



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

Факултет „Електроника и автоматика“

Катедра „Системи за управление“

маг. инж. Васил Любенов Попов

НОВИ ПОДХОДИ ЗА АДАПТАЦИЯ И ОБУЧЕНИЕ ПРИ УПРАВЛЕНИЕ НА ИНТЕЛИГЕНТНИ РОБОТИ

А В Т О Р Е Ф Е Р А Т

на дисертация за придобиване на образователна и научна степен
"ДОКТОР"

Област: 5 Технически науки

Професионално направление: 5.2 „Електротехника, електроника и
автоматика“

Научна специалност: „Системи с изкуствен интелект“

**Научни ръководители: проф. д-р Андон Топалов
доц. д-р Никола Шакев**

ПЛОВДИВ, 2020 г.

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „Системи за управление“ към Факултет „Електроника и автоматика“ ТУ-София филиал Пловдив на редовно заседание, проведено на 29.05.2020г.

Публичната защита на дисертационния труд ще се състои на 05.10.2020 г. от 11:00 часа в зала 4425 на 4-ти корпус на Технически университет – София, Филиал Пловдив, членове на научното жури са:

1. Проф. д-р Михаил Петров – председател
2. Проф. д-р Андон Топалов – научен секретар
3. Проф. д-р Чавдар Дамянов
4. Проф. д-р Диана Цанкова
5. Доц. д-р Веселин Начев

Рецензенти:

1. Проф. д-р Михаил Петров
2. Проф. д-р Чавдар Дамянов

Материалите по защитата са на разположение на интересуващите се в канцеларията на катедра „Системи за управление“ на ТУ-София филиал Пловдив, блок № 4, кабинет № 4242, ул. „Цанко Дюстабанов“ 25.

Дисертантът е редовен докторант към катедра „Системи за управление“ на факултет „Електроника и автоматика“ . Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Васил Попов

Заглавие: Нови подходи за адаптация и обучение при управление на интелигентни работи

Тираж: 15 броя

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Роботиката е непосредствено свързана с индустриалното производство - основно средство за материализиране на резултатите и информацията от кибер-системите към физическия свят, с ключова роля в цифровизираното производство. Едновременно с това развитието на технологиите, разширяването на областите на приложение на роботите, повишената функционалност поставят нови, по-високи изисквания към съвременната роботика. Те трябва да са по-гъвкави, адаптируеми, лесни за програмиране с интуитивен човеко-машинен интерфейс. Ефектите от по-високата функционалност на роботите, по-доброто им интегриране към информационните системи, улеснената комуникация между човек и машина, съвместното извършване на дейности от човека и робота са многопосочни. За производствените предприятия основните предимства се изразяват в по-добрата интеграция и оптимизация на производството, появата на нови услуги, нови продукти и бизнес модели. Успоредно с това все по-широко е навлизането на роботите в непроизводствените сфери: логистични работи, работи – домашни помощници, медицински работи, инспектиращи работи и др. Това развитие създава предпоставки за разработването на нови методи за обучение и адаптация, които да осигурят по-пълноценно включване на роботите не само в структурирана индустриална среда, но и в споделени работни пространства. Интегрирането на нови подходи за адаптация и обучение при управление на интелигентни работи, могат осезаемо да променят съществуващата индустрия, услуги, стандарти чрез обединение и по-пълно използване на човешкия потенциал, технологиите и информацията..

Цел на дисертационния труд, основни задачи и методи за изследване

Дисертационният труд се концентрира върху разработката на алгоритми за обучение и адаптация на интелигентни поведения на автономни работи. Целта е чрез използване на подходи и технологии, свързани с изкуствения интелект, богата сензорна информация, цифрови модели и симулации, големи информационни масиви – да се достигне до по-високи функционални възможности на роботите и до ефективното им прилагане в споделена работна среда.

Изследванията са насочени както към концепциите за използване на колаборативни работи в индустрията, така и към сервизните работи, функциониращи в житейската среда на човека.

За реализацията на тази цел, са идентифицирани няколко конкретни изследователски направления, които съществено биха спомогнали за това, поведенията на роботите в споделена работна среда да бъдат по-гъвкави, по-полезни, по-безопасни:

- i. Разработка и използване на естествени интерфейси за комуникация между човек и безпилотен летателен апарат;

- ii. Разработка и използване на естествени интерфейси за комуникация между човек и индустриален манипулатор;
- iii. Разработка и изследване на алгоритми за обучение и адаптация и възможностите за тяхното приложение при управлението на работи в споделена среда;
- iv. Разработка и изследване на алгоритми за управлението на работи в споделена среда с използването на мултисензорна информация;
- v. Разработка и изследване на алгоритми за управление на многоагентни системи от работи;

Научна новост

Съгласно поставената цел и задачи в дисертационния труд са представени нови подходи за адаптация и обучение при управление на интелигентни работи. Те могат да бъдат обединени в следните групи:

- Разработка и използване на естествени интерфейси за комуникация между човек и робот. Предложени са алгоритми за управление чрез жестове както на индустриален манипулатор, така и на четирироторен мини-хеликоптер. Алгоритмите са базирани на използване на визуална обратна връзка от оператор към робот.
- Разработка и изследване на алгоритми за обучение и адаптация и възможностите за тяхното приложение при управлението на работи в споделена среда. Изследвани са основно възможности за обучение при разпознаване на обекти в изображения, получавани от видео камера. Извличане на количествени и качествени параметри за обекта от изображението и използването на тези параметри в алгоритми за управление на клас колесни мобилни работи.
- Разработка и изследване на алгоритми за управлението на работи в споделена среда с използването на мултисензорна информация. Изследвани са възможностите и са предложени алгоритми за управление на колесни мобилни работи на базата на постъпваща информация за заобикалящата среда от различни сензори: видео камера, лазерен далекомер и сензор Kinect.
- Разработка и изследване на алгоритми за управление на многоагентни системи от мобилни работи. Предложени са алгоритми за два типа групови поведения: i) работи, които изпълняват обща мисия, но всеки от тях се движи по своя независима и предварително неизвестна траектория; ii) работи, които се движат във формация, като в този случай е доказана устойчивостта и сходимостта на движенията във формацията.

Практическа приложимост

Прилагането на подходи от научната област изкуствен интелект при управлението на работи има много широки и на практика неограничени възможности. Предложените в настоящия дисертационен труд алгоритми могат да бъдат успешно интегрирани в комплексни системи за управление на мобилни работи, осигуряващи висока степен на автономност. Те могат да бъдат вложени в системи за управление на сервизни или логистични мобилни работи, които оперират в среда споделяна с хора.

Разработеният алгоритъм за жестови команди за управление на индустриален манипулатор, предоставя по-интуитивна комуникация с устройството в производствена

среда. Предложението на този етап има концептуален характер с интересен потенциал за практическа приложимост, която трябва да се докаже в производствени условия.

Разработените лабораторни системи от мобилни роботи и/или индустриални манипулатори, заедно с интегрираните допълнителни сензорни, комуникационни възможности, алгоритмичното и програмно осигуряване са мощен ресурс за обучение на студентите в областта на системите за управление и приложението на методите на изкуствения интелект в роботиката.

Апробация

Работата е докладвана изцяло в катедра „Системи за управление“, Факултет “Електроника и автоматика” и частично в следните списания и научни конференции:

Списание IFAC- PapersOnLine – (2016, 2018, 2019); Springer Journal of AI & Soc (2018); Springer series “Computational Intelligence 756”, (2018), IEEE International Conference on Intelligent Systems (2016, 2018), International Conference on Advances in Neural Networks and Applications, ANNA (2018), IEEE International Conference on Control, Automation, Robotics and Vision (2018), Международна научна конференция “TexCis” 2019

Резултатите са получили и положителна оценка от рецензенти в отчети по:

- договор No: ДН 11-17/2017 към Фонд „Научни изследвания“;
- договор No:ДФНИ М07-3/2016 към Фонд „Научни изследвания“
- договор (в помощ на докторанти) № 182ПД0026-19 към НИС на ТУ-София.

Публикации

Към дисертационния труд ас. Попов представя 12 публикации. Една от тях е на български език и е самостоятелна. Останалите са на английски език и са в съавторство, като в 5 от тях, той е на първо място. 10 от публикациите са в издания, индексирани от Scopus.

Структура и обем на дисертационния труд

Дисертационният труд е в обем от 193 страници, като включва увод, една глава, идентифицираща проблемите, три глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията, използвана литература и декларация за оригиналност. Цитирани са общо 168 литературни източници на латиница. От тях 164 на английски език и 4 на български език. Работата включва общо 144 фигури и 6 таблици. Номерата на фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.

II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

Първа глава. Преглед на съвременното състояние при алгоритмите за управление на мобилни роботи и манипулатори с използването на техники от изкуствения интелект получени чрез обучение

В първа глава е направен обзор на приложението на техниките на изкуствения интелект в управлението на роботизирани системи, функциониращи в споделена среда с друг робот или човек. Обхванати са подходи за управление на многоагентни сензорни мрежи. Посочени са направленията, в които се съсредоточават научните усилия за

подобряване на ефективността на роботите с използването на методи за адаптация и обучение. Прегледът на научни изследвания в областта на изкуствения интелект и колоборативното поведение дава възможност да се идентифицират проблемите и да се формулират основната цел и направленията спомагащи достигането на целта на дисертационния труд.

Втора глава. Проектиране на алгоритми за откриване и следване на динамични обекти от мобилен робот с използване на невронна мрежа с дълбочинно обучение и лазерен далекомер

Във втора глава са представени теоретични концепции на изкуствения интелект и дълбочинното обучение, което е подобласт на изкуствения интелект. Ще се разгледат софтуерните инструменти използвани по време на извършените опити и описание на конфигурираната мрежа за комуникация. Главата включва описания на кинематиката на използваната мобилна платформа Kuka YouBot, която е с инсталиран ROS, и техническо описание на сензорите прикрепени към нея.

Представени са пет научни труда, свързани с изследванията във втора глава – [A4], [A5], [A6], [A7], [A8].

Разработени са приложения, които разпознават целеви обект, с използването на дълбочинни невронни мрежи, разпознават и проследяват целеви обекти, обработвайки мултисензорна информация. Данните, входящи за обучения невронен модел са от видео поток, осигуряван от RGB камера, прикрепена на мобилна роботизирана платформа Kuka YouBot.

Разработени са серия от помощни приложения за успешното решаване на проблема. Тренировъчните данни за настройка на теглата в математическия модел на конволюционна невронна мрежа, се запазват в паметта на компютър, посредством програма генерирана в програмната среда на Matlab.

Алгоритъм 3: Събиране на обучаващи примери за обучение на конволюционна невронна мрежа

Изискване: инициализация на RGB камера и видео поток от данни

while true

- Отделяне на кадър от видео потока→ *camera.snapshot*;
- Брояч генериращ нарастващо целочислено число за всеки входен кадър → $br = br + 1$;
- Преобразуване на целочислената стойност от брояча в символна поредица → *int2str(br)*;
- Преоразмеряване на входния кадър→
imresize(picture, [424, 512]);
- Запазване в паметта на компютъра на входящия кадър с индивидуално име съответстващо на стойността от брояча→
imwrite(picture, name);

end while

Наличните кадри се разделят на сегменти, които представляват тренировъчния и валидационния набор за математическия модел на невронна мрежа.

Алгоритъм 4: Генериране на тренировъчен и валидационен набор

Изискване: набор от изображения, от които ще се генерира обучаваща извадка

```
while  $i \leq$  брой елементи в текуща директория
- Входно изображение  $\rightarrow$  imread('name.jpg');
- Обхождане на входното изображение от матрица с размерност 128x128;
- Генериране на име на новия файл, съответстващо на символен низ, съдържащ позицията на апертурата в хоризонтално и вертикално направление  $\rightarrow$  strcat(num2str(n),num2str(n_c),num2str(n_r),jpg);
- Запазване в паметта на компютър на всеки регион под апертурата с размерност 128x128  $\rightarrow$  imwrite(save_pic, name);
-  $i = i + 1$ 
end while
```

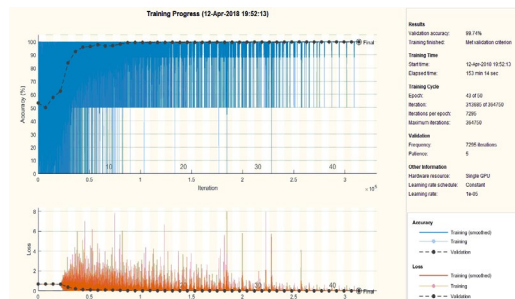
Разпознаването на целевия обект и локализирането му във видимото поле на камерата е посредством подхода за пълзящ прозорец, който обхожда входния кадър и бива оценяван за всяка своя позиция. Тестовите данни са входни за вече обучена конволюционна мрежа, моделът на която е оптимизиран да има ниска тренировъчна и ниска тестова грешки.

Алгоритъм 5: Разпознаване и локализиране на целеви обект

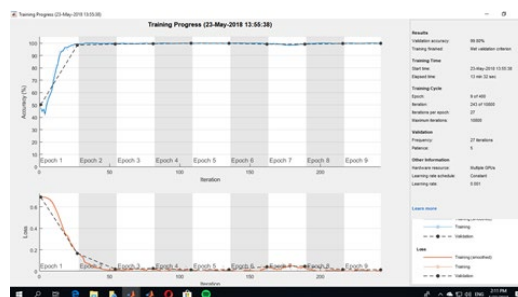
Изискване: .mat файл с обучена конволюционна мрежа, входен видео поток от данни

```
- Инициализация на променливи;
while true
- Прочитане на входен кадър като тримерна матрица;
- Преоразмеряване на входния кадър – 512x424x3;
- Обхождане на кадъра от апертура с размерност 128x128;
  For  $i = 1: 128: 512$ 
    For  $j = 1: 20: 300$ 
      End
    End
  End
- Всеки регион обхванат от апертурата е входна информация за обучената конволюционна мрежа;
- If  $P(y = 1|\theta^{(i)}) > 0.95$  then разпознат е целеви обект
  Else не е разпознат целеви обект
- Определяне горен ляв ъгъл и долен десен ъгъл на целевия регион;
- Построяване на граница на целевия регион
end while
```

Графична репрезентация на обучение на изкуствените невронни мрежи са на Фиг.2.46 и Фиг.2.64, вижда се стойността на грешката по време на обучението (червената графика) и точността на предсказване (синята графика). Различията в резултатите за двете графики е следствие на различния брой примери включени в една обучаваща итерация на невронния модел.

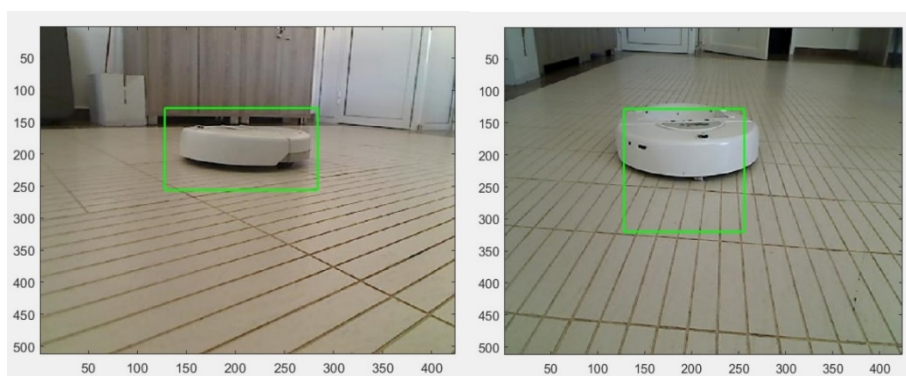


Фиг.2.46 Изменение на грешката и точността на дълбочинния модел по време на трениране.



Фиг.2.64 Точност и грешка при тренирането на конволюционна невронна мрежа.

Резултати от тестването на конволюционната невронна мрежа са дадени на Фиг.2.47 и Фиг.2.68.



Фиг.2.47 Локализиран разпознат обект.



Фиг.2.68 Разпознат и локализиран обект във видимото поле на RGB камерата.

Данни от 2D лазерен далекомер се използват, за засичане на движение и проследяване на подвижния обект. За ефективно и изчислително ненатоварващо засичане и следване на обекта се използва нетривиално използване на данни от лазерен далекомер. Входната информация от далекомера се представя като сегменти, които са считани като отделни обекти за следващите стъпки в предложения алгоритъм. Според имплементацията, първия засечен подвижен сегмент ще бъде следван от мобилния робот, като разстоянието между него и мобилна платформа се запазва референтна. Алгоритъмът стартира с инициализиращо сканиране на пространството пред мобилната платформа и сегментиране на данните от далекомера. Данните от лазера се разделят в брой сегменти $S_i(k)$ за всеки такт от време k . Всеки сегмент е вектор ред с $(j + n + 1)$ елемента, както е показано в (2.64).

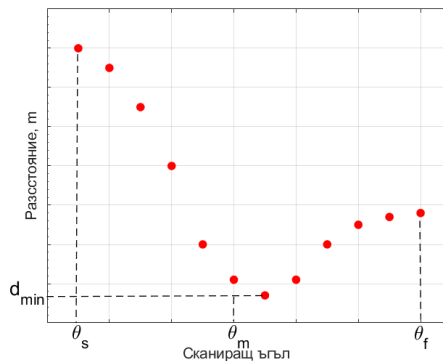
$$S_i(k) = [d(j, k) \ d(j + 1, k) \dots \ d(j + n, k)] \quad (2.64)$$

За да се определи дали разстоянията засечени от два последователни сканиращи лъча са от един и същи сегмент се дефинира разлика в разстоянията Δ , което трябва да е по-малко от стойността $d_r = 0.12m$ (2.65).

$$\Delta = d(j + 1, k) - d(j, k) \leq d_r$$

$$d(j + 1, k) \in \begin{cases} S_i(k), & \Delta \leq d_r \\ S_{i+1}(k), & \Delta > d_r \end{cases} \quad (2.65)$$

Всеки сегмент с брой съседни елементи по-малък от 4 се игнорира. Големината на сегментите може да варира при всяка сканираща итерация k . Сегментацията на участък завършва когато минималното разстояние d_{min} , средната точка θ_m , инициализиращата точка θ_s и финалната точка θ_f са определени - Фиг.2.53.



Фиг.2.53 Разлика в отдалечението на средната точка, инициализиращата точка, финална точка и минимално разстояние за сегмент.

Данните от последователни сканирания на видимото поле се сравняват, за да се определят възможните разлики в сегментите. Първоначално се търси разликата между средните точки $\Delta\theta_m(k)$ за всеки сегмент (2.66). За два последователни итерации средната точка на сегмент ще остане приблизително непроменена и ще бъде отбелязан като сегмент на неподвижен обект. Докато голяма стойност за $\Delta\theta_m(k)$ е признак за потенциално движение, което се потвърждава от промяната и в минималната дистанция $\Delta d_{min}(k)$ за кореспондиращия сегмент (2.67).

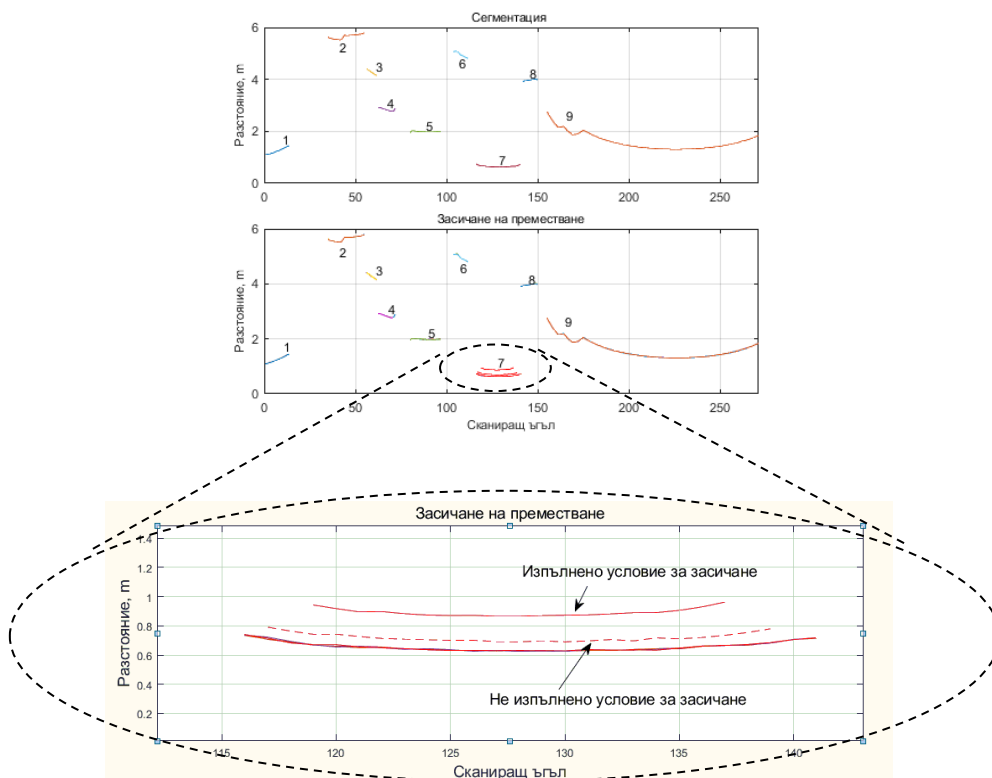
$$\text{if } \Delta\theta_m(k) < \Delta\theta_n \text{ and } \Delta d_{min}(k) > \Delta d_n$$

then $S_i(k)$ is moving

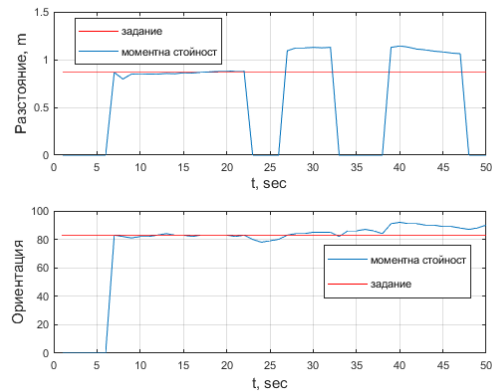
$$\Delta\theta_m(k) = |\theta_m(k) - \theta_m(k - 1)| \quad (2.66)$$

$$\Delta d_{min}(k) = |d_{min}(k) - d_{min}(k - 1)| \quad (2.67)$$

Фиг.2.55 представя етапа на засичане на подвижна цел. Сравнява се позицията на конкретен сегмент – сегмент 7, спрямо стартовото му отместване. При удовлетворяване на дефинираните условия за подвижност от участък, той става целеви във видимото поле на далекомера. Разстоянието до целта и ориентацията ѝ спрямо следящата мобилна платформа са показани на Фиг.2.56.



Фиг.2.55 Засичане на подвижна цел.

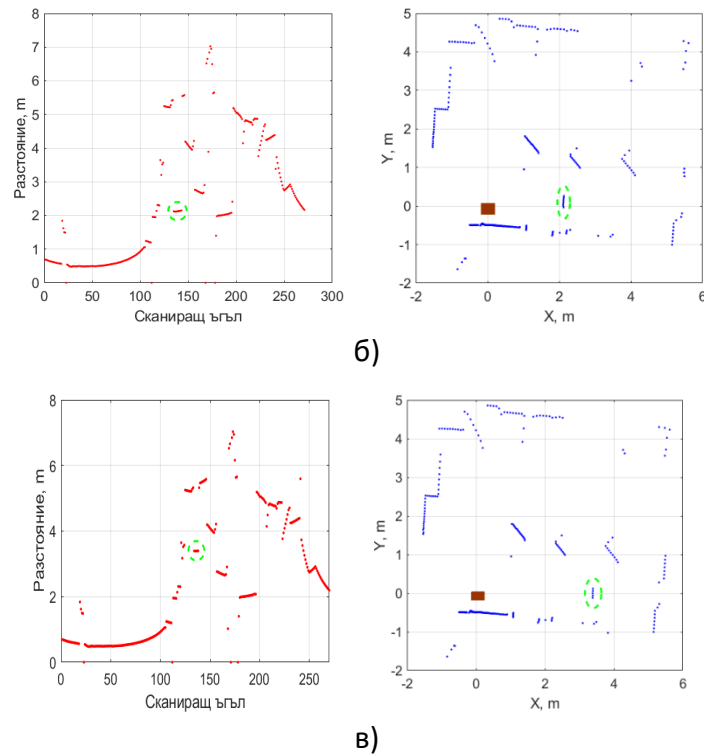


Фиг.2.56 Следене на обект с поддържане на разстояние и ориентация спрямо него.

Проведен е експеримент използващ мобилен робот Kuka YouBot с инсталиран ROS на него, роботът следи движещи се хора с използването на сензори Kinect 360 и 2D лазерен далекоммер. Разстоянието от сензора Kinect до става от скелета се измерва по оста Z_k от координатната система на Kinect. Хоризонталното отместване е по оста X_k , а вертикалното е по оста Y_k . Имайки данни за отместването на обекта по осите на координатна система, алгоритъмът определя относителната позиция на предмет носен от човека. Алгоритъмът е конфигуриран да засича точка от скелета, видима за Kinect 360, намираща се в средата на раменете, именувана *shoulder center* и използва тази точка, за да намери позицията на багажа. Далекоммерът засича целта, когато тя започне да се движи и започва следенето ѝ от мобилната платформа. Мобилния робот следи избраната цел като контролира скоростта си докато поддържа дефинирана ориентация и разстояние с целта. Фиг.2.59 представя предложения подход за следене на обекта, използвайки данни от 2D лазерния далекоммер и Kinect 360 сензор. Две различни представяния на данните са показани: 2D карта на средата в затвореното помещение и данни за разстоянието за всеки един изпратен лъч от далекомера.



а)

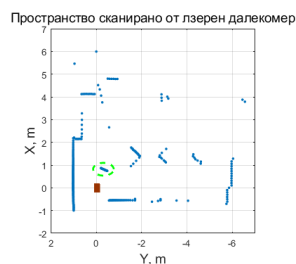


Фиг.2.59 Входни данни от RGB камерата на сензор Kinect 360 и 2D далекомер.

Фиг.2.61 показва инициализиращите условия, видени през бордовата камера, 2D лазерен далекомер и скелетизацията от сензор Kinect 360.



а)



б)

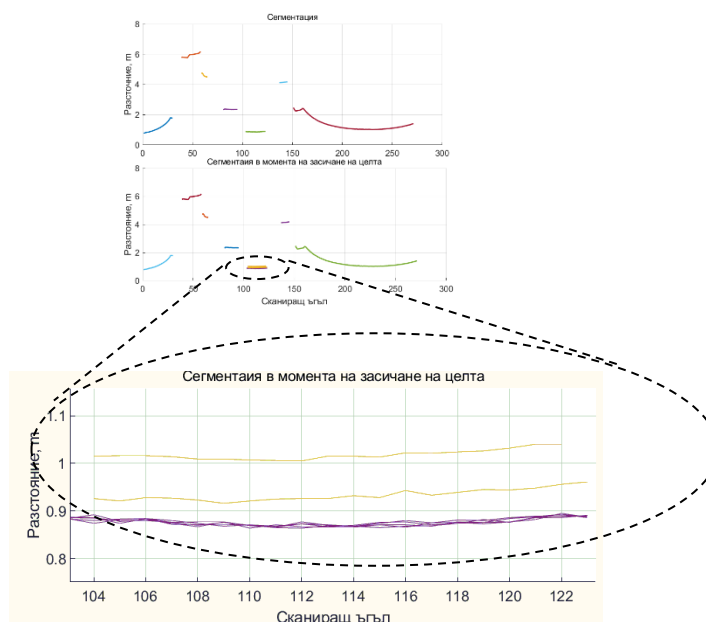


в)

Фиг.2.61 а) кадър от RGB камерата на сензор Kinect 360, б) разпределение на сегментите във видимото поле на лазерен далекомер и позицията на обекта за следене, в) стави от скелета и става, чиито координати се проследяват.

Човекът, премествайки себе си и багажа, започва да бъде проследяван от мобилната платформа. Фиг.2.62 илюстрира инициализиращото положение на

сегментите, и сравнението на позициите им за всяка следваща итерация със стойности за отместването им в инициализиращото състояние, засичането на движение и промяната на позицията на сегмент по време на експеримента.



Фиг.2.62 Движение на сегментите по време на проведения експеримент.

С предложените алгоритми се подобряват интелигентните поведения на мобилни роботизирани системи, функциониращи в споделена работна среда. Използвани са мултисензорни данни и подходи характерни за изкуствения интелект.

Трета глава. Проектиране на алгоритми за управление на многоагентни системи от мобилни роботи

Представените разработки в текущата глава са за генериране на управляващ алгоритъм за колективно поведение на група мобилни роботи. Разгледани са две концепции. Първата включва множество мобилни роботи, всеки от които е автономен, но обменящи данни помежду си чрез мобилна мрежа. Обменените данни позволяват да се изменя поведението на отделните агенти. Втората концепция предполага поведението на отделните роботи да има по-малка степен на самостоятелност и да бъде подчинено на общата задача за движение във формация.

Представени са три научни труда, свързани с изследванията в трета глава – [A3], [A9], [A12].

Разработени са алгоритъм и програма за намиране и обследване на целеви региони от един или множество мобилни роботизирани компоненти. Експериментите са правени с мобилна платформа iRobot Create, хардуерно разширена с Gumstix Verdex. Програмата е генерирана в Simulink, с добавено софтуерно разширение QuarC, позволяващо управление на обекти в реално време.

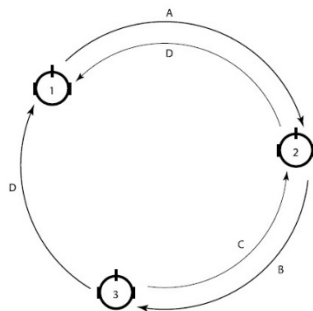
За работи действащи в замърсена среда, имат предварително дефинирани приоритети, които включват три основни етапа.

1. Търсене и откриване;
2. Преориентиране на работа, за да започне обикаляне на региона;

3. Наблюдение на региона от работа.

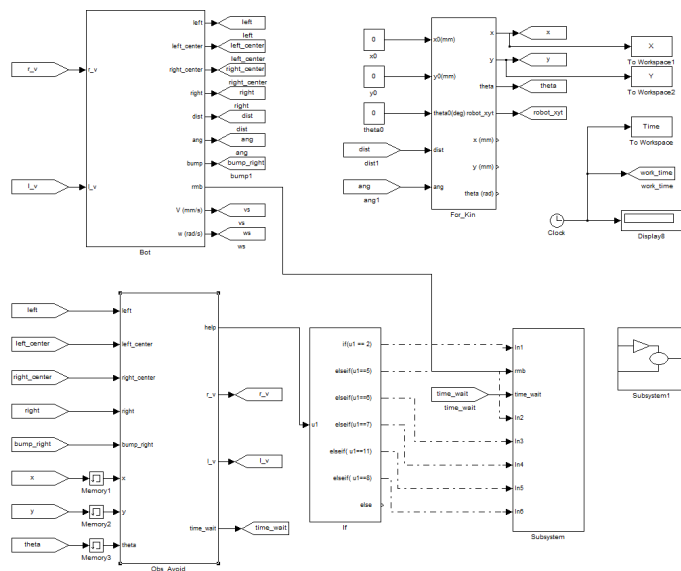
Преминанването между състоянието на работа на мрежата е показано на Фиг.3.1, където работните състояния са отбелязани с букви: A, B, C, D.

- A. Замърсеният регион е открит от участник в безжичната сензорна мрежа;
- B. Данни от сензорите се използват за следване контура на замърсения регион;
- C. Замърсеният регион е извън обхвата на сензорите;
- D. Никои елемент от сензорната мрежа не е открил замърсен регион.



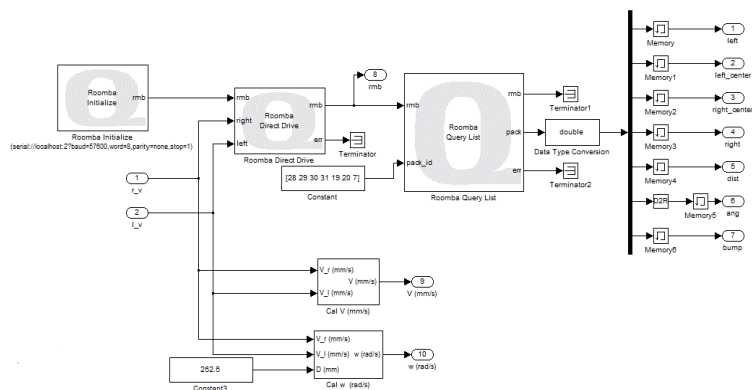
Фиг.3.1 Преходи между работните състояния на участниците в безжична сензорна мрежа.

Базовата конфигурация на предложената система за управление на роботизирани безжични сензорни мрежи се състои от централен (управляващ) host компонент и роботизиран сензорен – iRobot Create (target). Моделът е направен на Simulink, като са използвани функционални блокове за управление на робота. Режимът за изпълнение е External. На Фиг.3.7 е показан цялостния модел на Simulink за изпълнение за зададената задача.



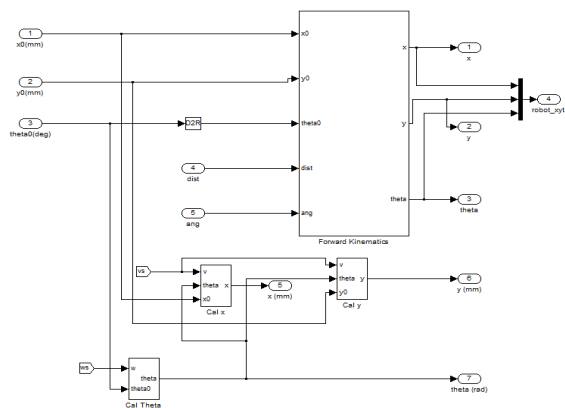
Фиг. 3.7 Общ вид на модела в Simulink.

Съдържанието на подсистемата именувана Bot е на Фиг.3.8.



Фиг.3.8 Подсистема Bot.

На Фиг.3.11 е показано съдържанието на блока за права кинематична задача. Входни данни са инициализиращите стойности за позицията и ориентацията, също и линейното и ъглово отместване между два времеви отрязъка, в които роботът изпраща данни към управляващия компютър.

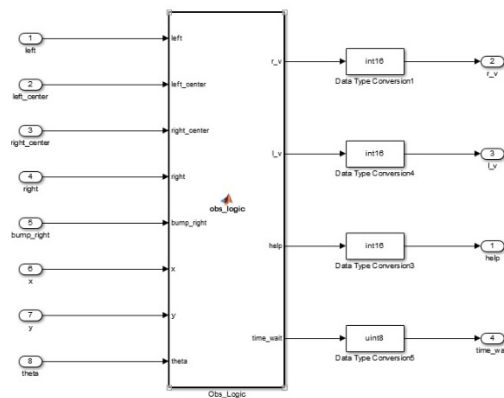


Фиг. 3.11 Подсистема For_Kin.

Алгоритъм 6: Пресмятане на позиция и ориентация на мобилна платформа iRobot Create.

- Присвояване на инициализиращите условия $\rightarrow states = pre_states, x = x0, y = y0, theta = check_angle(theta0);$
- while true**
 - Инкрементиране на позицията по $x, y, theta \rightarrow$
 - $x = states(2) + dist * \cos(theta)$
 - $y = states(3) + dist * \sin(theta)$
 - $theta$
 - $= check_angle(check_angle(states(4))$
 - $+ check_angle(ang))$
- end while**

В подсистемата Obs_Avoid се обработват входните данни от сензора за сблъсък и сензорите за пропадане и се подават управляващи въздействия към колелата, управляващите въздействия са входни променливи за блока Roomba Direct Drive. На Фиг.3.16 е показано съдържанието на подсистемата.



Фиг.3.16 Подсистема Obs_Logic.

Алгоритъм 7: Управлящи въздействия върху колелата на диференциално задвижваната платформа iRobot Create

- Изходни стойности: линейна скорост ляво колело - l_v , линейна скорост дясно колело - r_v , време на изчакване - $time_wait$
- Входни стойности: входен сигнал от сензор за сблъсък - $bump_right$, входен сигнал от краен ляв сензор за замърсяване - $left$, входен сигнал от ляв-централен сензор за замърсяване - $left_center$, входен сигнал от десен-централен сензор за замърсяване - $right_center$, входен сигнал от десен сензор за замърсяване - $right$, входни стойности за позиция по x , y и ориентация - $theta$

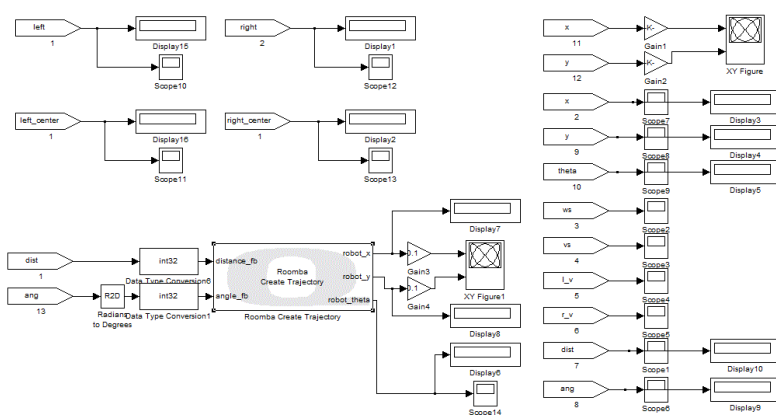
while true

- Има засечен целеви регион
 - If $left < 500 \ \&\& \ left_center < 500$ then $r_v = 75, l_v = 75$;
 - If $bump_right == 1 \ \&\& \ left < 500$ then $r_l = 0, l_v = 0, time_wait = 20$;
 - If $bump_right == 1 \ \&\& \ left < 500 \ \&\& \ left_center < 500$ then $r_l = 0, l_v = 0, time_wait = 20$;
 - If $left_center < 500$ then $r_v = 0, l_v = 50$;
 - If $right < 500 \ \&\& \ right_center < 500$ then $r_v = -100, l_v = 100$;
 - If $right_center < 500$ then $r_v = -100, l_v = 100$;
 - If $left_center < 500 \ \&\& \ right_center < 500$ then $r_v = -50, l_v = 50$;
 - If $left_center < 500 \ \&\& \ right_center < 500 \ \&\& \ bump_right == 1$ then $r_v = -50, l_v = -50, time_wait = 30$;
 - If $right < 500 \ \&\& \ left < 500$ then $r_v = 75, l_v = 10$;
 - If $right < 500 \ \&\& \ left < 500 \ \&\& \ bump_right == 1$ then $r_v = -50, l_v = 50$;
 - If $left < 500 \ \&\& \ right < 500 \ \&\& \ left_center < 500$ then $l_v = 50, r_v = 50$;

- If $left < 500 \ \&\& \ right_center < 500 \ \&\& \ left_center < 500$ then $r_v = -50, l_v = 50$;
- Няма засечен целеви регион
 - If $x < 1000 \ \&\& \ theta == 0 \ \&\& \ bymp_right == 0$ then $l_v = 50, r_v = 50$;
 - If $x < 1000 \ \&\& \ theta == 0 \ \&\& \ bumr_right == 1$ then $r_v = -50, l_v = -50, time_wait = 30$;
 - If $x > 1000 \ \&\& \ x < 1200 \ \&\& \ theta < 0.43$ then $l_v = -50, r_v = 50, time_wait = 20$;
 - If $theta > 1 \ \&\& \ y < 900$ then $l_v = 50, r_v = 50$;
 - If $theta > 1 \ \&\& \ y < 900 \ \&\& \ bump_right == 1$ then $r_v = -50, l_v = -50, time_wait = 30$;
 - If $y \geq 900 \ \&\& \ theta < 2.1$ then $l_v = -50, r_v = 50$;
 - If $x > 300 \ \&\& \ theta > 2.9$ then $l_v = 50, r_v = 50$;
 - If $x > 300 \ \&\& \ theta > 2.9 \ \&\& \ bump_right == 1$ then $r_v = -50, l_v = -50, time_wait = 30$;
 - If $x < 300 \ \&\& \ y > 900 \ \&\& \ theta > 2.9$ then $l_v = -50, r_v = 50, time_wait = 20$;
 - If $x < 220 \ \&\& \ y < 550 \ \&\& \ theta < 1.5$ then $l_v = 50, r_v = 50$;
 - If $x < 300 \ \&\& \ y < 550 \ \&\& \ theta < 1.5 \ \&\& \ bump_right == 1$ then $r_v = -50, l_v = 50$;

while end

Подсистемата Subsystem1 извършва визуализация на всички изследвани параметри, също така там се изчертава и траекторията на база стойностите получени за $x, y, theta$ от правата задача на кинематиката. На Фиг.3.22 е показано съдържанието на тази подсистема.

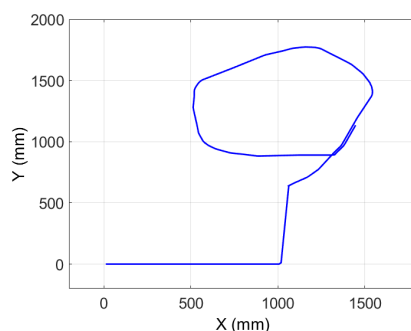


Фиг.3.22 Съдържание на Subsystem1.

От проведените опити са генерирани графики с изходни резултати. Параметрите, които биват наблюдавани са: стойности прочетени на ляв сензор за пропадане, стойности прочетени на ляв централен сензор за пропадане, стойности прочетени за десен сензор за пропадане, стойности прочетени за десен централен сензор за

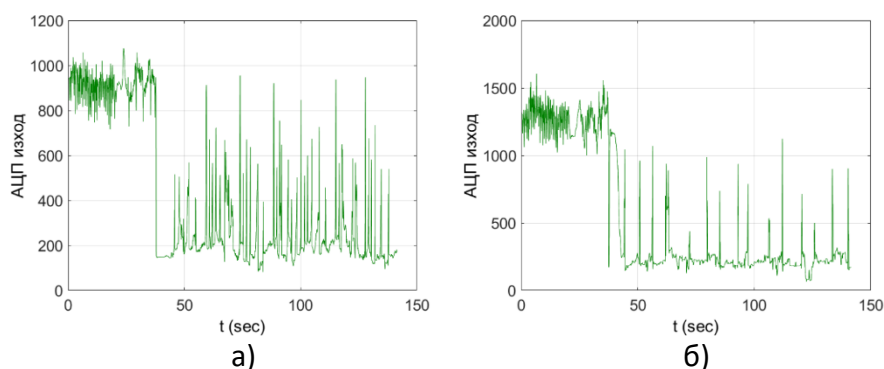
пропадане, стойности за x , y и θ , стойности за ъгловата скорост на лявото и дясното колело.

Експериментът показва получената информация за траекторията на единична мобилна платформа, която е локализираща замърсена площ и е започнала обикалянето ѝ по периметъра, одометрични данни са използвани за изчертаване на изминатия път Фиг.3.23

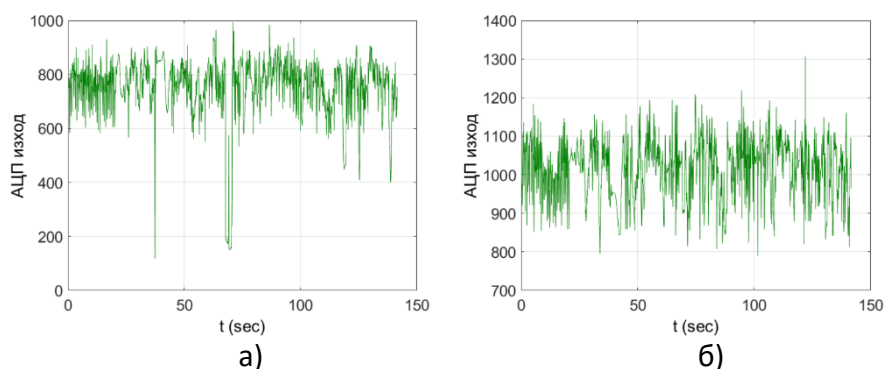


Фиг.3.23 Траектория на първоначално търсене на регион, последвалото му локализирането и започване на проследяването на контура му.

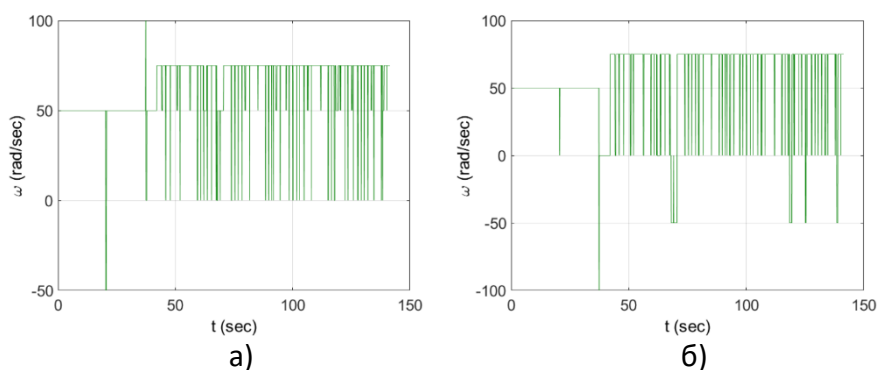
В зависимост от оцветеността на постилащата повърхност се изменя стойността върната от АЦП. По-тъмни нюанси на равнината съответстват на по-ниски стойности върнати към програмата в Simulink.



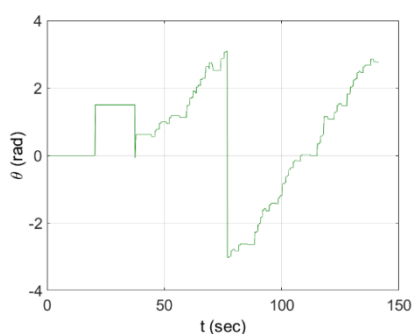
Фиг.3.24 Получени данни от лявата двойка сензори за засичане на замърсената площ.
а) стойностите от ляв централен сензор, б) стойностите за ляв сензор.



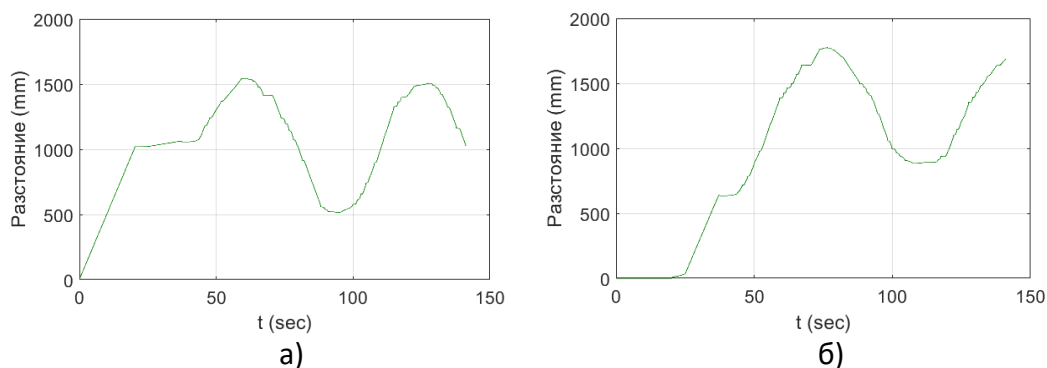
Фиг.3.25 Получени данни от дясната двойка сензори за засичане на замърсената площ.
а) стойностите от десен централен сензор, б) стойностите за десен сензор.



Фиг.3.26 Ъгловата скорост на всяко едно от задвижващите колела на мобилната диференциално задвижвана платформа. а) скоростта на лявото двигателно колело, б) скоростта на дясното.



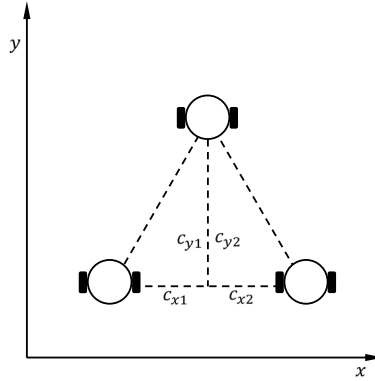
Фиг.3.27 Ориентация на шасито в инерциалната координатна система.



Фиг.3.28 Стойности за отместването на мобилната платформа по x и y в инерциалната координатна система. а) стойности по x , б) стойности по y .

Задачата на алгоритъма за управление на група от роботи, е да следват желаната траектория, като запазят предварително определената пространствена структура на формацията. Приема се, че лидерът на формацията трябва да следва референтната траектория, докато другите роботи запазват позициите си във формацията спрямо позицията на лидера. Подходът се класифицира като следване на лидерска машина. Задачата за управление може и да се модифицира спрямо описания по-горе вариант, ако се приеме, че лидерът на формацията не е физически, а е виртуален робот, следващ референтната траектория. В този случай подходът ще бъде класифициран като подход с виртуална структура. За да се постигне желаното поведение на формацията, ще бъде предложен нов контролер за управление на формация. Проектирането му е

направено на базата на предложен контролер, който се прилага за следене на траектории от отделен (самостоятелен) робот. Модифицирането и използването на контролера за случай с многороботна система осигурява сходимост на траекторията на управляваните роботи към позициите им във формацията. Отместването $c_i = [c_{ix} \ c_{iy} \ c_{i\theta}]^T$, задава желаната разлика между конфигурацията на лидера и конфигурацията на i -тия робот в инерциалната координатна система. Грешката $E_i(t)$, получена при отследяване на траекторията от i -тия робот в глобалната координатна система е дефинирана с израз (3.55). На Фиг.3.36 е показана формацията от роботи.



Фиг.3.36 Отместване на участниците във формацията, спрямо лидерския робот.

$$E_i(t) = \begin{bmatrix} E_{ix} \\ E_{iy} \\ E_{i\theta} \end{bmatrix} = p_L(t) - p_i(t) - c_i = \begin{bmatrix} x_L(t) - x_i(t) - c_{ix} \\ y_L(t) - y_i(t) - c_{iy} \\ \theta_L(t) - \theta_i(t) - c_{i\theta} \end{bmatrix} \quad (3.55)$$

С $e_i(t)$ е означено изразяването на $E_i(t)$ в локалната координатна система на i -тия робот:

$$e_i(t) = \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix} = T_i E_i(t) = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & 0 \\ -\sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E_{ix} \\ E_{iy} \\ E_{i\theta} \end{bmatrix} \quad (3.56)$$

Векторът $\dot{e}_i(t)$, съдържащ производните на грешката.

$$\begin{aligned} \dot{e}_{i1}(t) &= (\dot{x}_L - \dot{x}_i)\cos\theta_i + (\dot{y}_L - \dot{y}_i)\sin\theta_i - (x_L - x_i - c_{ix})\dot{\theta}_i\sin\theta_i \quad (3.59) \\ &+ (y_L - y_i - c_{iy})\dot{\theta}_i\cos\theta_i = \dot{x}_L\cos\theta_i + \dot{y}_L\sin\theta_i - v_i + e_{i2}\dot{\theta}_i \\ &= \dot{x}_L\cos(\theta_L - (c_{i\theta} + e_{i3})) + \dot{y}_L\sin(\theta_L - (c_{i\theta} + e_{i3})) - v_i + e_{i2}\omega_i \\ &= \dot{x}_L[\cos(\theta_L)\cos(c_{i\theta} + e_{i3}) + \sin(\theta_L)\sin(c_{i\theta} + e_{i3})] \\ &+ \dot{y}_L[\sin(\theta_L)\cos(c_{i\theta} + e_{i3}) - \cos(\theta_L)\sin(c_{i\theta} + e_{i3})] - v_i + e_{i2}\omega_i \\ &= (\dot{x}_L\cos(\theta_L) + \dot{y}_L\sin(\theta_L))\cos(c_{i\theta} + e_{i3}) \\ &+ (\dot{x}_L\sin(\theta_L) - \dot{y}_L\cos(\theta_L))\sin(c_{i\theta} + e_{i3}) - v_i + e_{i2}\omega_i \\ &= v_L\cos(c_{i\theta} + e_{i3}) - v_i + e_{i2}\omega_i \end{aligned}$$

$$\begin{aligned}
\dot{e}_{i2}(t) &= -(\dot{x}_L - \dot{x}_i)\sin\theta_i + (\dot{y}_L - \dot{y}_i)\cos\theta_i - (x_L - x_i - c_{ix})\dot{\theta}_i\cos\theta_i \\
&\quad - (y_L - y_i - c_{iy})\dot{\theta}_i\sin\theta_i = -\dot{x}_L\sin\theta_i + \dot{y}_L\cos\theta_i + \dot{x}_i\sin\theta - \dot{y}_i\cos\theta_i - e_{i1}\dot{\theta}_i \\
&= -\dot{x}_L\sin(\theta_L - (c_{i\theta} + e_{i3})) + \dot{y}_L\cos(\theta_L - (c_{i\theta} + e_{i3})) - e_{i1}\omega_i \\
&= -\dot{x}_L[\sin(\theta_L)\cos(c_{i\theta} + e_{i3}) - \cos(\theta_L)\sin(c_{i\theta} + e_{i3})] \\
&\quad + \dot{y}_L[\cos(\theta_L)\cos(c_{i\theta} + e_{i3}) + \sin(\theta_L)\sin(c_{i\theta} + e_{i3})] - e_{i1}\omega_i \\
&= (\dot{x}_L\cos(\theta_L) + \dot{y}_L\sin(\theta_L))\sin(c_{i\theta} + e_{i3}) \\
&\quad + (-\dot{x}_L\sin(\theta_L) + \dot{y}_L\cos(\theta_L))\cos(c_{i\theta} + e_{i3}) - e_{i1}\omega_i \\
&= v_L\sin(c_{i\theta} + e_{i3}) - e_{i1}\omega_i \\
\dot{e}_{i3}(t) &= \dot{\theta}_L - \dot{\theta}_i = \omega_L - \omega_i
\end{aligned}$$

Линейната и ъглова скорост на i -тия робот се изчисляват според закона за управление показан в (3.60).

$$q_i(t) = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} v_L \cos(c_{i\theta} + e_{i3}) + k_1 e_{i1} \\ \omega_L + k_2 v_L e_{i2} + k_3 v_L \sin(c_{i\theta} + e_{i3}) \end{bmatrix} \quad (3.60)$$

Законът реализира управление с обратна връзка, като в структурата му участват грешките при траекторно следене, изразени в локалната координатна система. Предложеният алгоритъм е модификация на известен в литературата закон за управление на Канаяма, която позволява приложението му за управление във формация. За анализ на устойчивостта нека дефинираме следната кандидат функция на Ляпунов $V(t)$:

$$V_i(t) = \frac{1}{2}(e_{i1}^2 + e_{i2}^2) + \frac{1 - \cos(c_{i\theta} + e_{i3})}{k_{i2}} \quad (3.61)$$

Производната на уравнение (3.61) се изчислява:

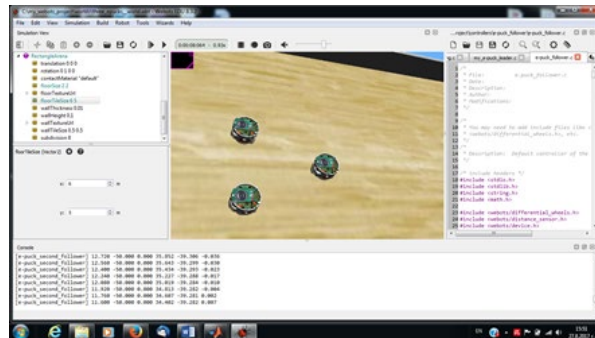
$$\dot{V}_i = \dot{e}_{i1}e_{i1} + \dot{e}_{i2}e_{i2} + \dot{e}_{i3}\frac{\sin(c_{i\theta} + e_{i3})}{k_2} \quad (3.62)$$

Полагайки уравнения (3.59) в уравнение (3.62) и прилагайки закона за управление (3.60) при изчисляването на $v_i(t)$ и $\omega_i(t)$ получаваме:

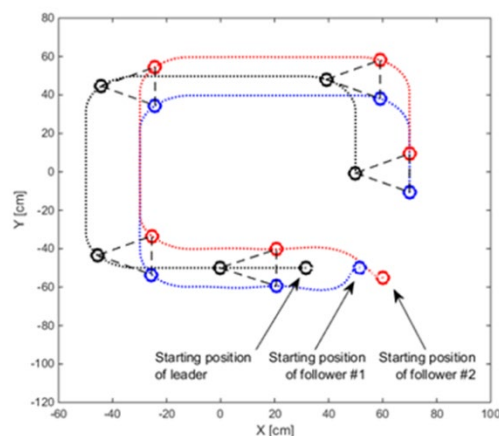
$$\begin{aligned}
\dot{V}_i &= [v_L \cos(c_{i\theta} + e_{i3}) - v_i + e_{i2}\omega_i]e_{i1} + [v_L \sin(c_{i\theta} + e_{i3}) - e_{i1}\omega_i]e_{i2} \\
&\quad + [\omega_L - \omega_i]\frac{\sin(c_{i\theta} + e_{i3})}{k_2} = -k_1 e_{i1}^2 - k_3 v_L \frac{\sin^2(c_{i\theta} + e_{i3})}{k_2} \leq 0 \quad (3.63)
\end{aligned}$$

Ако линейната скорост $v_L(t)$ на водещия робот приема неотрицателни стойности и коефициентите на регулатора k_1, k_2, k_3 са избрани също неотрицателни, то неравенството (3.63) ще бъде винаги изпълнено и системата ще бъде устойчива около $e_i(t) = 0$. От това следва, че траекторията на I – тия робот ще сходя към траекторията на водещия робот, отместена с предварително зададените стойности $c_i = [c_{ix} \ c_{iy} \ c_{i\theta}]^T$. Проведени са симулационни експерименти с използването на специализиран за симулация на роботехнически системи софтуерен пакет Webots. Пакетът представлява 3D симулатор на автономни роботи, позволяващ моделиране на кинематиката и динамиката на робота, моделиране на работното пространство, моделиране на взаимодействието между средата и робота (сили на триене, данни от сензори със съответни смущения и неточности и др.). На базата на създадените модели е извършена симулация, която изчислява траекторията и сигналите от сензорите на

роботите във формацията. Симулацията дава надеждна и достатъчно достоверна информация за поведението на роботите.

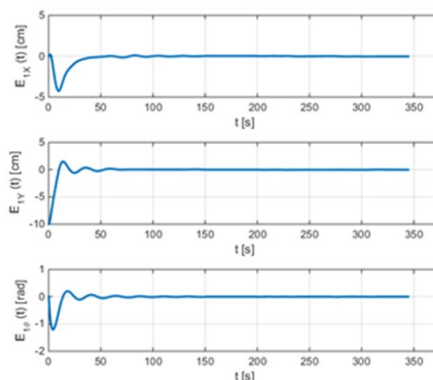


Фиг. 3.37 Симулационна среда Webots, използваща модели на мобилни роботи E-Ruck за симулацията.

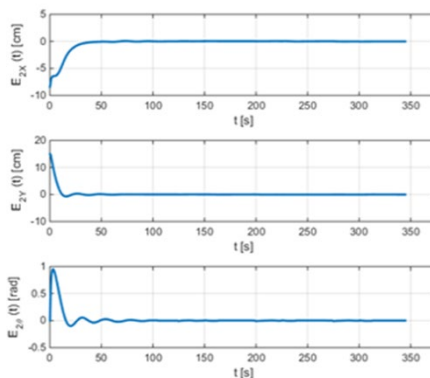


Фиг. 3.38 Симулационни траектории за робота лидер (черен цвят) и за роботите последователи (червен и син цвят).

Траекториите на робота-лидер и двата следващи го робота са показани на Фиг. 3.38. Може да се види, че при използване на предложения закон за управление, и двата следващи робота поддържат желаното положение спрямо лидера, като първоначално въведената грешка в позициите им клони към нула. Изменението на грешките по отделните координати и на двамата последователи е показано съответно на Фиг. 3.39 и Фиг. 3.40.



Фиг.3.39 Изменение на грешките E_{1x} , E_{1y} , $E_{1\theta}$ при движение на първия робот – последовател.



Фиг.3.40 Изменение на грешките E_{2x} , E_{2y} , $E_{2\theta}$ при движението на втория робот-последовател.

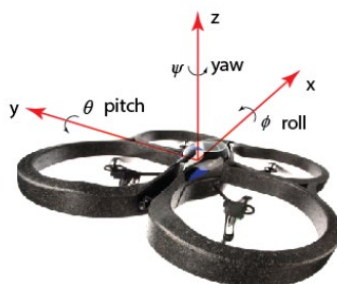
За проведените експерименти е създадена програма за управление и изследване на замърсена среда от робота. Проведените експерименти потвърждават очакваните резултати за поведението на мобилната роботизирана платформа iRobotCreate. Целта на алгоритъма за управление на формация от множество роботи е да осигури надеждно управление за траекторно следене на мобилните роботи. Стратегията за движението на формацията съдържа две задачи: 1) лидерът на формацията трябва да отследява желаната траектория и 2) роботите във формацията трябва да следват лидера, като същевременно запазват предварително определената структура на формацията. Сходимостта на грешката е теоретично доказана за управление при траекторно следене за формация от роботи. Представените симулационни изследвания потвърждават работоспособността и устойчивостта на системата.

Четвърта глава. Проектиране на алгоритми за жестово управление на роботи

В текущата глава ще бъде разгледан: безпилотния летателен апарат AR. Drone 1.0, сензорът Kinect, използваните софтуерни инструменти за създаване и прилагане на алгоритмите за управление, конфигурацията на създадената мрежа за комуникация за предаване на управляващи сигнали към летателното устройство, дефиниция на управляващите жестове за промяна позицията на квадрокоптера в пространството. Проведени експерименти за жестово управление на четирироторен летателен апарат. Показани са резултатите от разработването на алгоритъм за жестово управление на индустриален манипулатор Mitsubishi.

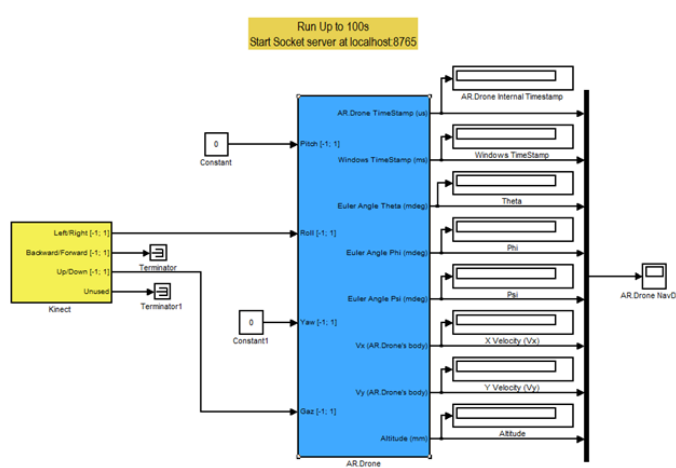
Представени са четири научни труда, свързани с изследванията в четвърта глава – [A1], [A2], [A10], [A11].

Позицията на квадрокоптера се определя от координатите по трите оси - x, y, z определящи позицията от центъра на гравитация, ориентацията от ротацията на около трите оси - ψ, θ, ϕ Фиг.4.5 Контролният сигнал за движение нагоре u е нормала към равнината от витлата и моментите $\tau_\theta, \tau_\psi, \tau_\phi$.



Фиг.4.5 Прикрепената координатна система към летателния апарат.

Комуникацията между AR.Drone 1.0 и управляващия компютър се осъществява с изпращане на управляващи въздействия към хеликоптера в интервала $[-1, +1]$. Фиг.4.34. показва модела на Simulink за прехвърляне данни към дрона и от дрона. Приложението приема данните изпратени от компютъра с прикачения към него сензор Kinect 360. Моделът се състои от два клиентски блока, единия приема данните от компютъра с Kinect 360, а другия данните от квадрокоптера, който е създал ad-hoc мрежа. Всеки блок клиент има направени настройки, които отговарят на предназначението.



Фиг.4.34 Модел за трансфер на информация.

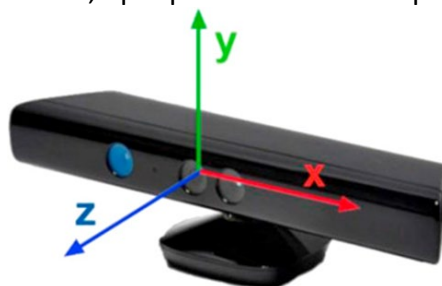
Фиг.4.11 показва трансфера на данни между устройствата участващи в изградената система за жестово управление.



Фиг.4.11 Структура на двупосочна комуникация между дрон и управляващ компютър.

Разработка прилагаща управление на мобилна платформа, използвайки мрежова комуникация и среда за програмиране Visual Studio е [162]. Сензорът Kinect 360 засича движенията на човек и разпознава жестовете. На лаптопа има приложение, което

разпознава жестовете за нагоре, надолу, наляво, надясно, напред, назад. Програмата е написана на C#, а средата за разработка е Microsoft Visual Studio 2010. На всеки жест отговарят три стойности, които се изпращат към управляващия компютър, който препраща управляващи сигнали към квадрокоптера. На компютъра има мрежови клиент, реализиран с функционални блокове на QuarC. Информацията, която се предава са стойности -1, 1, 0. Стойностите -1 и 1 описват крайни за всяка двойка команди, а 0 е спиране на движението. Разработеното приложение използва информация идваща от потока SkeletonStream, който е съставен на дълбочинния поток. Положението на точки от скелета се определя като отместване по трите оси на референтна координатна система, прикрепена към сензора Kinect 360 Фиг.4.16.



Фиг.4.16 Координатна система на Kinect 360.

Входната информация от Kinect 360 е непрекъсната и постоянно постъпваща в потоците от данни към написаното конзолно потребителско приложение на C#. В него освен разпознаването на жестове, установява и комуникация между компютъра към който е свързан сензора Kinect 360 и компютъра който изпраща сигнали към квадрокоптера, използвайки QuarC. В Табл.4.1 са показани условията за команди наляво, надясно и спиране.

Координати	Ъгли	Команда
handRight.Position.Y > elbowRight.Position.Y	Градусът образуван от лакетната става на дясната ръка трябва да е по-малък от 65 градуса.	Дясно
handRight.Position.Y > elbowRight.Position.Y	Градусът образуван от лакътната става на дясната ръка трябва да е по-голям от 115 градуса.	Ляво
handRight.Position.Y > elbowRight.Position.Y	Градусът образуван от лакътната става на дясната ръка трябва да е по-голям от 65 градуса и по-малък от 115 градуса.	Нулева

Табл.4.1 Условия за команди наляво, надясно и спиране.

Табл.4.2 показва условията за команда напред, назад и спиране.

Координати	Ъгли	Команди
handLeft.Position.Y > elbowLeft.Position.Y	Градусът образуван от лакътната става на лявата ръка трябва да е по-голям от 115 градуса.	Напред
handLeft.Position.Y > elbowLeft.Position.Y	Градусът образуван от лакътната става на лявата ръка трябва да е по-малък от	Назад

	65 градуса.	
handLeft.Position.Y > elbowLeft.Position.Y	Градусът образуван от лакътната става на лявата ръка трябва да е по-малък от 115 градуса и по-голяма от 65 градуса.	Нулева

Табл.4.2 Условия за команди за напред, назад и спиране.

Табл.4.3 показва условията за команда нагоре, надолу и спиране.

Координати	Ъгли	Команди
head.Position.Z - handRight.position.Z > 0.3; shoulderRight.Position.Y < handRight.Position.Y	Ъгълът образуван от лакътната става на дясната ръка трябва да е по-малък от 40 градуса, а ъгълът между лявото рамо, дясното рамо и десния лакът да е по-малък от 70 градуса.	Нагоре
head.Position.Z - handRight.position.Z > 0.3; shoulderRight.Position.Y > handRight.Position.Y	Ъгълът образуван от лакътната става на дясната ръка трябва да е по-голям от 40 градуса, а ъгълът между лявото рамо, дясното рамо и десния лакът да е по-голям от 70 градуса.	Надолу
head.Position.Z - handRight.position.Z > 0.3	Ъгълът образуван от лакътната става на дясната ръка трябва да е по-малък от 40 градуса, а ъгълът между лявото рамо, дясното рамо и десния лакът да е по-малък от 70 градуса.	Нулева

Табл.4.3 Условие за команди нагоре, надолу и нулиране.

Алгоритмично представяне на приложението в C# за комуникация с компютър и разпознаване на управляващи жестове е показано в Алгоритъм 8.

Алгоритъм 8: Разпознаване на управляващи жестове и предаване на данни към управляващ компютър.

Изискване: SDK със софтуерни инструменти за обработка на данни от сензора Kinect, налични хардуерни мрежови компоненти

- Програмно инициализиране на сензора Kinect;

while connection == **true**

- Обработка на дълбочинния поток от данни, постъпващ от сензора Kinect;

- **If – else** логика за разпознаване на жестове, отговарящи на дефинирани условия;

- Изпращане на индекс, дефиниращ разпознатия жест, към управляващия компютър;

- Получаване на маркер за успешно получаване на индекс за разпознат жест;

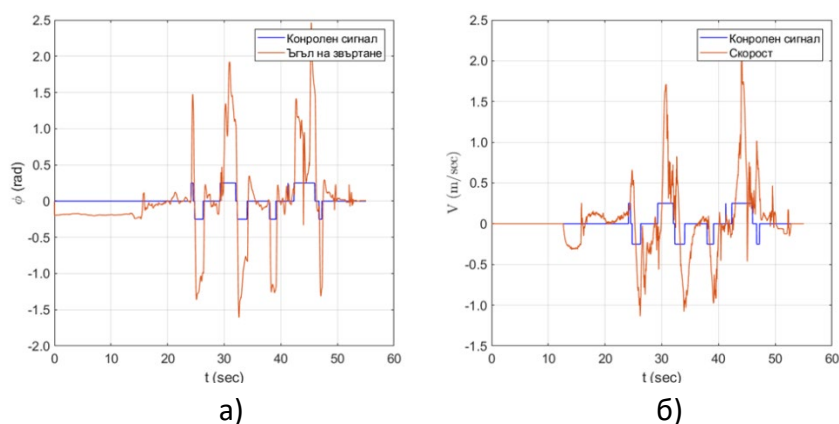
end while

while connection == false

- Изчакване до възстановяването ѝ или изчакване на откриването ѝ;

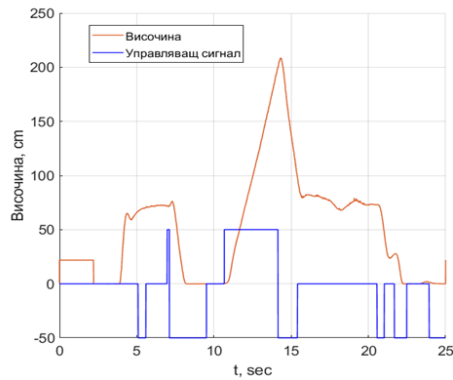
end while

Комуникацията на дрона с управляващия компютър е посредством WiFi връзка, която е ad-hoc и е иницирирана от летателния апарат [147]. Проведените експерименти потвърждават работоспособността на системата. Необходимост за успешното преминаване на опитите е да има достатъчно пространство, където да се осъществи полета и да няма пряка опасност за хората намиращи се в околност. Друго важно условие е човекът, който управлява да има добра видимост към квадрокоптера, и сензорът Kinect 360 да бъде позициониран на място, откъдето възможно най-добре ще обхваща подаващия команди. Ниското тегло на дрона, го прави лесно податлив на въздушните течения, най-добрата среда за провеждане на опитите е в затворено пространство. Дългия по време команден жест ще доведе до нарастване на скоростта на движение на летящия обект и ще намали времето за реакция на управляващия. В процеса на движение в обекта се събира кинетична енергия, заради която движението по зададена траектория продължава, дори след като е подаден нов контролиращ сигнал и той е бил приет. На Фиг.4.35 се виждат графиките за данните от канала за управление по посока. Измерени са скоростта и ъгъла на завъртане.



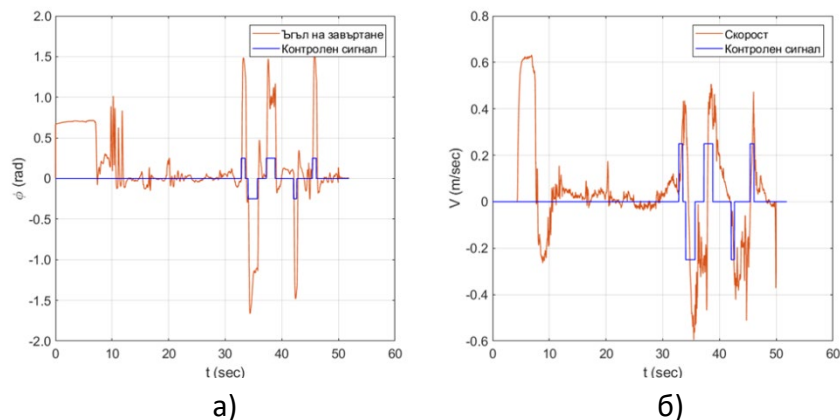
Фиг.4.35 Управлящо въздействие, скорост и ъгъл на завъртане при хоризонтално движение.

На Фиг.4.36 се вижда графиката от същия експеримент обхващаща канала за височина.



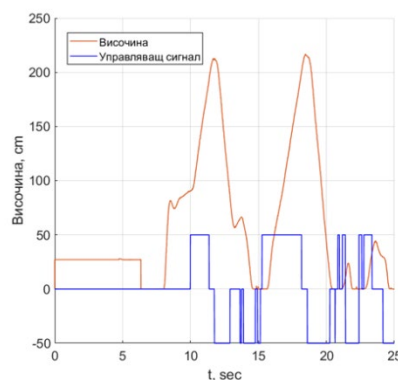
Фиг.4.36 Управляващо въздействие и достигната височина.

На Фиг.4.37 е графиката за данните от канала за управление по посока при втори проведен експеримент. Измерени са линейната скорост и ъгъла на завъртане.



Фиг.4.37 Управляващо въздействие, скорост и ъгъл на завъртане при хоризонтално движение.

На Фиг.4.38 е показана графиката за височината за същия опит.



Фиг.4.38 Управляващо въздействие и достигната височина.

За разработката на комуникационния интерфейс между човек и индустриален манипулатор, се използва способността на сензора Kinect да следи определени стави от скелетната структура на хората позиционирани във видимото му поле. Измерването на отместването на предварително определени точки се използва за дефиниране на командни жестове, които се изпращат към индустриален SCARA манипулатор

Mitsubishi. Предложеният интерфейс се причислява за категория манипулатори, които позволяват водене с ръка. В предложения подход за управление на индустриален манипулатор, засичането на точки от скелета става спрямо отместването на тези точки по осите x , y и z на прикрепената координатна система за сензора – Фиг.4.16. Разработеното приложение за жестово разпознаване е направено в средата на Matlab. Трансферът на управляващи команди от приложението на Matlab към контролера на робота се осъществява посредством TCP/IP комуникационен протокол и наличната инфраструктура в лабораторията. Комуникацията между устройствата е двупосочна, към манипулатора се изпращат управляващи сигнали, а той връща позицията на работния си инструмент в картезиански координати. Контролните жестове са 7, в Табл.4.5. са показани съответстващите управляващи команди на разпознатите жестове.

,2'	Ляво
,1'	Дясно
,6'	Напред
,7'	Назад
,4'	Горе
,5'	Долу
,3'	Нулева команда

Табл.4.5 Управляващи сигнали.

Алгоритъм 9: Разпознаване на управляващ жест и комуникация с индустриален манипулатор

Изискване: SDK с налични инструменти за обработка на данни от сензор Kinect, постъпващи в потребителско приложение на Matlab, хардуерни мрежови компоненти

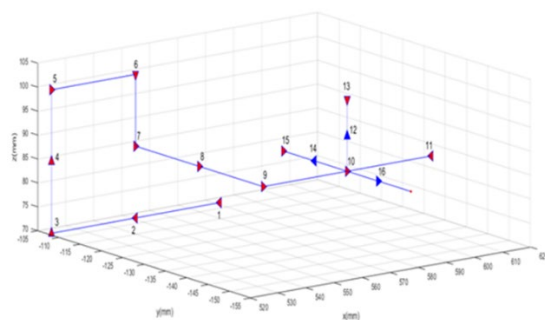
- Инициализация на синхронна комуникация между Matlab приложение и индустриален манипулатор;

while true

- Приемане на данни изпратени от манипулатора: изчакване жест, получен управляващ жест, позиция на работния орган по осите x, y, z .
- Разпознаване на управляващи жестове;

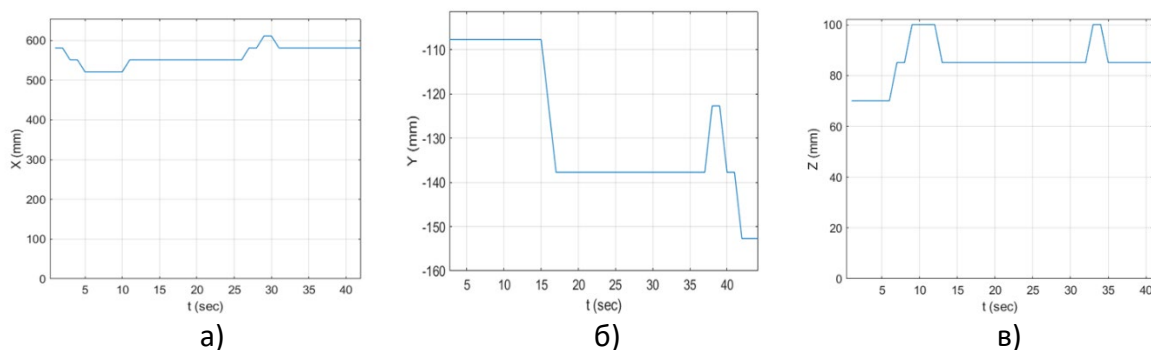
end while

След изпълнението на всяко преместване, роботът спира, чакайки нулиращ управляващ сигнал, след който може да бъде зададен следващата команда за позициониране.



Фиг.4.40 Траектория на работния орган на манипулатора.

Изменение позицията на актуатура по всяка една от осите в показано на Фиг.4.41.



Фиг.4.41 Отместване на работния орган по осите x , y , z .

В Табл.4.6 са показани позицията на върха на работния орган на манипулатора и командните сигнали изпратени към контролера на работа.

Управляващ сигнал	Позиция по $X(\text{mm})$	Позиция по $Y(\text{mm})$	Позиция по $Z(\text{mm})$
Стартова позиция	580.8680	-107.6990	70.1221
'2'	550.8680	-107.6990	70.1221
'3'	550.880	-107.6990	70.1221.
'2'	520.8680	-107.6990	70.1221
'3'	520.8680	-107.6990	70.1221
'4'	520.8680	-107.6990	85.1221
'3'	520.8680	-107.6990	85.1221
'4'	520.8680	-107.6990	100.1221
'3'	520.8680	-107.6990	100.1221
'1'	550.8680	-107.6990	100.1221
'3'	550.8680	-107.6990	100.1221
'5'	550.8680	-107.6990	85.1221
'3'	550.8680	-107.6990	85.1221
'6'	550.8680	-122.6990	85.1221
'3'	550.8680	-122.6990	85.1221
'6'	550.8680	-137.6990	85.1221
'3'	550.8680	-137.6990	85.1221
'3'	550.8680	-137.6990	85.1221
'3'	550.8680	-137.6990	85.1221
'1'	580.8680	-137.6990	85.1221

'3'	580.8680	-137.6990	85.1221
'1'	610.8680	-137.6990	85.1221
'3'	610.8680	-137.6990	85.1221
'2'	580.8680	-137.6990	85.1221
'3'	580.8680	-137.6990	85.1221
'4'	580.8680	-137.6990	100.1221
'3'	580.8680	-137.6990	100.1221
'5'	580.8680	-137.6990	85.1221
'3'	580.8680	-137.6990	85.1221
'7'	580.8680	-122.6990	85.1221
'3'	580.8680	-122.6990	85.1221
'6'	580.8680	-137.6990	85.1221
'3'	580.8680	-137.6990	85.1221
'6'	580.8680	-152.6990	85.1221

Табл.4.6 Позиция на работния орган.

Проведените експерименти за управление на четирироторен летателен апарат и на индустриален SCARA манипулатор Mitsubishi доказва приложимостта на жестовия интерфейс за контрол на устройства. Повишавайки съвместимостта на управляващите сигнали с операторските движения, се постига по-добра кооперираност между човек и машина.

Приноси в дисертационния труд

Принос 1: Разработени са и са експериментално тествани алгоритъм и програма за невронно разпознаване на обекти чрез дълбочинно обучение на конволюционна невронна мрежа въз основа на създадена база от изображения.

Принос 2: Разработени са и са експериментално тествани метод, алгоритъм и програма за откриване и следване от клас мобилни роботи на подвижни обекти по данни, получавани в реално време от 2D лазерен далекомер.

Принос 3: Предложен е нов метод за управление на малък безпилотен летателен апарат - четирироторен хеликоптер (дрон) с приложение при обследване (инспектиране) на трудно достижими от човека места в индустриални обекти. Управляващата схема позволява реализирането на два отделни режима на полет: 1) автоматично управляван (автономен) полет и 2) полет в полу-автономен режим, чрез възприемане и изпълнение на жестови команди от оператор.

Принос 4: Разработени са алгоритъм, програмна и физическа реализация за жестов интерфейс за управление от оператор на индустриален манипулатор тип SCARA на фирма Mitsubishi Electric в споделена работна среда.

Принос 5: Разработен е алгоритъм за управление на мобилни роботизирани възли от безжична сензорна мрежа при търсене и откриване на замърсени площи в средата, където мрежата е разположена.

Принос 6: Разработен е устойчив алгоритъм за траекторно управление на формация от колесни мобилни роботи. Доказана е сходимост на грешката към нула. Представени са симулационни експерименти в програмната среда WeBots.

Списък с публикации за дисертационния труд

- [A1] **В. Л. Попов** (2013) "Създаване на приложение на C# за разпознаване на управляващи жестове посредством Microsoft Kinect", Национална Студентска Научно – Техническа Конференция 2013, Технически Университет – София
- [A2] **V. L. Попов**, K. B. Shiev, A. V. Topalov, N. G. Shakev, S. A. Ahmed (2016) "Control of the Flight of a Small Quadrotor using Gestural Interface", In Proceedings of IEEE 8th International Conference on Intelligent Systems, 4-6 September, Sofia, Bulgaria, ISBN 978-1-5090-1353-1, pp: 622-628, 2016, DOI: 10.1109/IS.2016.7737492
- [A3] S. A. Ahmed, **V. L. Попов**, A. V. Topalov and N. G. Shakev (2016) "Perimeter Detection and Surveillance of Polluted Areas by Robotized Agents in a Hybrid Wireless Sensor Network", IFAC- PapersOnLine, Volume 49, Issue 29, 2016, Pages 247-252, ISSN: 24058963, DOI: 10.1016/j.ifacol.2016.11.061 (SJР 0.26)
- [A4] N. G. Shakev ; S. A. Ahmed ; **V. L. Попов** ; A. V. Topalov „Recognition and Following of Dynamic Targets by an Omnidirectional Mobile Robot using a Deep Convolutional Neural Network "9th IEEE International Conference on Intelligent Systems, IS 2018, Funchal, Madeira, Portugal, September 25-27, 2018
- [A5] **V. L. Попов**, N. G. Shakev, S. A. Ahmed, and A. V. Topalov, (2018) "Recognition of Dynamic Targets using Deep Convolutional Neural Network" In Proc. of 2018 International Conference on Advances in Neural Networks and Applications, ANNA 2018, St. Konstantin and Elena Resort, Bulgaria, pp. 90-95. ISBN 978-3-8007-4756-6 , © 2018 VDE VERLAG GMBH · Berlin · Offenbach
- [A6] S. A. Ahmed, A. V. Topalov, N. G. Shakev and **V. L. Попов** (2018) "Model-Free Detection and Following of Moving Objects by an Omnidirectional Mobile Robot using 2D Range Data", IFAC- PapersOnLine, Volume 51, Issue 22, 2018, Pages 226-231, ISSN: 2405-8963, DOI: 10.1016/j.ifacol.2018.11.546 (Proceedings of 12th IFAC Symposium on Robot Control (SYROCO 2018), 27 – 30 Aug., 2018, Budapest, Hungary) (SJР 0.26)
- [A7] **V. L. Попов**, S. A. Ahmed, A. V. Topalov and N.G.Shakev (2018) "Development of Mobile Robot Target Recognition and Following Behavior using Deep Convolutional Neural Network and 2D Range Data", IFAC-PapersOnLine, Volume 51, Issue 30, 2018, Pages 210-215, ISSN: 2405- 8963, DOI: 10.1016/j.ifacol.2018.11.288 (Proceedings of 18th IFAC Conference Technology, Culture and International Stability (TECIS 2018), 13-15 Sept., 2018, Baku, Azerbaijan), (SJР 0.26)
- [A8] **V. L. Попов**, S. A. Ahmed, N. G. Shakev and A. V. Topalov (2018) "Detection and Following of Moving Targets by an Indoor Mobile Robot using Microsoft Kinect and 2D Lidar Data", In Proceedings of 15th International Conference on Control, Automation, Robotics and Vision (ICARCV 2018), 18 - 21 Nov. 2018, Singapore, art. no. 8581231, pp. 280-285; DOI: 10.1109/ICARCV.2018.8581231
- [A9] S. A. Ahmed, **V. L. Попов**, A. V. Topalov, N. G. Shakev, (2018) "Environmental monitoring using a robotized wireless sensor network", Journal of AI & Soc (2018) 33:207-214, Volume 33, Number 2, ISSN: 0951-5666 (Print) 1435-5655 (Online), DOI 10.1007/s00146-018- 0815-y (SJР 0.227)
- [A10] N. G. Shakev, S. A. Ahmed, A. V. Topalov , **V. L. Попов**, and K. B. Shiev (2018) "Autonomous Flight Control and Precise Gestural Positioning of a Small Quadrotor", Chapter in the book "Learning Systems: From Theory to Practice", Sgurev V., Piuri V., Jotsov V., (Eds.), Springer series "Computational Intelligence 756", May 2018, ISSN: 1860949X, (SJР 0.18); DOI: 10.1007/978-3-319-75181-8_9

[A11] **V. L. Popov**, S. A. Ahmed, N. G. Shakev, A. Topalov, (2019) "Gesture-based Interface for Real-time Control of a Mitsubishi SCARA Robot Manipulator", 19th IFAC Conference on Technology, Culture and International Stability TECIS 2019, 26 - 28 September, 2019, Sozopol, Bulgaria (IFAC-PapersOnLine – SJR 0.26)

[A12] N. G. Shakev, A. V. Topalov, S. A. Ahmed and **V. L. Popov**, (2019) "A Stable Control Algorithm for Multi Robot Formation". 8th International Scientific Conference "TechSys 2019", Technical University of Sofia, Plovdiv Branch 16-18 May 2019, Plovdiv, Bulgaria

Vasil Lyubenov Popov

NEW APPROACHES FOR ADAPTATION AND TRAINING IN INTELLIGENT ROBOT CONTROL

Abstract

The requirements for intelligent robots are increased and they create new challenges for service mobile robots, industrial manipulators and research community with the rapid development of technology and the expansion of the areas of application. The vision of future robotic systems is to provide more flexibility, the robots should be easy to program with intuitive human-machine interface. The complexity of the tasks and problems of sharing a workspace by robots and man/other robot also requires the use of complex research methodologies and techniques in the development of new training and adaptation approaches. The dissertation explores aspects of the artificial intelligence, multisensory data fusion and collaborative behavior in shared environment with other robots or humans. The dissertation aims to increase the potential of the robotized devices. It is focused on the development of algorithms for training and adaptation of intelligent behaviors of autonomous robots, cooperative behavior in multi-robot mobile sensor network and hand gesture interfaces. The goal is through the use of approaches and technologies related to artificial intelligence, sensory information processing, simulation models – to achieve higher functionality of robots in a shared working environment. Methods have been developed to define new capabilities that take advantage of the computer vision, artificial intelligence and gesture recognition.

A method for object detection has been developed using deep convolutional neural networks. The algorithm is able to encircle the detected object. The usage of 2D Lidar data, allows the tracking task to be performed. The approach relies on model-free detection. Additionally, an experiment is carried out to follow a human, using multisensory information provided by 2D Lidar and Kinect 360 sensor.

A method for cooperative multi-robot behavior executes searching and detecting of polluted areas by a team of robots. The robotized nodes inside the mobile wireless network can change their states, depending on the interactions with other mobile robots. A stable tracking algorithm for a formation with a leading robot is developed. Two following mobile robots follow the leading robot, keeping pre-defined displacement. The algorithm is asymptotically stable, the statement is proven using Lyapunov stability theorem. A simulation program is created to visualize the following of the leader robot in a Webots environment.

A method for intuitive to people gesture based interface is proposed. The Kinect 360 sensor ability to track specified points of human body is used. The algorithm uses the displacement along axes of Kinect sensor coordinate system to recognize specific control

gestures. The gestures are digitalized, and control numbers are sent to quadrotor, unmanned aerial vehicle. Gesture based interface is also applied to an industrial SCARA manipulator made by the company Mitsubishi Electric. The experiment explores the applicability of an intuitive gesture interface in manufacturing industrial environment.