - https://ifr.org/img/office/Service_Robots_2016_Chapter_1_2.pdf
5. Статистически данни за сервизните работи - https://ifr.org/downloads/press/02_2016/Executive_Summary_Service_Robots_2016.pdf
6. Статистическите данни на ООН <https://esa.un.org/unpd/popdev/Profilesofageing2015/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 работи - <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. Операциона система ROS - <http://www.ros.org/wiki/>
36. П. Нортън, Мрежи, SAMS 1999
37. Х. Остерло, ТСР/ІР – 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/visi/on/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.

Декларация за оригиналност

Декларирам, че дисертацията съдържа оригинални резултати, получени, при проведени от мен, научни изследвания с подкрепата и съдействието на научния ми ръководител.

Резултатите, които са получени, описани и/или публикувани от други учени, са коректно и подробно цитирани в библиографията.

Настоящият дисертационен труд не е прилаган за придобиване на научна степен в друго висше училище, университет или научен институт.

Подпис:.....
/инж. Денис Чикуртев/

Публикации по дисертационната тема

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. Grueva, 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