



Софийски университет “Св. Климент Охридски”

Факултет по математика и информатика

Катедра „Компютърна информатика“

Моделиране и управление на антропоморфен модел на робот

АВТОРЕФЕРАТ

на дисертационен труд

на Любомира Лъчезарова Митева

за присъждане на образователна и научна степен „Доктор“ в
професионално направление 4.6. Информатика и компютърни науки

Докторска програма:

„Информационни системи“ – Вградени и автономни системи

Научни ръководители:

проф. д-р Евгений Кръстев

доц. д-р Иван Чавдаров

гр. София, 2023 г.

Дисертационният труд е обсъден и насочен за защита на разширено заседание катедра „Компютърна информатика“ към Факултета по математика и информатика на Софийски университет „Св. Климент Охридски“, състояло се на 06.02.2023 г.

Авторът е докторант в редовна форма на обучение във Факултета по математика и информатика на Софийски университет „Св. Климент Охридски“.

Дисертацията е с обем от 132 страници, в които се съдържат 75 Фигури, 11 Таблици, цитирани са 98 източника. Списъкът от публикации на автора по същността на дисертацията включва 6 заглавия.

Публичната защита на дисертационния труд ще се състои на открито заседание на Г. от часа в

Материалите по защитата са на разположение в на ФМИ към СУ (София, бул. Джеймс Баучър 5).

Съдържание

Обща характеристика на дисертационния труд	4
Актуалност на проблема и мотивация.....	4
Цел и задачи на дисертацията	5
Структура и съдържание на дисертационния труд	5
Глава 1. Обзор на предметната област	5
Глава 2. Моделиране на равнинен антропоморфен робот.....	10
Глава 3. Управление на равнинен антропоморфен робот	16
Глава 4. Експериментална верификация, чрез 3D принтиран робот.....	26
Заключение	32
Цитирани източници	33
Приноси в дисертационния труд	35
Научноприложни приноси.....	35
Приложни приноси.....	35
Публикации, доклади и участие в проекти, свързани с темата на дисертацията.....	36
Публикации.....	36
Доклади	36
Участие в проекти, свързани с темата на дисертацията	37
Декларация за оригиналност	37
Биографични данни за автора на дисертационния труд	37
Благодарности.....	37

Обща характеристика на дисертационния труд

Индустриалните работи обикновено изпълняват предварително дефинирани задачи, като едни от основните им дейности са обработка на метали, заваряване, сглобяване, транспортиране, боядисване и др. Всеки робот разполага с различен брой независими стави, които еднозначно определят позицията и ориентацията на изпълнителното му звено. Казваме, че броят на ставите на един робот определя броя на степените на свобода на този робот. От роботите се изисква да изпълнят прецизно и точно поставените им задачи, спазвайки определена траектория и съобразявайки се с налични препятствия в работното им пространство. Поради тази причина някои работи се проектират по такъв начин, че да разполагат с повече степени на свобода, отколкото са необходими за изпълнение на желаната задача. Това ги прави по-гъвкави, по-лесно преодоляват налични препятствия в работната им зона, както и повишава производителността им [1]. В дисертационния труд са разгледани и предложени подходи за изпълнение на движение за оптимално време и планиране на траектория при наличие на статични и динамични препятствия в работната зона на робот с допълнителни степени на свобода.

Актуалност на проблема и мотивация

Мотивацията за въвеждане на допълнителни степени на свобода към механичната структура идва от целта да се увеличи надеждността на роботизираната система и да се намали вероятността за грешка. Наличието на допълнителни степени на свобода позволява движения на манипулатора, които не изместват хващача от определена позиция. Това означава, че една и съща позиция и ориентация на хващача може да бъде изпълнена с различни ставни конфигурации, което дава възможност за избягване на препятствия и води до по-голяма гъвкавост на манипулатора [2]. Тази характеристика е ключова за използване на роботизирани системи в динамична и недетерминирана среда. Най-много работи с допълнителни степени на свобода се използват за индустриални цели, като боядисване, сглобяване, заваряване и др. [3].

Роботите с допълнителни степени на свобода преодоляват значително по-лесно препятствията в работната им зона, докато с изпълнителното си звено изпълняват основната си задача. Въпреки всички предимства, които този тип работи имат пред роботите без допълнителни степени на свобода, има и при тях някои особености. Роботите с допълнителни степени на свобода имат множество от ставни конфигурации, които са решения на обратната задача на кинематиката. Множеството решения позволяват избор на най-подходящата ставна конфигурация, но правят по-трудно решаването на обратната задача на кинематиката и планирането на движение. Как точно ще се избере най-подходящата ставна конфигурация зависи до голяма степен от поставената задача. Решенията на обратната задача на кинематиката могат да се разделят на няколко типа в зависимост от стойностите на ставните ъгли. Работната зона на роботите манипулатори също може да се раздели на отделни зони в зависимост от съществуващите типове решения на обратната задача на кинематиката. При изпълнение на дадено движение, като преместване на хващача на работа от една точка до друга, роботът преминава от една ставна конфигурация в друга или извършва преход между две зони. Но това преминаване може да доведе до отклоняване от желаната траектория или увеличаване на времето за изпълнение. Затова е необходимо да се изследват различните зони в работното пространство на работа и да се предложи подход за намиране на точки, в които роботът може да промени ставната си конфигурация без отклоняване от желаната траектория. Допълнително, при наличие на препятствия, преходът между някои от зоните може да се окаже невъзможен. Ако пък поставената задача на работа е сглобяване и е необходимо в определена точка от работното му пространство да промени ориентацията на изпълнителното си звено, то е възможно при

неправилно планиране на неговото движение, хващачът на робота да се отмести от желаната точка.

Възможните ограничения на роботите с допълнителни степени на свобода при изпълнение на зададено движение при наличие на препятствия мотивираха създаването на дисертацията с цел да се изследват различни методи за планиране на траектория в среда с налични статични и динамични препятствия. Планирането на движение е проблем изследван от много години и от много учени, но в литературата липсват изследвания относно класифицирането на решенията на обратната задача на кинематиката в зависимост от стойностите на ставните ъгли и използването на тази класификация при планиране на движение.

Цел и задачи на дисертацията

Целта на тази дисертация е да се създаде математически модел и прототип на равнинен антропоморфен робот с допълнителни степени на свобода, както и да се изследват и създадат методи за управление на неговото движение. Провеждането на това изследване трябва да доведе до създаване на методи за планиране на траектория при наличие на статични или динамични препятствия в работната зона на робота. За постигане на желаната цел могат да се формулират следните задачи:

1. Класифициране на различни типове решения на обратната задача на кинематиката в зависимост от стойностите на ставните координати на антропоморфния робот с допълнителни степени на свобода.
2. Изследване и създаване на алгоритми за планиране на траектория за изследвания робот с допълнителни степени на свобода с цел преодоляване на статични и/или динамични препятствия и достигане до желана целева позиция.
3. Създаване на прототип на равнинен робот с допълнителни степени на свобода, посредством методите на 3D принтирането.
4. Анализирание и подбор на подходящи хардуерни компоненти и проектиране на подходяща софтуерна система за управление на създадения робот с допълнителни степени на свобода.
5. Верифициране на предложената хардуерна и софтуерна система за управление и на алгоритмите за планиране на траектория, чрез компютърна симулация и експеримент с проектираната роботизирана система.

Структура и съдържание на дисертационния труд

Структурата на дисертационния труд се състои от Увод, 4 глави и Заключение и е с обем от 132 страници. Представените в тези глави резултати са докладвани на 4 международни конференции. Публикациите са реферирани в Scopus.

Глава 1. Обзор на предметната област

Разработването на роботизирани системи става възможно благодарение на развитието на различни научни области като механика, електроника, информатика, математика и др. Сред предимствата от използването на роботи са: намалени разходи за труд, по-добра прецизност и производителност, по-добра гъвкавост в сравнение със специализираните машини и замяна на човека в повтарящи се или животозастрашаващи задачи [4]. В тази глава се разглеждат основните характеристики на роботите с допълнителни степени на свобода и се анализират различни методи за планиране на движение при наличие на препятствия за манипулационни работи.

Антропоморфни работи с допълнителни степени на свобода

Тенденцията е все повече работи да се използват в неструктурирана и динамична среда, където трябва да работят съвместно с други работи или хора. Поради тази причина все повече работи се проектират по такъв начин, че да имат повече степени на свобода отколкото са необходими за изпълнението на поставените им задачи. Разликата на размерността на ставното пространство и това на поставената задача определя степента на излишество (допълнителните степени на свобода) за дадена роботизирана система [5]. Роботите с допълнителни степени на свобода могат да достигнат дадена позиция и ориентация на хващача, използвайки различни конфигурации на механичната си структура [2]. Тези роботизирани системи се характеризират с по-голяма производителност и точност при изпълнение на поставената задача [1, 6]. Наличието на множество ставни конфигурации за дадена позиция и ориентация на крайното звено позволява по-лесно преодоляване на препятствия [7] в сравнение с останалите роботизирани системи. Роботите, които по един или друг начин притежават човешки характеристики се наричат антропоморфни.

Планирането на движение на работи с допълнителни степени на свобода е интересна тема, тъй като е необходим критерий за избор на подходяща ставна конфигурация в зависимост от поставената задача. В литературата са изследвани и предложени различни методи за избор на подходяща ставна конфигурация. Hollerbach предлага избор на такава ставна конфигурация, при която може да се постигне оптимизиране на въртящия момент [8], Hirakawa и Kawamura представят метод за избор на ставни конфигурации, който може да оптимизира консумираната от робота енергия [9], Chembuly и Voruganti изследват метод за преодоляване на препятствия [10] и Йошикава въвежда метриката коефициент на манипулативност за избор на ставна конфигурация [11]. Предложените методи в литературата за избор на подходяща ставна конфигурация са приложими за конкретните цели като оптимизиране на консумираната от робота енергия или прилагане на по-голяма сила. Решенията на обратната задача на такъв тип робот могат да се класифицират на няколко вида спрямо стойностите на ставните ъгли. Допълнително, на базата на различните типове решения, работната зона на робота може да се раздели на различни зони. Когато роботът променя ставната си конфигурация, може да се наложи да премине от една зона в друга. В предложените методи в литературата не се взема предвид факта, че когато роботът трябва да премине от една ставна конфигурация в друга, има вероятност да се отмести от желаната траектория. Това би довело до нарушения в движението, повече време за изпълнение на поставената задача или до некоректно изпълнение на задачата. Затова е необходимо да се изследват и намерят точките (зоните), в които роботът може да осъществи промяната в ставната конфигурация без това да доведе до отклонение от желаното движение.

Методи за планиране на движение

Изпълнителното звено на дадена роботизирана система трябва да изпълни движение от дадена стартова точка до определена целева позиция в работното пространство, заобикаляйки налични статични или динамични препятствия и съобразявайки се със ставните ограничения на робота. Различни методи за планиране на движение са изследвани и представени в литературата. Едни от най-популярните и използвани са Dijkstra, A*, Rapidly-exploring random tree и probabilistic roadmap методите [12].

Методът **Dijkstra** е предложен през 1959 г. от Edsger Dijkstra и се използва за намиране на най-късите пътища между възлите в даден граф [13]. Той се смята за ефективен и ефикасен метод дори при графи с голям брой възли. Алгоритъмът намира най-къс път от даден връх до друг даден връх [14].

Алгоритъмът **A*** се основава на best-first search алгоритъма и използва евристична функция за намиране на най-краткия път чрез оценка на цената, оставаща до целевия връх [14]. Подходящ

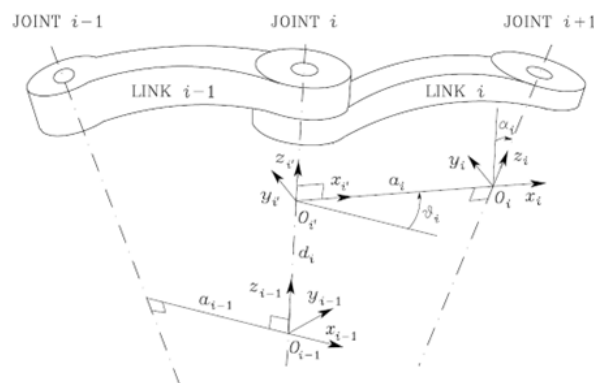
е за планиране на движение с преодоляване на статични препятствия, но ако графът с възможните пътища е твърде голям, алгоритъмът ще загуби от ефективността си [15].

Алгоритмите **Rapidly-exploring random tree (RRT)** и **Probabilistic roadmap method (PRM)** са също едни от най-популярните методи за планиране на движение [12]. RRT методът е често използван в различни комерсиални и индустриални задачи. Методът създава дърво, посредством което се изследва работното пространство на робота, чрез произволно търсене на подходящи ставни конфигурации [16]. Алгоритъмът е подходящ за динамична среда. PRM методът се използва по-често в статична среда с предварително известни препятствия [17].

Моделиране на кинематиката

Кинематиката изследва позицията, скоростта и ускорението на ставните координати. Правата и обратната задача на кинематика се използват за проектиране на всяка една роботизирана система [18].

Правата задача на кинематиката намира позицията и ориентацията на крайното звено на дадена роботизирана система по предварително зададени ставни (обобщени) координати. За нейното решаване е необходимо да се поставят локални координатни системи към всяко звено на роботизираната система. Най-разпространеният метод за избор на координатни системи за манипулатори с отворена кинематична верига е конвенцията на Денавит-Хартенберг (Фиг. 1.6), която дефинира релативната позиция и ориентация на две съседни звена. Необходимо е да се определят координатните системи на две съседни звена и да се намери координатната трансформация между тях. Избира се фиксирана координатна система $O_0x_0y_0z_0$ свързана с центъра на основата на манипулатора. Оста z_0 съвпада с оста на въртене и/или оста на трансляция на първото звено спрямо основата, а осите x_0 и y_0 се избират, така че $O_0x_0y_0z_0$ да е дясноориентирана координатна система. Номерират се звената от основата към последното звено с $i = 1, 2, \dots, N$ и към всяко звено се свързва неподвижно координатна система $O_ix_iy_iz_i$. Конвенцията на Денавит-Хартенберг определя координатната система на i -тото звено. Оста $z_i, i = 1, 2, \dots, N - 1$ се избира по посока на оста на звеното. Оста $x_i, i = 1, 2, \dots, N - 1$ се определя от общия перпендикуляр между осите z_{i-1} и z_i с положителната посока от звено i към звено $i + 1$. Оста $y_i, i = 1, 2, \dots, N - 1$ се избира, така че да допълва дясноориентирана координатна система. Последната координатната система $O_nx_ny_nz_n$ се свързва неподвижно с крайното звено на манипулатора, като оста z_n определя ориентацията му и обикновено се явява негова ос на симетрия. Останалите оси се определят по описания начин [19].



Фиг. 1.6. Параметри на Денавит-Хартенберг [19].

След като са поставени всички локални координатни системи, позицията и ориентацията на координатната система i спрямо координатната система $i - 1$ са напълно определени от следните параметри: a_i е разстоянието между O_i и $O_{i'}$ по оста x_i , d_i е разстоянието между O_{i-1} и $O_{i'}$ по оста z_{i-1} , α_i е ъгълът между осите z_{i-1} и z_i спрямо оста x_i , θ_i е ъгълът между осите x_{i-1} и x_i спрямо оста z_{i-1} . За да се изрази координатната трансформация между координатните системи

$O_{i-1}x_{i-1}y_{i-1}z_{i-1}$ и $O_i x_i y_i z_i$ се изпълняват следните стъпки, състоящи се от постъпателни премествания и завъртания:

- Завъртане на ъгъл θ_i по оста z_{i-1} докато осите x_{i-1} и x_i' станат успоредни.
- Постъпателно преместване по оста z_{i-1} на разстояние d_i до съвпадане на осите x_{i-1} и x_i' .
- Постъпателно преместване по оста x_i' на разстояние a_i до съвпадане на координатното начало O_i' с O_i .
- Завъртане на ъгъл α_i около оста x_i' до съвпадане на всички координатни оси.

Трансформационната матрица между две координатни системи може да се изрази като произведение от четири трансформации:

$$\mathbf{T}_i^{i-1} = \mathbf{R}(Z_{i-1}, \theta_i) \mathbf{Tr}(Z_{i-1}, d_i) \mathbf{Tr}(X_i, a_i) \mathbf{R}(X_i, \alpha_i) \quad (1.1)$$

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

След извършване на необходимите изчисления за трансформационната матрица \mathbf{T}_i^{i-1} получаваме:

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

Решението на правата задача на кинематиката се изразява като произведение на всички трансформационни матрици, които представят координатните трансформации между отделните звена. Конвенцията на Денавит-Хартенберг е приложима и при решаване на правата задача на кинематиката при манипулационни работи с допълнителни степени на свобода.

При планиране на движение по зададена траектория е необходимо роботът да изпълни определени позиции и ориентации с изпълнителното си звено. За тази цел трябва да се намерят стойностите на ставните координати по зададена позиция и ориентация на хващача на манипулатора. Тази задача е известна като **обратната задача** на кинематиката [4]. Конфигурацията на работа се изразява с вектор стълб $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)^T$, който описва положението на всяка една става на роботизираната система. Казваме, че всеки манипулатор има n на брой звена и θ_i наричаме ставен ъгъл. Желаната позиция и ориентация на хващача може да се опише с хомогенна трансформационна матрица \mathbf{H} с размерност 4×4 , съгласно уравнение (1.4):

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix} \quad (1.4)$$

където \mathbf{R} е ротационна матрица с размерност 3×3 , \mathbf{p} е вектор стълб, който описва транслацията, \mathbf{H} е хомогенна трансформационна матрица, която описва позицията и ориентацията на хващача. Задачата се свежда до намирането на ставните променливи $\theta_1, \dots, \theta_n$, такива че $\mathbf{T}_n^0(\theta_1, \dots, \theta_n) = \mathbf{H}$ [4].

За разлика от правата задача на кинематиката, за решението на обратната задача няма общоприет алгоритъм, по който да се реши. Обикновено се казва, че обратната задача е решена за даден манипулатор, ако ставните променливи са определени по алгоритъм, който позволява намирането на всички конфигурации на ставните променливи за дадена позиция и ориентация [20]. При работи с допълнителни степени на свобода е възможно да има безкрайно много

решения на обратната задача на кинематиката. Затова методите за намиране на решения на обратната задача на кинематиката на работи без допълнителни степени на свобода не са приложими за работи с допълнителни степени на свобода.

При работи с допълнителни степени на свобода за решение на обратната задача на кинематиката най-често се използват итеративни методи, базирани на матрицата на Якоби. Те дават приближено решение и на всяка стъпка се опитват да минимизират разликата между текущата позиция и ориентация на хващача и желаната позиция и ориентация. Решенията на Якобиана представляват линейни приближения на обратната задача на кинематиката. Те моделират линейно движенията на крайното звено спрямо моментните промени в ставните ъгли. Матрицата на Якоби е функция на ставните ъгли $\theta = (\theta_1, \theta_2, \dots, \theta_n)$. Orin и Schrader описват в [21] как се пресмята матрицата на Якоби за различните видове стави. Най-лесният метод базиран на матрицата на Якоби е методът на **транспонираната матрица на Якоби**. Методът използва транспонираната матрица на Якоби. Използван е за първи път за решение на обратната задача на кинематиката в [22]. Промяната в ставните скорости $\Delta\theta$ се изразява с уравнение (1.7):

$$\Delta\theta = \alpha J(\theta)^T e \quad (1.7)$$

където α е скаларна константа, която влияе върху промяната на θ на всяка итерация [23] и e е вектор на грешката (6×1), представящ отклонението на позицията и ориентацията между текущите позиция и ориентация s и целевата позиция и ориентация t : $e = t - s$. Стойността на α се избира такава че, новата стойност на вектора e да бъде минимална. Предимствата на този метод са: минималното време за изчисление и лесната имплементация [24], но недостатъкът на метода са лошите му резултати в близост до сингулярни конфигурации [25].

Обратният метод на Якоби използва обратната матрица на Якоби за изчисляване на промяната в ставните координати. Обратната матрица на Якоби съществува само при условие, че матрицата на Якоби има пълен ранг. Размерността на матрицата на Якоби расте с увеличаване на степените на свобода на робота, затова и методът ще изисква повече време за намиране на обратната матрица при работи с допълнителни степени на свобода [26]. В близост до сингулярна конфигурация този метод е неприложим, тъй като рангът на матрицата на Якоби не е пълен [27].

Добро приблизително решение на обратната задача на кинематиката дава методът на **псевдообратната матрица на Якоби** или известна още като обратната матрица на Moore-Penrose. Тя се означава с J^\dagger и е $n \times m$ матрица. За роботизирани системи с допълнителни степени на свобода, псевдообратната матрица на Якоби се изразява със следното уравнение (1.9).

$$J^\dagger = J^T (JJ^T)^{-1} \quad (1.9)$$

Методът на псевдообратната матрица на Якоби е сравнително бърз, но с не много добра апроксимация [46]. Методът дава като резултат много голяма скорост на крайното звено в близост до сингулярна конфигурация или когато точката, в която трябва да се позиционира хващачът на робота е извън зоната му на достижимост.

Основният общ проблем на разгледаните до тук методи е лошата им производителност в близост до сингулярна конфигурация. С този проблем може да се справи методът на Левенберг-Маркуад – **метод на затихващите най-малки квадрати** (Damped Least Square, DLS) с коефициент на затихване (damping factor). Коефициентът на затихване може да намали голямата скорост в близост до сингулярност. Методът е бил използван за решаване на обратната задача на кинематиката за първи път от Wampler [28]. Определян е като най-добрият за намиране на решение на обратната задача на кинематиката на работи с допълнителни степени на свобода в сравнение с останалите методи, базирани се на матрицата на Якоби [24, 29]. Неговият недостатък е че е по-бавен в сравнение с останалите разгледани методи, необходимо е повече изчислително време [24].

Глава 2. Моделиране на равнинен антропоморфен робот

В тази глава се представят функционалните изисквания за изследвания робот и се въвеждат основни означения. Описват се използваните методи за решаване на правата и обратната задача на кинематиката за разглеждания робот. В литературата не се среща класифициране по тип на решенията на обратната задача на кинематиката за роботи с допълнителни степени на свобода. Тази класификация е важна при планирането на движение, тъй като промяната в типа решение може да доведе до отклонение от желаната траектория или некоректно изпълнение на поставената задача. Научноприложните приноси в тази глава са свързани със създаване на подход за класифициране по тип решенията на обратната задача на кинематиката за разглеждания робот, извършване на анализ на работното пространство в зависимост от препятствията в работната зона на равнинен робот с допълнителни степени на свобода и изследване на ъгъла на сервис в различните зони (определени от типовете решения) в работното пространство на робота. Тази глава представя получените резултати от изпълнението на първа задача от задачите на дисертационния труд.

Функционални изисквания

Основната задача на роботите е да изпълняват зададено движение от точка до точка или по предварително зададена траектория. За тази цел е необходимо да се вземат предвид конструктивните ограничения на робота, както и ограниченията породени от средата, в която той манипулира. Обикновено тези ограничения са препятствия в работната зона на робота. Те могат да бъдат статични или динамични. Роботът, моделиран в дисертационния труд е необходимо да може да изпълни поставена задача и в статична и в динамична среда. За тази цел той трябва да отговаря на следните изисквания:

- Роботът трябва да бъде с допълнителни степени на свобода при изпълнение на равнинно движение, което означава, че трябва да има поне три звена и 4 ротационни стави за изпълнение на движения в равнината.
- Дължините на звената и ставните ограничения трябва да са съобразени с параметрите на задвижващите механизми, които ще се използват. Необходимо е да се покрива максимално работно пространство.
- Необходимо е роботът да може да изпълнява пространствени движения.
- Необходимо е моделираният робот да може да изпълнява дадена траектория прецизно, без отклонение от зададения път и за минимално време.
- Роботът трябва да разполага с подходяща хардуерна и софтуерна система с помощта, на която да може да локализира и преодолява наличните препятствия в работната си зона.

Описание на антропоморфен робот

Нека с m да означим броя на независимите параметри, описващи изпълнението на дадено задание в работното пространство на робота, а с n броя на степените на свобода на робота. За да бъде роботът с допълнителни степени на свобода при изпълнение на движение в равнината O_{xyz} , n трябва да бъде по-голямо от m . За изпълнение на произволно равнинно движение са необходими 3 степени на свобода, т.е. $m = 3$. Затова е необходимо роботът да има повече от 3 степени на свобода.

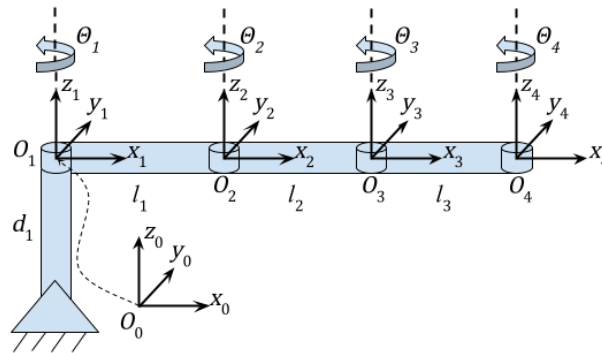
Нека с $l_i, i = 1, \dots, 4$ да означим дължините на звената на робота. Моделираният робот ще има 4 ротационни стави (Фиг. 2.1). Нека с $\theta_i, i = 1, \dots, 4$ да означим ставните координати на робота. Крайното звено на робота ще наричаме изпълнително звено или хващач. Позицията на хващача в равнината O_{xyz} ще означаваме с (x, y, z) . Роботът ще изпълнява задания, при които оста z се

приема за константа ($z = const$) и един от векторите на ориентация остава успореден на O_z . Ротационните стави ще имат ставни ограничения:

$$\theta_i^{min} \leq \theta_i \leq \theta_i^{max}, \text{ за } i = 1, \dots, 4 \quad (2.1)$$

Права задача на кинематиката

При роботизирани системи с отворена кинематична верига ставните координати определят еднозначно позицията и ориентацията на крайното звено [30]. Изследваният робот с допълнителни степени на свобода има 4 ротационни стави и съответно ставни координати - $\theta_1, \theta_2, \theta_3$ и θ_4 . Когато четвъртото звено е с нулева дължина, то четвъртата ставна координата θ_4 променя само ориентацията на крайното звено и не оказва влияние върху неговата позиция. В такъв случай, ставните координати $\theta_1, \theta_2, \theta_3$ определят позицията на хващача в O_{xy} равнината. Интересуваме се само от позицията на хващача, тъй като ориентацията на хващача се получава от сумата на четирите ставни координати. Тя се променя, чрез задаване на ставната координата θ_4 . На Фиг. 2.1 е представена кинематичната схема на моделирания равнинен робот с допълнителни степени на свобода. Нека да означим позицията на хващача в работното пространство с (x, y) и ставните координати в обобщеното пространство с $\theta = (\theta_1, \theta_2, \theta_3)$.



Фиг. 2.1. Кинематична схема на равнинен робот с допълнителни степени на свобода.

За решаването на правата задача на кинематиката ще използваме конвенцията на Денавит-Хартенберг. Координатна система $O_i x_i y_i z_i$ е поставена на всяка става i , $i = 0, 1, 2, \dots, n$, като координатната система $O_0 x_0 y_0 z_0$ е поставена към основата на робота. Оста z_i съвпада с оста на въртене на става $i + 1$ спрямо става i . За удобство оста z_0 е избрана да съвпада с оста z_1 и $O_0 \equiv O_1$. Оста z_n е избрана, така че да представя оста на ротация на хващача. Оста x_i е перпендикулярна на двете оси z_{i-1} и z_i . Оста y_i допълва $O_i x_i y_i z_i$ към дясно ориентирана координатна система. След като към всяка става е поставена неподвижна локална координатна система $O_i x_i y_i z_i$, може да се определи позицията и ориентацията на координатна система $O_i x_i y_i z_i$ спрямо координатна система $O_{i-1} x_{i-1} y_{i-1} z_{i-1}$. За тази цел се използват следните параметри. Разстоянието между O_{i-1} и O_i по оста x_i се означава с a_{i-1} , разстоянието между две съседни координатни системи по оста z_i се представя с d_i , α_{i-1} е ъгълът между осите z_{i-1} и z_i спрямо оста x_i . Обобщените координати θ_i представят ротация спрямо оста z_i , като ставата i е съответно ротационна става [31]. Стойностите на параметрите на Денавит-Хартенберг за робот с допълнителни степени на свобода са представени в Таблица 2.1.

Таблица 2.1. Параметри на Денавит-Хартенберг.

i	α_{i-1} [rad]	a_{i-1} [mm]	d_i [mm]
1	0	0	d_1
2	0	l_1	0
3	0	l_2	0
4	0	l_3	0

Конвенцията на Денавит-Хартенберг ни позволява да изразим всяка трансформационна матрица между две координатни системи, като се извършат следните постъпателни премествания и завъртания. Завъртане около оста z_{i-1} на ъгъл θ_i , докато осите x_{i-1} и x_i станат успоредни. Постъпателно преместване по оста z_{i-1} на разстояние d_i до съвпадане на осите x_{i-1} и x_i . Постъпателно преместване по оста x_i на разстояние a_i до съвпадане на координатното начало O_{i-1} с O_i . Завъртане на ъгъл α_i около оста x_i до съвпадане на всички координатни оси. Тези трансформации се представят като произведение на четири матрици (1.1). След умножение на дадените матрици за трансформационната матрица T_i^{i-1} получаваме матрицата от (1.3).

За да решим правата задача на кинематиката е необходимо да намерим трансформационна матрица T_4^0 , която се изразява като произведение на всички трансформационни матрици:

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 \quad (2.2)$$

След изразяване и заместване в (1.3) и (2.2) за трансформационната матрица T_4^0 получаваме следния резултат (2.3):

$$T_4^0 = \begin{bmatrix} c_{1234} & -s_{1234} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ s_{1234} & c_{1234} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Където за $1 \leq i, j, k, l \leq 4$:

- $s_i = \sin(\theta_i), c_i = \cos(\theta_i),$
- $s_{ij} = \sin(\theta_i + \theta_j), c_{ij} = \cos(\theta_i + \theta_j),$
- $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k), c_{ijk} = \cos(\theta_i + \theta_j + \theta_k),$
- $s_{ijkl} = \sin(\theta_i + \theta_j + \theta_k + \theta_l), c_{ijkl} = \cos(\theta_i + \theta_j + \theta_k + \theta_l).$

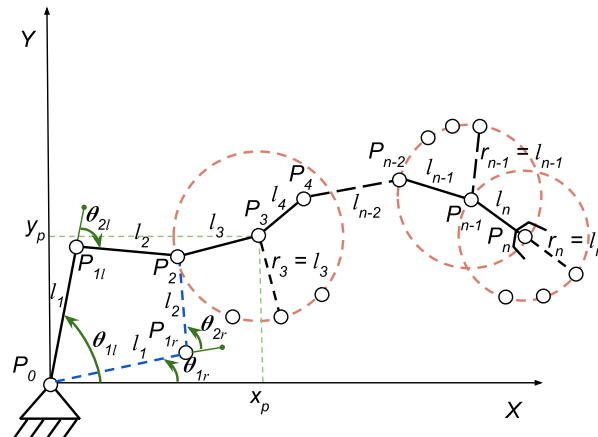
Така за x, y и z координатите, определящи позицията на крайното звено на робота, получаваме (2.4):

$$x = a_1 c_1 + a_2 c_{12} + a_3 c_{123}, y = a_1 s_1 + a_2 s_{12} + a_3 s_{123}, z = d_1 \quad (2.4)$$

Обратна задача на кинематиката

Целта на обратната задача на кинематиката е да се намерят необходимите ставни координати при дадена позиция и ориентация на хващача на робота. Решението на обратната задача на кинематиката при работи с отворена кинематична структура най-често се използва за определяне на множество от точки, с които може да се достигне до определена точка в конфигурационното пространство при планиране на движение. Тъй като роботите с допълнителни степени на свобода могат да достигнат дадена позиция в работната им зона с множество ставни конфигурации, то и решенията на обратната задача на кинематиката са безкрайно много. В дисертационния труд е използван геометричен подход за решаване на обратната задача на кинематиката [20]. На Фиг. 2.3 е показан равнинен манипулатор с n звена, роботът е с допълнителни степени на свобода. С l_1, \dots, l_n , където $i = 1, \dots, n$, са означени дължините на звената. Хващачът на робота трябва да се позиционира в точка P_n . За тази цел е необходимо да се намерят стойностите на ставните координати, с помощта на които хващачът достига целевата точка. Подходът за намиране на решение на обратната задача на кинематиката започва от изпълнителното звено (от последната ставна координата) и търси възможно решение за всяка една ставна координата докато се стигне до първата ставна координата. Генерира се множество S_n от точки равномерно разпределени върху окръжност с център P_n и радиус $r_n = l_n$. За всяка точка P_{n-1} от множеството S_n се проверява дали съществува конфигурация, с която да позиционира края на звено $n - 1$ в точка P_{n-1} . За тази цел по аналогичен начин се генерира множество S_{n-1} от точки равномерно

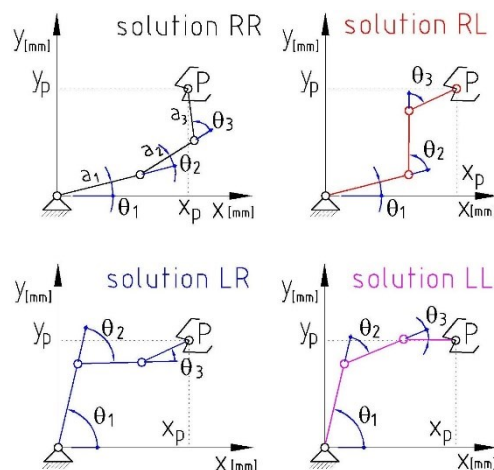
разпределени върху окръжност с център P_{n-1} и радиус $r_{n-1} = l_{n-1}$. Тези действия се повтарят за всяка една ставна координата от крайното звено към основата на робота. Когато се достигне до множество S_3 от точки равномерно разпределени върху окръжност с център P_3 и радиус $r_3 = l_3$, се търси решение на обратната задача на кинематиката на двузвенеен механизъм. Необходимо е за всяка точка P_2 от множеството S_3 , да се провери дали съществува валидна конфигурация за двузвенеия механизъм, състоящ се от звена l_1 и l_2 . Ако съществува поне едно решение, се проверява дали намерените ставни координати отговарят на ставните ограничения. Ставните ъгли, които удовлетворяват ставните ограничения определят ставните конфигурации, които могат да бъдат изпълнени от робота. Тогава определените от точки P_0, P_{1r}, \dots, P_n и P_0, P_{1l}, \dots, P_n конфигурации се запазват като решение на обратната задача.



Фиг. 2.3. Решение на обратната задача на кинематиката на равнинен робот с n звена.

Класификация на различните типове решения на обратната задача на кинематиката

Нека да разгледаме равнинен манипулатор с три звена с ненулеви дължини (Фиг. 2.5.).



Фиг. 2.5. Четири типа решения на обратната задача на кинематиката.

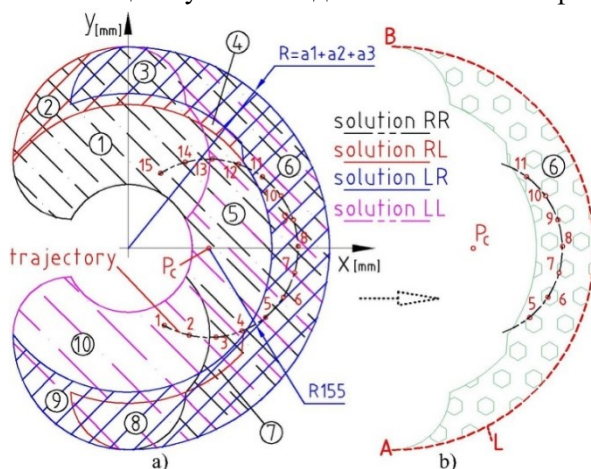
В зависимост от знака на втората и третата ставна координата могат да се дефинират четири различни типа решения на обратната задача на кинематиката: Решение RR, когато $\theta_2 > 0$ и $\theta_3 > 0$. Решение RL, когато $\theta_2 > 0$ и $\theta_3 < 0$. Решение LR, когато $\theta_2 < 0$ и $\theta_3 > 0$. Решение LL, когато $\theta_2 < 0$ и $\theta_3 < 0$. Различните типове решения са показани на Фиг. 2.5. Четирите типа решения не са възможни за всяка точка от работното пространство. Преминването от един тип решение към друг е от важно значение при планиране на движение на робота.

Нека с $l_i, i = 1, \dots, 4$ да означим дължините на звената на робота и да приемем, че изследваният робот има следните дължини на звената (2.10) и ставни ограничения (2.11):

$$l_1 = 150 \text{ mm}, l_2 = 100 \text{ mm}, l_3 = 100 \text{ mm}, l_4 = 0 \text{ mm}, \quad (2.10)$$

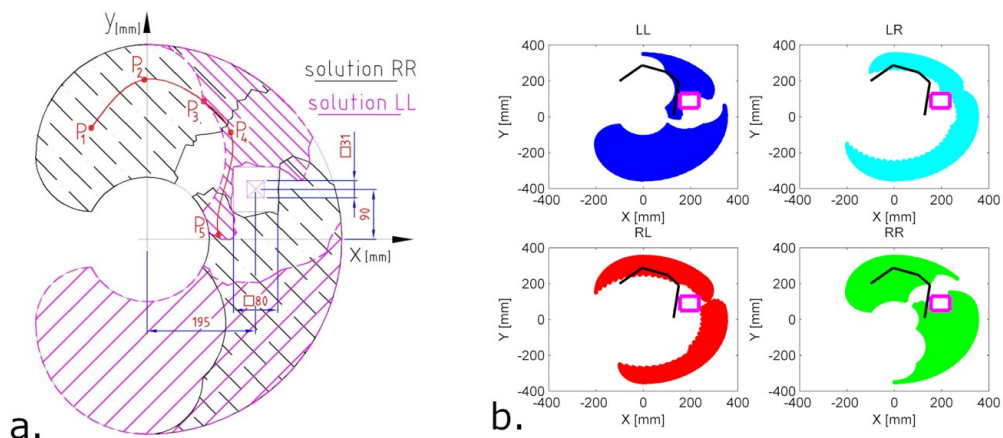
$$-\frac{\pi}{2} \leq \theta_i \leq \frac{\pi}{2}, \text{ за } i = 1, \dots, 4. \quad (2.11)$$

Работното пространство на робота може да бъде разделено на различни зони, в зависимост от различните видове решения на обратната задача на кинематиката. За разглеждания робот са налични десет различни зони (Фиг. 2.6). Обединението на всички зони представлява цялото работно пространство на разглеждания манипулатор (Фиг. 2.6.a.). Сечението на всички зони представлява област от работното пространство, в която се намират и четирите различни вида ставни конфигурации (Фиг. 2.6.b). В работното пространство на робота съществуват траектории, за чието изпълнение се изисква смяна на вида решение на обратната задача на кинематиката. На Фиг. 2.6 е показана такава траектория. Роботът не може да извърши движение от точка 1 до точка 15 без да премине от един вид ставна конфигурация в друг. Това е така, тъй като точка 1 се намира в зона 10, а точка 15 в зона 1 и зоните 1 и 10 не се пресичат. Разбира се, ако желаното движение е изцяло в зона 1 или зона 10, то движението ще бъде изпълнено без смяна на типа решение, тъй като в тези зони съществува само един тип ставна конфигурация.



Фиг. 2.6. Различните зони в работното пространство на робота а) обединението на всички зони б) сечението на всички зони.

Освен ставните ограничения, които определят работното пространство на манипулатора, в работното пространство е възможно да има препятствия (Фиг. 2.8.a).



Фиг. 2.8. Работно пространство на робота при наличие на препятствие: а. работно пространство на робота; б. зоните на четирите типа решение и траектория на желаното движение.

В зависимост от разположението и геометрията им, работната зона на манипулатора се променя. Нека да предположим, че имаме препятствие с квадратна форма и координати (195, 90). Дължина на страната 31 mm и допълнителни 49 mm за ширината на ставите, за да се избегне сблъсък с препятствието (Фиг. 2.8.a). Както се вижда от фигурата, зоните съдържащи тип

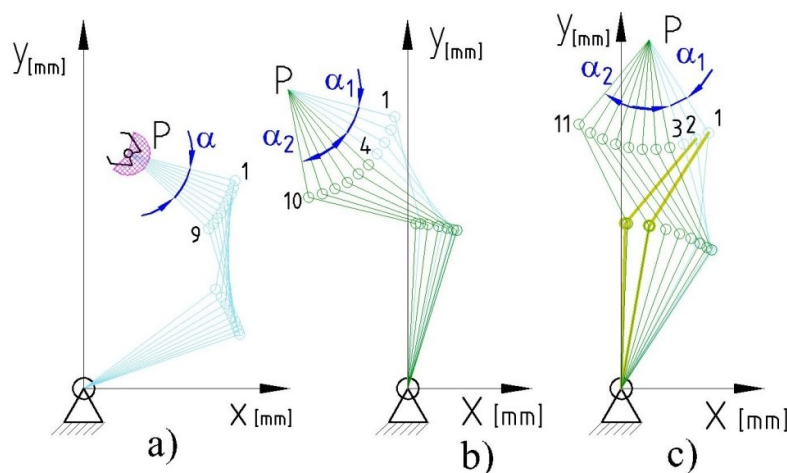
решение RR и LL са разделени на две отделни части. Препятствието променя зоните на четирите типа решения на обратната задача на кинематиката (Фиг. 2.8.b). Ако роботът трябва да изпълни движение с начална позиция от зона номер 1 и крайна точка в зона номер 5 (Фиг. 2.6), то това движение не би било възможно без смяна на типа решение. В следствие на ставните ограничения или наличието на препятствия в работното пространство, роботът трябва да направи смяна на типа решение на обратната задача на кинематиката, за да успее да реализира желаното движение.

Ъгъл на сервис

Терминът ъгъл на сервис е въведен от Виноградов през 1971 г. Ъгълът на сервис е мярка на множеството ориентации на хващача, които манипулаторът може да реализира, когато крайното звено е позиционирано в дадена точка от работното пространство [32].

Четирите типа решения разделят работното пространство на робота на десет зони (Фиг. 2.6). В някои от зоните, като първа и десета съществува само един тип решение на обратната задача на кинематиката. Затова възможните ориентации на четвъртата става могат да бъдат изпълнени само от един тип решение. В останалите зони съществуват точки, които роботът може да достигне с изпълнителното си звено с една и съща ориентация, но с различен тип решение на обратната задача на кинематиката. В тези точки, ако хващачът на робота трябва да промени ориентацията си, то е възможно да се отмести от точката, в която е позиционирано, при преминаване от един тип решение в друг. Поради тази причина е необходимо да разгледаме ъгъла на сервис в различните зони. Заради симетричността на работното пространство на робота е достатъчно да се изследва ъгълът на сервис за първите шест зони.

На Фиг. 2.10 са показани всичките възможни ориентации за позициониране на хващача в точка P от зона 1. Третото звено може да се завърти на ъгъл α от позиция 1 до позиция 9. Четвъртото звено има дължина равна на нула и има ставни ограничения дадени в (2.11). Фиг. 2.10.a представя ъгъла на сервис на четвъртата става. Точка P може да бъде достигната само с решения от тип RR.



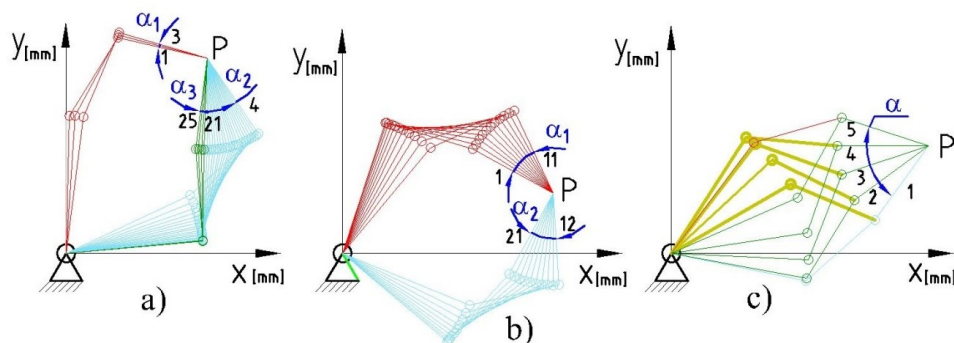
Фиг. 2.10. Различни ставни конфигурации, с които хващачът на робота може да достигне до точка P в: a. зона 1; b. зона 2; c. зона 3.

На Фиг. 2.10.b. са изобразени възможните конфигурации за достигане на точка P в зона 2. В тази зона съществуват два типа решения на обратната задача на кинематиката RR и RL. Преходът между двата типа решения настъпва около конфигурация 4. Ъгълът на сервис може да се представи като сума на ъглите α_1 и α_2 . В тази зона хващачът на робота може да промени своята ориентация без да е необходимо отместване от точка P .

Ориентация в зона 3 е показана на Фиг. 2.10.с. В тази зона съществуват три типа решения: RR, RL и LR. Хващачът на робота може да промени своята ориентация без да е необходимо да се отмести от точка P .

Ако точка P е в зона 4 са възможни 3 типа решения на обратната задача на кинематиката: RR, LL и RL. Както се вижда от Фиг.2.11.а., различните ъгли на хващача са групирани в две зони. Тук, за да се премине от една зона в друга е необходимо да се напусне точка P . Това е важно условие, което трябва да се има предвид при планиране на движения. В зона 5 (Фиг.2.11.б.) ситуацията е аналогична на зона 4. Тук съществуват два типа решения: LL и RR.

В зона 6 съществуват всички типове решения (Фиг.2.11.с). В зависимост от позицията на точка P в зона 6 е възможно, всички възможни конфигурации да формират непрекъснат интервал. По този начин би била възможна промяна в ориентацията на хващача без отместване от точка P . Също така, в тази зона има точки, за които решенията на обратната задача на кинематиката не образуват непрекъснат интервал. В този случай, хващачът на робота трябва да се отмести от точка P , за да премине от един тип решение в друг.



Фиг. 2.11. Различни ставни конфигурации, с които хващача на робота може да достигне до точка P в: а. зона 4; б. зона 5; с. зона 6.

Ако хващачът на робота е позициониран в определена позиция и трябва да промени само своята ориентация, то трябва да се вземат предвид направените разсъждения и да се избере такава ставна конфигурация, при която се променя ориентацията и се минимизира отместването от целевата точка.

Глава 3. Управление на равнинен антропоморфен робот

Основната цел на управлението е да се намерят и зададат подходящи команди (позиции) към задвижващите устройства. Задачата на задвижващите устройства е да позиционират ставните координати по такъв начин, че хващача на робота да достигне желаната позиция и ориентация. За тази цел е необходима хардуерна и софтуерна система, която да извършва необходимите изчисления и да задвижва робота, както и подходящ алгоритъм за планиране на движение. Предложените методи в литературата за планиране на движение не взимат предвид подходящи преходни точки, в които да се извърши промяната в типа решение на обратната задача на кинематиката. В предложените алгоритми в тази глава за планиране на движение са взети предвид подходящите преходни точки, за да може роботът да изпълни поставената му задача за минимално време и без отклонение от зададеното движение. В тази глава са постигнати следните научноприложни приноси: създаден е алгоритъм за планиране на траектория посредством теорията на графите за робот с допълнителни степени на свобода и ограничено ставно пространство, създадени са и подходи за планиране на движение при наличие на статични и динамични препятствия за робот с допълнителни степени на свобода. Приложният принос в тази глава е проектиране на хардуерна и софтуерна система за управление на равнинен робот с

допълнителни степени на свобода. В тази глава са представени резултатите от изпълнението на задачи 2 и 3 от задачите на дисертационния труд.

Хардуерни изисквания

Необходимо е роботът да може да изпълнява предварително зададено движение или последователност от движения. За тази цел разглежданият робот се нуждае от хардуерна система, която да управлява задвижването на робота. Изискванията към хардуерните компоненти са следните:

- Управляващият контролер трябва да разполага с необходимите изчислителни ресурси за пресмятане на правата и обратната задача на кинематиката.
- Управляващата електроника е необходимо да разполага с достатъчно памет, за да позволява зареждане на последователност от позиции, които роботът трябва да изпълни с изпълнителното си звено.
- Управляващата електроника е необходимо да е базирана на платформи с отворен код и удобна среда за разработка.
- Необходимо е роботът да позволява управление както през USB порт, така и по Wi-Fi връзка.
- Управляващият контролер трябва да разполага с поне 3 серийни UART порта. Един за комуникация със задвижващите устройства, един за модула, който ще осигурява безжичната комуникация с робота и един за UART комуникацията с компютъра на оператора.
- Управляващата електроника трябва да разполага със светлинни индикатори за обозначаване на състоянието на моторите (включени/изключени), дали управляващият контролер има захранване и дали е осъществена безжична връзка с робота.
- Задвижващите устройства трябва да позволяват 180 градусово движение на ротационните стави и управление по позиция.
- Тъй като антропоморфният робот с допълнителни степени на свобода ще се използва за научноизследователски и учебни цели, проектираната хардуерна система трябва да може да се разширява и подобрява.

Хардуерни компоненти

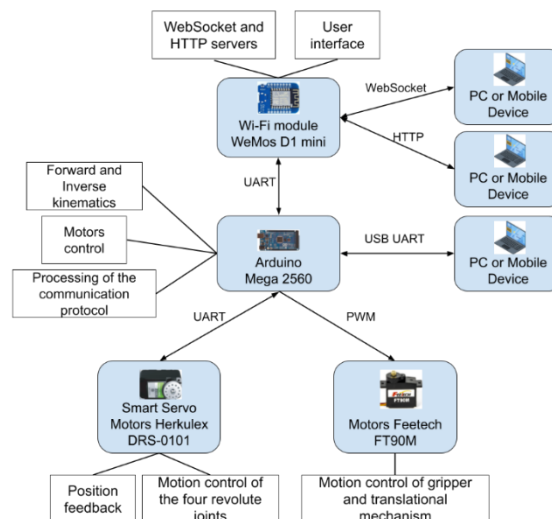
Избраната електроника за управление на робота се базира на платформата Arduino и Wi-Fi модула WeMos D1 ESP8266. Необходимо е ротационните стави на робота да се управляват по позиция, затова за тяхното управление са избрани умни сервомотори HerkuleX DRS-0101. За управлението на хващача и транслагационния механизъм се използват сервомоторите FeeTech FT90M.

Умните сервомотори: **HerkuleX DRS-0101** се управляват, чрез серийна комуникация и могат да върнат информация за текущата им позиция и скорост. Позволяват плавен контрол на позицията им. Поддържат UART комуникация, която позволява промяна на позицията, цвета на LED индикатора, управление едновременно до 254 сервомотора. Поддържат и до седем различни типа грешки. За наличието на някоя от тези грешки се съобщава, чрез LED индикатора [33]. Транслагационният механизъм и хващача на робота се задвижват от мини сервомотори **FeeTech FT90M**. Те нямат обратна връзка, както и контрол на скоростта и се управляват от широчинно-импулсни сигнали (ШИМ, PWM) [34]. За управлението на робота са избрани микроконтролера Arduino Mega 2560 и Wi-Fi модула WeMos D1 mini. **Arduino Mega 2560** е микроконтролер, базиран на микроконтролера ATmega2560. Разполага с 4 хардуерни серийни UART порта, което го прави подходящ за управление на разглеждания робот [35]. Микроконтролерът има всички необходими изчислителни ресурси за пресмятане на правата и обратната задача на кинематиката,

които са необходими за изпълнение на дадено движение и правилно позициониране на хващача. Wi-Fi модулът **WeMos D1 mini** позволява на робота да се подават команди през WebSocket, както и предоставя удобен за програмиране и демонстрация уеб базиран графичен потребителски интерфейс. WeMos D1 Mini е малко Wi-Fi устройство, базирано на ESP8266EX чип [36]. Може да се препрограмира през Wi-Fi връзка. Управляващата електроника разполага с три основни светлинни индикатора.

Хардуерна архитектура и комуникация

Начинът на комуникация на описаните хардуерни компоненти е показан на Фиг. 3.4.



Фиг. 3.4. Хардуерна архитектура.

Три от хардуерните UART порта на микроконтролера Arduino се използват за комуникация със сервомоторите DRS-0101, с Wi-Fi модула и с компютъра на оператора. UART интерфейсите позволяват двупосочна комуникация. Умните сервомотори са свързани на същия комуникационен канал. Те отговарят за задвижването на четирите ротационни стави и връщат информация за текущата позиция. Малките сервомотори се управляват, чрез генерираните ШИМ сигнали от микроконтролера Arduino.

Микроконтролерът Arduino Mega отговаря за пресмятането на правата и обратната задача на кинематиката, за управлението на моторите и за обработката на комуникационния протокол. Wi-Fi модулът осигурява както WebSocket, така и HTTP сървър. WebSocket действа като прокси и препраща заявки и отговори от и към платката на Arduino. HTTP сървърът обслужва разработения графичен уеб-базиран интерфейс. Този интерфейс използва връзката WebSocket за управление на робота. Също така, роботът може да бъде управляван директно, чрез USB свързване между компютър и микроконтролера Arduino.

Софтуерни изисквания

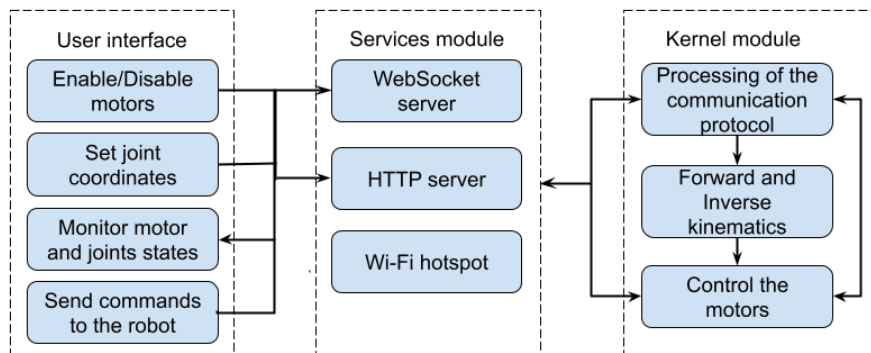
Основната цел на робота е да може да изпълни предварително зададено движение или множество от движения. За тази цел трябва да се разработи софтуер, който отговаря на следните изисквания:

- Софтуерът трябва да може да решава правата и обратната задача на кинематиката коректно и бързо.
- Позволява на потребителя да управлява робота, да включва или изключва моторите, да задава желана позиция в декартови или обобщени координати, да задава последователност от точки за определено движение.
- Софтуерната система трябва да предоставя комуникационен протокол за комуникация с робота и възможност за следене на върнатите отговори от робота.

- Софтуерната система трябва да предоставя и потребителски интерфейс, който може да бъде както графичен, така и текстови.
- Проектираната софтуерна система трябва да позволява добавяне на нови компоненти и функционалности. Това би позволило разработената роботизирана система да се използва за различни образователни, изследователски и индустриални задачи.

Софтуерна система за управление

Разработена е софтуерна система, която отговаря на описаните изисквания. Софтуерната система се състои от три основни модула: модул ядро (kernel module), обслужващ модул (service module) и графичен потребителски интерфейс. Първият модул се изпълнява върху микроконтролера Arduino, а останалите два върху Wi-Fi модула WeMos D1 mini. На Фиг. 3.6 е представена архитектурата на проектираната софтуерна система.



Фиг. 3.6. Архитектура на софтуерната система за управление на равнинен робот.

Модулът на ядрото отговаря за обработката на подадените команди, чрез комуникационния протокол, както и решаването на правата и обратната задача на кинематиката. Модулът отговаря и за подаването на необходимите команди към моторите. Софтуерът за модула на ядрото е реализиран, посредством програмния език C++ и средата за програмиране Arduino IDE. Командите към моторите на робота могат да се изпратят, чрез USB свързване на компютъра и микроконтролерът или чрез WebSocket комуникация, за която отговаря service модула.

Обслужващият (Service) модул се изпълнява независимо от модула на ядрото и предоставя възможност за WebSocket и HTTP комуникация, както и Wi-Fi точка за достъп на робота. Този модул отговаря за осъществяване на връзката между потребителския интерфейс и модулът на ядрото. Wi-Fi модулът поддържа HTTP сървър, който позволява разработването на графичен потребителски интерфейс. **Потребителският интерфейс** може да се достъпва от всеки уеб браузър, след като е осъществена връзка с робота посредством Wi-Fi точка за достъп. За реализацията на потребителския интерфейс са използвани само JavaScript и HTML/CSS.

Приложение на теорията на графите за планиране на траектория

Обикновено задачата на индустриалните манипулатори е да изпълнят предварително дефинирано движение без отклонение от желаната траектория и за минимално време. Затова и времето за изпълнение на дадено движение или зададена траектория е важна част от управлението на роботизираните системи. Колкото по-бързо и точно роботът може да изпълни зададената му траектория, толкова по-голяма ще бъде и неговата продуктивност.

Алгоритъм тип „Алчен“ (Greedy approach)

В зависимост от стойностите на ставните координати, могат да бъдат класифицирани четири типа решения на обратната задача на кинематиката. При изпълнение на зададена траектория е възможно да се наложи роботът да промени типа решение. За да бъде изпълнено дадено движение, то може да бъде планирано по следния начин. Движението започва от произволно

решение на обратната задача на кинематиката. Типът на това решение се запазва и се използва за следващите точки от желаната траектория, докато не се наложи смяна на вида решение на обратната задача на кинематиката. Ако се налага смяна на типа решение, то движението трябва да спре и да се смени типа на ставната конфигурация на робота. След което, изпълнението на зададеното движение може да продължи. Този метод се нарича алчен алгоритъм (greedy approach). Необходимо е да се отбележи, че при прилагане на този метод за робот с допълнителни степени на свобода и ограничено ставно пространство, при смяна на типа решение на обратната задача на кинематиката, роботът ще се отклони от желаната траектория.

Планиране на траектория с минимално отклонение от желания път

От съществено значение е една роботизирана система да може да изпълни поставената задача максимално прецизно и за минимално време. Тъй като повечето роботи изпълняват предварително определена траектория, то това означава, че по време на своето движение не трябва да се отклоняват от тази траектория. Алгоритъмът трябва да определи видовете решения на обратната задача на кинематиката и възможните преходни конфигурации. Желаната траектория трябва да се раздели на определен брой точки, за да може да се използва дискретна структура като насочен граф за планиране на траектория. Подходът трябва да намери възможните решения на обратната задача на кинематиката за всяка точка и да ги класифицира според вида им, след което да построи граф с тегла и да намери път с минимална цена за желаната траектория. Така описан алгоритъмът за планиране на траектория може да се представи в няколко фази:

1. Кинематичен анализ на равнинен робот с допълнителни степени на свобода

Първата фаза от алгоритъма се състои в извършване на кинематичен анализ на равнинен робот, взимайки се предвид ставните му ограничения. Кинематичният анализ ще даде информация относно видовете решения на обратната задача на кинематиката.

2. Анализ на желаната траектория

Втората фаза се състои в извършване на анализ на желаната траектория, за която трябва да се планира движението на робота. Тази траектория трябва да бъде разделена на множество от точки, намиращи се на равно разстояние една от друга.

3. Генериране на решения

В третата фаза е необходимо да се изчислят различните решения на обратната задача на кинематиката за всяка от избраните точки от траекторията от предходната стъпка. Тъй като разглежданият робот е с допълнителни степени на свобода, то съществува множество от решения на обратната задача на кинематиката за всяка една точка. Затова решенията трябва да бъдат ограничени и да бъдат на равни отстояния едно от друго.

4. Класификация на генерираните решения

В тази фаза всички генерирани решения трябва да бъдат класифицирани според намерените типове решения на обратната задача на кинематиката в кинематичния анализ от първата фаза.

5. Конструиране на насочен граф с тегла

Следващата фаза се състои в конструирането на насочен граф с тегла. Всеки един от върховете на графа съответства на генерирано решение от стъпка 3. Между два върха v и u ще има ребро (v, u) , ако v и u съответстват на две съседни точки p_v и p_u от желаната траектория и p_v е преди p_u . Теглото $\omega(v, u)$ за всяко ребро (v, u) може да се дефинира по следния начин:

$$\omega(v, u) = \sum_{i=1}^k |\theta_i^v - \theta_i^u| \quad (3.1)$$

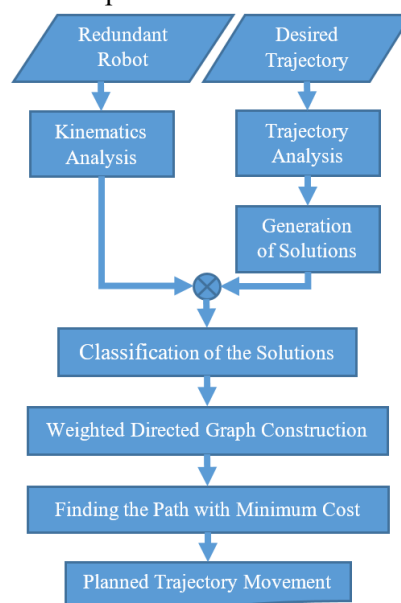
Където k е броят на ставните координати и $(\theta_1^v, \theta_2^v, \dots, \theta_k^v)$ и $(\theta_1^u, \theta_2^u, \dots, \theta_k^u)$ са решенията, съответстващи на върховете v и u . За разглеждания робот $k = 3$. Уравнение (3.1) не взема

предвид времето, необходимо за ускорение или спиране на движението. Тези времена могат да бъдат игнорирани, тъй като се счита, че планираната траектория има ускорение само в началото и намаляване на скоростта в края на движението. Уравнението (3.1) отчита разликата в координатите на ставите на всички k върха. Необходимо е да се отбележи, че преминаването през сингулярна конфигурация може да изисква по-дълго време за движение.

6. Намиране на път с минимална цена за желаната траектория

Последната фаза от алгоритъма е да се намери оптимален път от всеки връх, който представлява решение на стартовата позиция до всеки връх, който представлява решение на целева позиция от желаната траектория. Два от най-използваните метода в теорията на графите са Floyd-Warshall и Dijkstra. Floyd-Warshall алгоритъмът има висока изчислителна сложност и е подходящ за графи с малък брой върхове. Друг популярен алгоритъм е евристичният метод A^* . При прилагане на алгоритмите Dijkstra и A^* за обхождане и търсене в граф има само един начален и краен връх. В разглеждания случай за планиране на траектория за изследвания робот с допълнителни степени на свобода има повече от един начален и целеви връх, в зависимост от броя генерирани решения. Всички намерени начални върхове трябва да бъдат разгледани. Затова за планиране на траектория за равнинен робот с допълнителни степени на свобода, базирано на теория на графите, ще се използва алгоритъмът Floyd-Warshall. На Фиг. 3.14 е представена схема, включваща отделните стъпки от алгоритъма за планиране на оптимална по време траектория.

Крайният резултат на алгоритъма ще бъде път от начален до целеви връх с минимално тегло. Това ще бъдат решенията на обратната задача на кинематиката за последователни точки от желаната траектория. Между всеки две последователни точки представящи различен тип решение, трябва да се планира спиране на движението с цел преминаване през преходна конфигурация и промяна на текущия тип решение. По този начин движението по траекторията може да бъде планирано с минимално време за изпълнение.



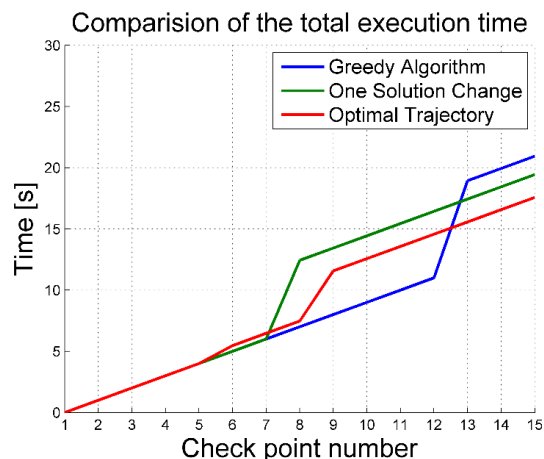
Фиг. 3.14. Схематично представяне на алгоритъма за планиране на траектория.

Сравнение на алгоритмите за планиране на траектория

За да сравним представените алгоритми за планиране на траектория ще използваме траекторията от Фиг. 2.6.а. и разглеждания в дисертацията робот с допълнителни степени на свобода. Отново ще припомним, че роботът има дължини на звената, дефинирани в (2.10) и ставите му имат ограничения, които са дефинирани в (2.11).

Траекторията, която роботът трябва да изпълни представлява движение по дъга. Дъгата е част от окръжност, чийто център е с координати (140, 0) и радиус 155 mm. Желаната траектория не може да бъде изпълнена от конкретния робот, използвайки само един тип решение на обратната задача на кинематиката. Затова ще бъде необходима поне една смяна на типа решение. Желаната траектория е разделена на 15 точки, намиращи се на равно разстояние една от друга. За всяка точка се генерират до 100 решения на обратната задача на кинематиката. Тези решения се категоризират според 4-те възможни типа решения на обратната задача на кинематиката. Тъй като роботът има ограничено ставно пространство, съществуват точки с малък брой решения за избраната траектория. Общият брой намерени решения е 408.

Ако планираме движението с алчния алгоритъм (greedy approach) може да се изпълни с една единствена смяна на типа решение (в точка 12). Но ограниченото ставно пространство ще изисква отклонение от желаната траектория при изпълнение на тази смяна. Движението може да бъде планирано без отклонение от желаната траектория, ако смяната на типа решение се извърши в по-ранна позиция. При изпълнение на предложения алгоритъм за планиране на траектория с минимално време за изпълнение общият брой на намерените решения на обратната задача на кинематиката е 408. Това означава, че конструираният граф има 408 върха. Затова Floyd-Warshall алгоритъмът може да бъде директно приложен, тъй като броят на върховете на графа е по-малък от 1000. Планираният оптимален път се състои от траектория, за чието изпълнение са необходими две смени на типа решение на обратната задача на кинематиката. Първата смяна е необходимо да се направи в точка 5, а втората в точка 8. Сравнение на времето за изпълнение на желаното движение, използвайки трите подхода: greedy approach, движение с една смяна и предложеният алгоритъм е показано на Фиг. 3.18.



Фиг. 3.18. Сравнение на времето за изпълнение на желаната траектория, използвайки трите разгледани подхода.

Както се вижда от фигурата, алчният алгоритъм е най-бърз до момента на смяната на типа решение, в точка 12. Тук е необходимо повече време за извършване на промяна в типа решение на обратната задача на кинематиката, тъй като смяната в точка 12 предизвиква отклонение от желаната траектория. Планираното движение с една смяна на типа решение в точка 7 е с по-малко време за изпълнение, тъй като смяната е извършена в позиция, която изисква по-малко ставно отместване. Както виждаме, предложеният алгоритъм има най-малко време за изпълнение. Извършени са две смени на типа решение и след втората смяна алгоритъмът взема преднина в сравнение с другите два алгоритъма.

Планиране на движение при наличие на статични препятствия

Когато даден манипулатор се намира в среда с препятствия, за които се знаят геометричните особености и позицията им остава постоянна по време на движението на робота, казваме че

роботът се намира в среда със статични препятствия. В този случай трябва да се планира движение, което може да се изпълни без опасност от сблъсък с налично препятствие. За тази цел ще се създаде алгоритъм за планиране на движение с избягване на препятствия от точка до точка за равнинен робот, като се вземат предвид кинематичните характеристики на робота, както и подходящите преходни точки. Тъй като ще се разглежда равнинно движение роботът е с допълнителни степени на свобода. Ставните му ограничения са дефинирани в (2.11).

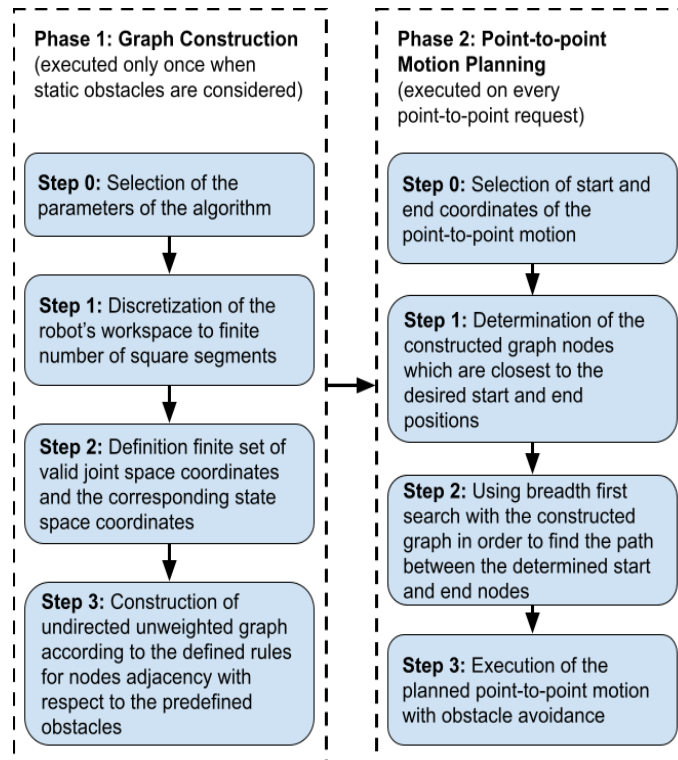
Работното пространство на робота може да бъде разделено на краен брой квадратни сегменти с дължина на страната a и краен брой точки. Допълнително, можем да разгледаме крайно множество A от валидни ставни координати, за които няма сблъсък с препятствия. Решението на правата задача на кинематиката за разглежданото множество ще бъде резултат от крайно множество B от координати в работното пространство на робота. След което, може да се създаде неориентиран непретеглен граф G по следния начин.

Върховете на графа съответстват на всяка една точка от множеството B . За всеки два различни върха u, v съществуват ребра (u, v) и (v, u) , ако:

- нормата на векторната разлика на съответстващите на u и v координати в работното пространство е по-малка от предварително дефинирана стойност α ;
- нормата на векторната разлика на съответните ставни координати на u и v е по-малка от предварително определена стойност β ;
- няма опасност от сблъсък с налично препятствие при движение на робота от u до v ;
- съответните координати в работното пространство се намират в рамките на един и същи сегмент от дискретизираното работно пространство (това позволява на робота да промени конфигурацията си от един тип решение на обратната задача на кинематиката на друг) или ако двете конфигурации имат един и същ тип решение на обратната задача на кинематиката. Това позволява на робота да извърши движение до съседна позиция без промяна на текущия тип решение.

Алгоритъмът за търсене в ширина (breadth first search, BFS) може да бъде използван за намиране на път между два върха в графа G . За построяването на графа G и изпълнението на BFS е достатъчно да се реши само правата задача на кинематиката. Това е изчислително ефективна задача за манипулатори с отворена кинематична верига. Ако желаната стартова или целева позиция на движението от точка до точка не съответства на никой от върховете в графа G , тогава алгоритъмът за планиране на движение трябва да намери най-близкия съответстващ връх на стартовата и/или целевата позиция. След което, пътят между двата върха се планира, посредством BFS алгоритъма. Движението между две позиции се планира като едновременно движение на всички ставни координати с еднаква продължителност. При планиране на движение от точка до точка, чрез използване на граф G и BFS се гарантира, че няма опасност от сблъсък с налично препятствие от работната зона на робота. Последователността на алгоритъма за планиране на движение е показана на Фиг. 3.19.

Алгоритъмът може да бъде разделен на две фази: построяване на граф G и планиране на движение от точка до точка. Фазата за създаване на графа G се изпълнява само веднъж преди планирането на движението от точка до точка и се вземат предвид само известни статични препятствия в работната зона на робота. В тази фаза се избират подходящи параметри на алгоритъма и се разделя работното пространство на робота на краен брой квадратни сегменти. Дефинира се крайно множество от ставни координати и съответстващите им координати в работното пространство, както и се конструира неориентиран непретеглен граф. При създаването на графа се вземат предвид дефинираните правила за съседство на два върха и информацията за налични препятствия.



Фиг. 3.19. Алгоритъм за планиране на движение при наличие на статични препятствия.

Втората фаза се състои в планиране на движение от точка до точка и се изпълнява при всяка заявка за движение до определена точка. В тази фаза се избират стартова и целева позиция за изпълнение на движение от точка до точка. Определят се върховете от графа, които са най-близо до желаните стартова и целева позиция. Използва се BFS алгоритъма върху построенния граф в първата фаза за намиране на път между стартовия и целевия връх. Фазата завършва с изпълнение на планираното движение от точка до точка с преодоляване на статични препятствия.

Управление на движение при наличие на динамични препятствия

Когато даден манипулатор работи съвместно с други роботи или хора, казваме че се намира в среда с динамични препятствия. Движението на робота не може да бъде разглеждано като движение от точка до точка, тъй като разглеждаме случай, в който роботът се намира в динамична среда. Във всеки един момент робот, човек или друг обект може да попадне в неговото работно пространство. За тази цел управляващата система на робота трябва динамично да планира неговата траектория в зависимост от наличните препятствия и успешно да ги преодолее. Управляващата система на робота трябва да планира новата траектория по такъв начин, че отклонението от предварително определената траектория да е възможно най-малко.

Нека да разгледаме роботизирана система с n звена, която трябва да изпълни предварително дефинирана равнинна траектория $\theta_d(t)$, където $t = [0, T]$ е времевият интервал. Траекторията θ_d е планирана по такъв начин, че роботът ще преодолее всяко едно предварително известно статично препятствие. Управляващата система трябва да открие всяко едно новопоявило се препятствие и да промени движението на робота, така че препятствието да бъде заобиколено без опасност от сблъсък с него. Откриването на препятствията и планирането на нова трябва да се изпълняват в реално време. Хващачът на робота е необходимо да се позиционира на желаната целева позиция, преодолявайки наличните препятствия в работното пространство на робота.

Препятствията се представят в декартови координати. Когато се появи ново препятствие, управляващата система на робота трябва да открие частта от траекторията, която няма да може да бъде изпълнена безопасно и да я блокира. След което, управляващата система е необходимо

да планира отново траекторията по такъв начин, че роботът да избегне препятствието. Алгоритъмът за планиране на траектория при наличие на динамични препятствия е базиран на теория на графите и използва алгоритъм за търсене в ширина. По този начин се извършва изчислително ефективно промяната на вече планираното движение. Това дава възможност за прилагане на алгоритъма в реално време. Целта е в края на движението роботът да позиционира изпълнителното си звено в предварително зададена крайна позиция. При робот с три звена, какъвто е изследваният в дисертацията робот, броят на различните ставни конфигурации за една и съща позиция на хващача на робота може да бъде безкраен. Това е необходимо да се вземе предвид от управляващата система при планиране на траекторията.

Препятствията се откриват, когато се появят в работното пространство на робота. За планиране на траектория в реално време работното пространство на робота може да се представи като граф с върхове, съответстващи на всяко едно възможно множество от ставни координати, отговарящи на позицията на хващача на робота в работното пространство. Ребрата на графа свързват два върха в зависимост от тяхната съответстваща позиция в работното пространство. Два върха от графа са свързани, ако техните съответстващи позиции са еднакви или съседни и ако необходимото движение е под предварително зададен прагов параметър (*threshold*). След това, работното пространство на робота може да бъде разделено на малък брой позиции, за да се намали броят на върховете и ребрата в графа. Това разделяне се определя от параметъра за разделителна способност на алгоритъма (*resolution*).

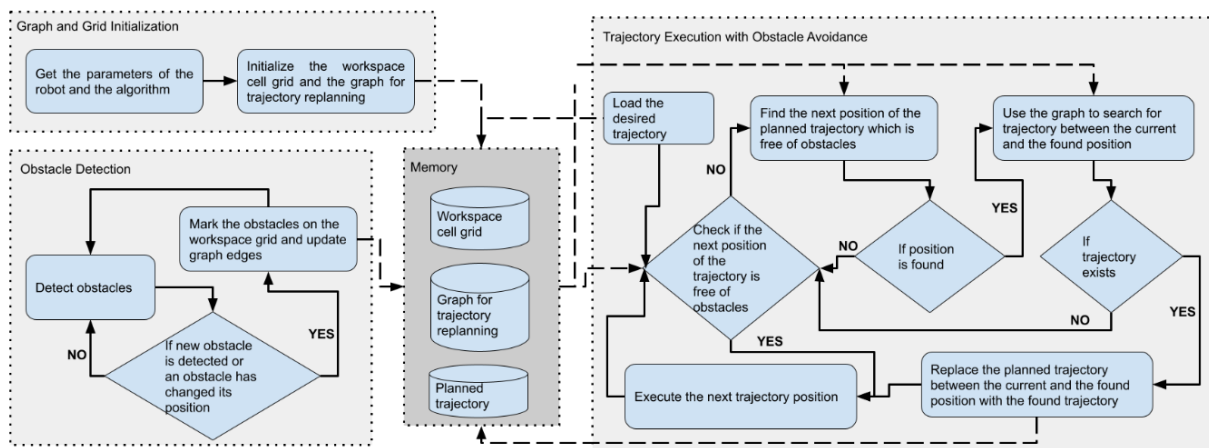
По време на изпълнението на траекторията се следи за препятствия и ако се появи препятствие, някои ребра от графа се маркират като блокирани. Когато роботът попадне в ситуация близка до сблъсък с налично препятствие, управляващата система намира частта от траекторията, която трябва да бъде повторно планирана и прилага алгоритъм за търсене в широчина на граф (breadth first search, BFS) за планиране на траектория в съответствие със създадения граф. Алгоритъмът има три основни стъпки: **създаване на граф, проследяване на препятствия и изпълнение на траектория и повторно планиране на траектория**. В обща памет се пази информация за създадения граф, планираната траектория и клетките, на които е разделено работното пространство. По време на всяка една от стъпките се следи за промяна в графа и планираната траектория и съответната информация в общата памет се обновява.

Създаването на графа зависи от следните параметри на алгоритъма: броят на звената n , дължината на звената a_i , където $1 \leq i \leq n$, *threshold* стойността, която определя кога две различни ставни конфигурации се считат за съседни и *resolution* параметърът, който разделя работното пространство на робота на определен брой позиции. Работното пространство на робота се представя като мрежа от клетки (*workspace_grid*). Всяка клетка представлява позиция от работното пространство на робота. След което, се създава граф, като всеки връх съответства на решение на обратната задача на кинематиката, което позиционира хващача на робота в центъра на клетката. За всеки връх от графа се намира съответната клетка от мрежата, нейната съседна клетка и съответните върхове от графа. Алгоритъмът добавя ребро в графа, свързвайки онези върхове, за които съответната клетка на мрежата е една и съща или съседна и необходимото движение отговаря на *threshold* параметъра.

Проследяването на препятствия зависи от следните параметри: *obstacles* - описание на намерените препятствия, *workspace_grid* - мрежа на работното пространство, създадена по време на фазата за създаване на графа, *desired_trajectory* - първоначално желаната и планирана траектория, *current_time* – текущо време и *reaction_time* - предварително дефинирана константа, която отчита възможностите на робота да спре движението си, което определя колко близо до препятствието може да бъде планирана новата траектория. Когато се открие дадено препятствие се отбелязва в мрежата от клетки (*workspace_grid*) и ребровете от графа, които свързват върховете, съответстващи на клетките се маркират като неизползваеми. Алгоритъмът проверява и дали

препятствието е ново или е сменило своята позиция. Откриването на препятствие се извършва в реално време.

По време на проследяването на изпълнението на траекторията и движението на препятствията, е възможно управляващата система да планира отново движението, поради новопоявили се препятствия. **Повторното планиране на траекторията** може да се извърши, посредством генерирания граф и стандартния BFS алгоритъм и зависи от следните параметри: *workspace_grid* – създадена в първата фаза мрежа на работното пространство от генерирания граф, *graph* – също генериран в първата фаза, *start_node* и *end_node* – начален и финален връх, намерен по време на фазата за следене (monitoring phase). Тези върхове съответстват на началото и края на блокираната част от зададената траектория. Алгоритъмът проверява дали следващата позиция от планираната траектория е безопасна за изпълнение (няма опасност от сблъсък с препятствие). Ако няма опасност от сблъсък с препятствие, роботът изпълнява следващата позиция от траекторията. В противен случай, алгоритъмът намира позиция от планираната траектория, която не е блокирана от препятствие. С помощта на създадения граф и BFS алгоритъма се намира път между текущата позиция на хващача на робота и следващата безопасна за изпълнение позиция. Ако такава траектория не може да бъде намерена се търси следваща точка от траекторията. След като е намерена, новата траектория заменя частта от планираната траектория между текущата позиция и новата намерена позиция и роботът продължава своето движение с изпълнение на следващата точка от траекторията. Планирането на новата траектория се извършва в реално време. На Фиг. 3.24 са показани всички стъпки на алгоритъма за планиране на траектория и преодоляване на динамични препятствия.



Фиг. 3.24. Алгоритъм за планиране на траектория и преодоляване на динамични препятствия.

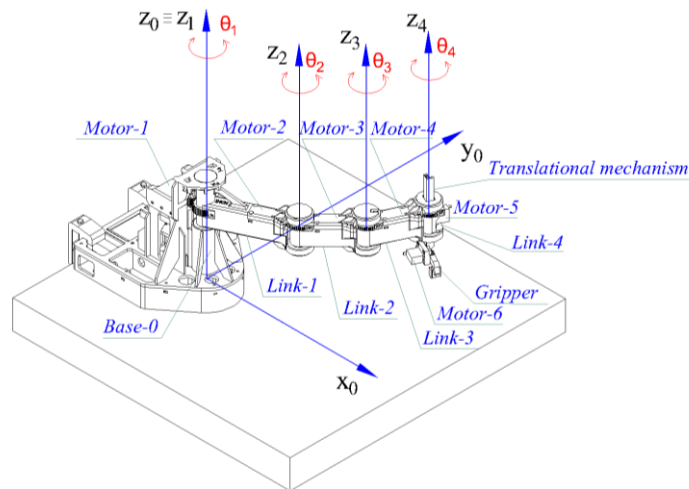
Глава 4. Експериментална верификация, чрез 3D принтиран робот

Тази глава верифицира приложимостта на предложените алгоритми за планиране на движение на равнинен робот с допълнителни степени на свобода при наличие на статични и динамични препятствия в работната му зона, чрез компютърна симулация и реален експеримент с 3D принтиран прототип на робота. Приложният принос на тази глава се заключава в създаване на компютърна симулация с помощта на симулационния софтуер Webots [37] и реални експерименти с прототипа на робота. Експериментите имат за цел да валидират предложените подходи за избягване на препятствия. Тази глава представя резултатите от изпълнението на задачи 4 и 5 от задачите на дисертационния труд.

Проектиране на 3D принтиран прототип на робот

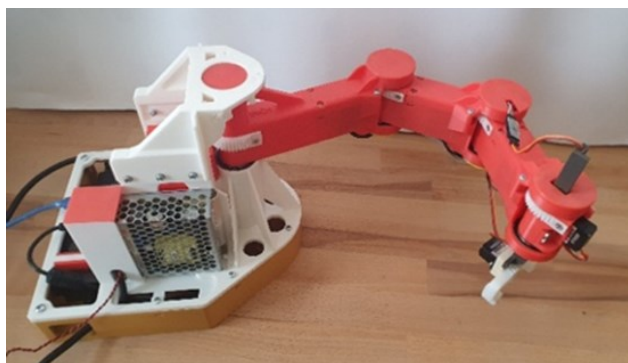
Използвана е технологията на 3D печата за създаване на прототип на разглеждания робот. За създаването на 3D модела на желания робот (Фиг. 4.1) е използвана софтуерната програма

AutoCAD. Едно от изискванията към робота е да бъде с допълнителни степени на свобода, когато изпълнява равнинни движения. Затова проектираният робот ще има 4 ротационни успоредни оси.



Фиг. 4.1. 3D модел на равнинен робот с допълнителни степени на свобода.

Роботът се състои от Base-0 – основа, Link-1 – първо звено, Link-2 – второ звено, Link-3 – трето звено, Link-4 – четвърто звено, Translational mechanism - транслационен механизъм, Gripper – хващач. В основата на робота е поставена управляващата електроника. Първият мотор Motor-1 е монтиран в основата на робота и задвижва първото звено. За да може транслационният механизъм да извършва движение по оста z , първото звено е закрепено към основата на височина от 150 mm по z . Вторият мотор е позициониран в края на звено 1 и задвижва второто звено. По аналогичен начин е монтиран и третият мотор в края на второто звено и задвижва третото звено. Четвъртият мотор се намира в края на третото звено и променя ориентацията на хващача. Петият мотор Motor-5 отговаря за задвижването на транслационния механизъм, а шестият мотор Motor-6 за отварянето и затварянето на пръстите на хващача. Готовият 3D прототип на разглеждания робот е показан на Фиг. 4.5. Материалът, който е избран за изготвяне на модела е PLA.



Фиг. 4.5. 3D принтиран прототип на робот.

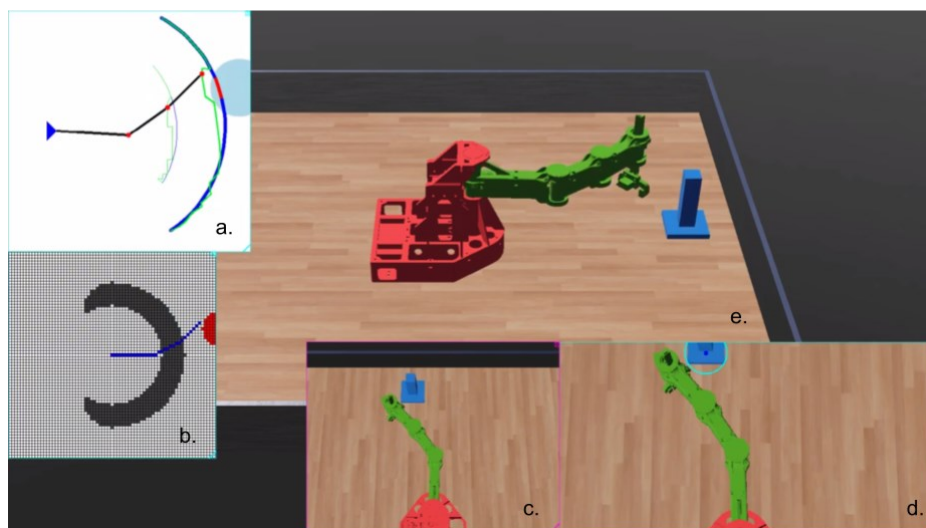
Експериментална верификация в симулационна среда Webots

Създаден е модел на физическия 3D принтиран робот в симулационна среда Webots. За тази цел е използван 3D моделът на робота, създаден в AutoCAD. Моделът на всяко едно звено на робота се записва като .stl файл. След това се създава .proto файл в Webots, където се импортират файловете на робота и се определят ставните ограничения и геометрията му. В симулационната среда могат да се добавят и различни сензори, камери, както и препятствия или други необходими обекти. За целите на този експеримент са добавени камера и препятствие с дължина на страната 31 mm и височина 120 mm. Целта на експеримента е да докаже, че предложеният

алгоритъм за планиране на движение с преодоляване на динамични препятствия е ефективен и подходящ за такъв тип робот. Роботът трябва да достигне крайната позиция от зададената траектория с изпълнителното си звено, преодолявайки наличните динамични препятствия в работната му зона. Роботът има дължини на звената и ставни ограничения, дефинирани съответно в (2.10) и (2.11). Алгоритъмът представя работното пространство като мрежа от клетки, параметърът *resolution* е избран да бъде 10 mm. Желаната траектория е дефинирана в (4.3):

$$\theta_1(t) = 0.4 - \frac{0.7t}{5}, \theta_2(t) = 0.5 - \frac{0.9t}{5}, \theta_3(t) = 0.5 - \frac{0.12t}{5}, t = [0,5] \quad (4.3)$$

По време на движението на робота препятствието ще променя своята позиция и роботът трябва успешно да го заобиколи. Когато симулационната камера разпознае препятствието, позициите в графа, които са блокирани от препятствието се отбелязват като недостъпни. Частта от траекторията, която е блокирана от препятствието се маркира в червено (Фиг. 4.8.a).



Фиг. 4.8. Блокиране на част от траекторията от динамично препятствие.

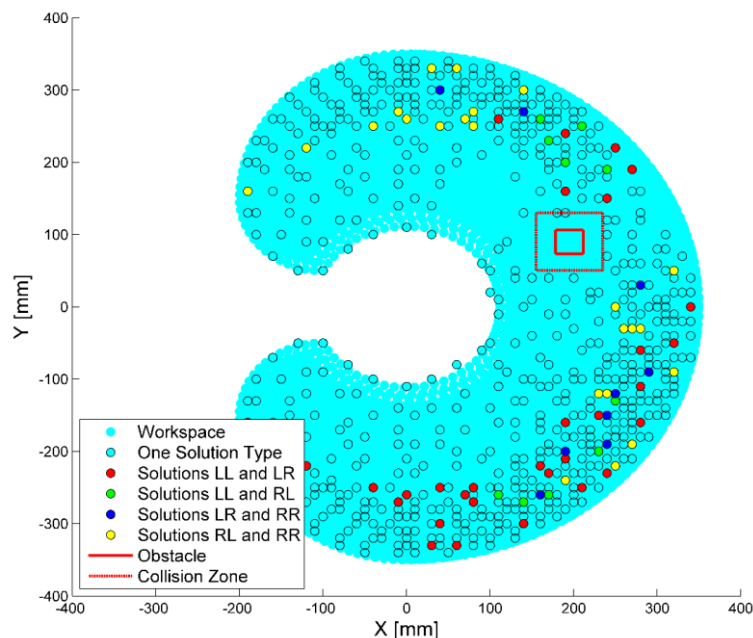
Управляващата система трябва да планира в реално време нова траектория, която успешно да заобиколи движещото се препятствие. Допълнително, новата траектория трябва да е планирана по такъв начин, че максимално бързо да върне хващача на робота към целевата траектория. Зелената линия представлява новата планирана траектория (Фиг. 4.8.a). По време на повторното планиране на траекторията е възможно препятствието отново да се премести по такъв начин, че отново да застъпва желаната траектория. Затова управляващата система трябва да следи постоянно по време на движението на робота за промяна на позицията на вече откритите препятствия, както и за новопоявили се такива. При появило се препятствие третото звено започва да се сгъва. Целта на движението на третото звено е да намали разстоянието между него и основата на робота. За всяка една нова позиция, която изпълнява третата ротационна става се проверява дали може да се създаде безопасна траектория и ако има такава, роботът продължава своето движение. Траекторията е препланирана успешно и роботът успява да достигне желаната крайна позиция с желаната ставна конфигурация.

Заобикаляне на препятствие

За експерименталната верификация на алгоритмите за управление на траектория се използва създадения прототип на равнинен робот. За валидиране на предложения алгоритъм в Глава 3 за планиране на движение при наличие на статични препятствия е поставена следната задача.

Роботът трябва да хване и пренесе определен предмет, докато изпълнява движение от точка до точка и преодолява налични препятствия в работното му пространство.

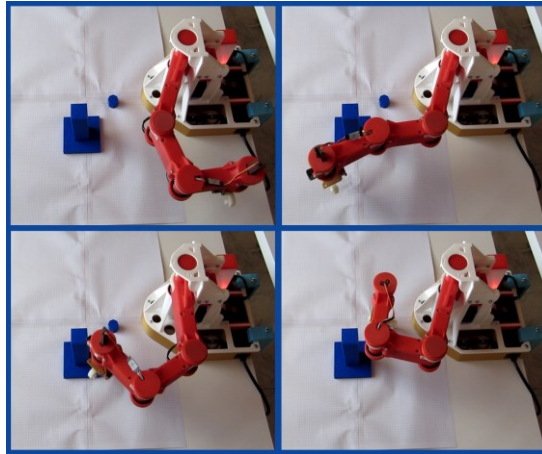
Препятствието е поставено на позиция с координати (195,90) в O_{xy} равнината. Препятствието има дължина на страната 31 mm и е високо 120 mm. Антропоморфният робот започва движението си от позиция с координати (-100, 210) в равнината и трябва да достигне предмет, поставен на позиция с координати (120, 10) в равнината. Роботът има ставни ограничени, дефинирани в (2.11). Първата стъпка от предложения подход е да се създаде крайно множество **A** от валидни ставни координати. Възможният диапазон от валидни ставни координати се дискретизира линейно и се разглеждат 10 стойности за всяка координата. Това ще даде резултат от максимум 1000 точки за множеството **A**. След проверка на всички възможни конфигурации за възможен удар с препятствие, получаваме множествата **A** и **B**, всяко с 811 точки. Дължината на страната на сегментите на работното пространство α е избрана да бъде 10 mm. На Фиг. 4.15 са показани различните типове сегменти, означени с точките от множеството **B** със съответния им тип решение на обратната задача на кинематиката.



Фиг. 4.15. Точките от работното пространство, които са в графа.

За построяването на графа **G** нормата на работното пространство α е зададена на 50 (координатите в работното пространство са в mm), а нормата на ставните координати β е 0.8 (ставните координати са в радиани). С тези параметри създаденият неориентиран граф **G** се състои от 811 върха и 5104 ребра. Алгоритъмът за планиране на движение определи позицията (-97.18, 210.19) като най-близка до желаната стартова позиция и позицията (111.07, 12.78) като най-близка до желаната крайна позиция. На Фиг. 4.17 са показани няколко ставни конфигурации, които са близки до препятствието. За намиране на път между двете позиции се използва BFS алгоритъма.

Той намира път с 22 върха от началната позиция до крайната. Алгоритъмът е изпълнен на една нишка на Intel® Core™ i7-4710HQ CPU. Графът е създаден за 19.8 секунди. Бяха необходими 0.019 секунди на BFS алгоритъма, за да планира движението на робота. Алгоритъмът е тестван и с недостижима точка (100, -300) за планиране на движение. На BFS алгоритъма му бяха необходими 0.24 секунди, за да установи, че не може да планира движение. Това е и максималното необходимо време за планиране на движение от точка до точка.



Фиг. 4.17. Избрани ставни конфигурации от изпълнението на движението.

Заобикаляне на препятствия в реално време

След валидиране на предложения алгоритъм за изпълнение на движение при наличие на статични препятствия в работната зона на робота може да се премине към провеждане на експеримент за изпълнение на предварително зададена траектория при наличие на динамично движещи се препятствия в работното пространство на робота. Алгоритъмът за заобикаляне на динамични препятствия е разширение на алгоритъма за преодоляване на статични препятствия и е представен в трета глава и компютърно валидиран в симулационна среда Webots.

Целта на експеримента е да покаже, че за проектирания робот този алгоритъм може да работи в реално време. За разлика от предишния експеримент, в който позицията на препятствието беше предварително дефинирана, в този експеримент нямаме предварителна информация за позицията на препятствието. За да работи алгоритъмът в реално време е необходимо във всеки един момент да може да се получи информация за позицията на препятствието.

За целите на валидация на алгоритъма, без ограничение на общността може да се разгледа препятствие, което е едноцветно и което ще се движи само в равнината O_{xy} . Неговата позиция по z няма да се променя, а височината му ще е фиксирана и константна. Такова препятствие може да бъде открито в реално време, чрез обикновена цветна камера и алгоритми за обработка на изображения. Камерата не е необходимо да е на точно определена позиция, защото можем да направим първоначална калибрация. Важното е да не променя своята позиция по време на управлението. За целите на експеримента е използвана цветна камера Canyon. Камерата се свързва посредством USB към компютър, където изпраща отчетените данни и се извършва тяхната обработка. За целите на експеримента се използват 3D принтирания робот, препятствие и камера за разпознаване на динамично движещия се обект. Тъй като експериментът изисква динамично движещи се обекти е необходимо периодично препятствието да променя своята позиция. За тази цел, то ще бъде ръчно премествано. Задачите, които трябва да се решат, за да може да се изчисли позицията на обекта са следните: **откриване на обекта, начин на калибрация и намиране на координатите на пространствените координати** в калибрирано изображение. За имплементирането на тези задачи е използван програмният език Python и библиотеката за компютърно зрение OpenCV [38].

Първата задача е откриване на обекта и определяне на координатите на неговия център в изображението. Това може да стане, чрез филтриране по цвета на обекта в HSV пространството, бинаризация и намиране на центъра на най-големия регион в бинаризираното изображение.

След като могат да се определят координатите в кадъра, то може да се премине към калибрацията. Тя се състои в определяне на координатите на обекта в кадъра, когато той е поставен на четири предварително избрани позиции в пространството. Позициите в m са

следните: $(0.5, 0.4, 0.0)$, $(0.0, 0.4, 0.0)$, $(0.0, -0.4, 0.0)$ и $(0.5, -0.4, 0.0)$. Те са избрани такива, защото общата дължина на звената на робота е 0.35 m. Когато обектът е по-далече, няма как да има сблъсък и не се налага препланиране на траекторията. След като са определени съответствията на позициите вече може да се премине към същинската работа на алгоритъма.

Когато имаме съответствията на позициите може на всяка стъпка да се вземе само частта от кадъра, която попада в четириъгълника образуван от координатите, които са съответни на калибриращите позиции. В зависимост от разположението на камерата (което не е предварително определено) ще се получи четириъгълник, който не е правоъгълник. Може чрез операция за промяна на перспективата, да коригираме изображението и да го сведем до правоъгълник с правилни пропорции. Целта е да се преобразува от произволен четириъгълник до правоъгълник с размери 500 на 800 пиксела. Това може да стане с функция `warpPerspective` в OpenCV. Разгледаните четири позиции в пространството, използвани при калибрацията, имат следните координати като съответствия в нетрансформираното изображение: $(580, 219)$, $(521, 51)$, $(180, 45)$, $(126, 208)$ и съответните им координати в трансформираното изображение са: $(0, 0)$, $(499, 0)$, $(499, 799)$, $(0, 799)$.

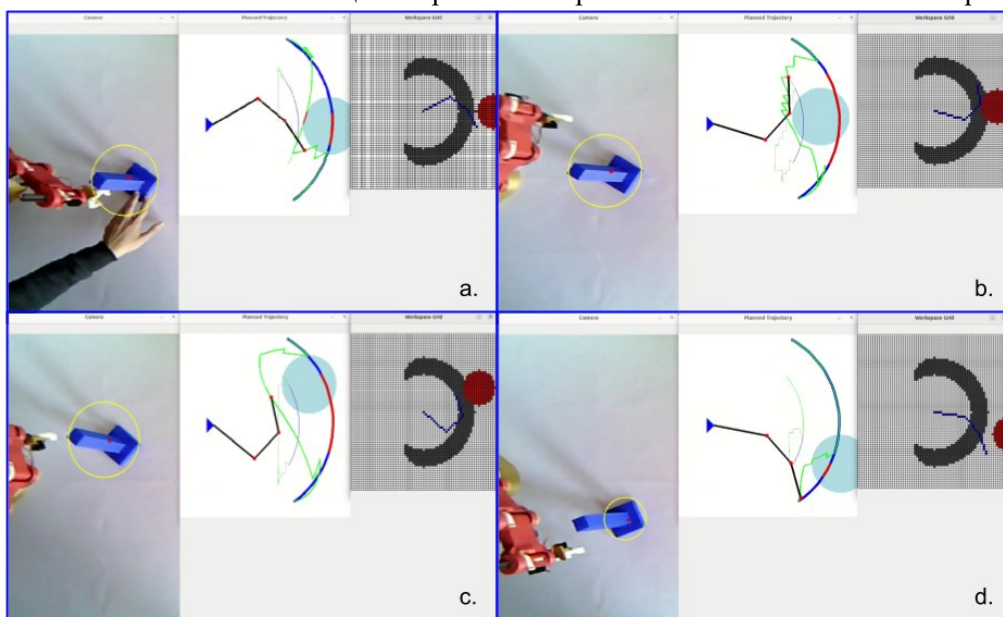
След като имаме кадър 500 на 800 пиксела може лесно да се намерят координатите на препятствието в него. Пак филтрираме по цвят и търсим центъра на най-големия регион. Този център ще има координати, които вече лесно могат да се съпоставят на пространствени. Центърът $(0.0, 0.0)$ в O_{xy} равнината на робота ще има координати $(0, 400)$ в изображението. Координатите на препятствието в изображението могат да се преобразуват към координати в работното пространство на робота с помощта на уравнения (4.4) и (4.5).

$$obstacle_x = \frac{(500.0 - object_x)}{500} * 0.5 \quad (4.4)$$

$$obstacle_y = \frac{(400.0 - object_y)}{400} * 0.4 \quad (4.5)$$

където $obstacle_x$ и $obstacle_y$ са съответно търсените координати x и y в O_{xy} равнината, $object_x$ и $object_y$ са координатите x и y на препятствието в изображението.

На Фиг. 4.19 са показани 4 позиции на робота по време на изпълнението на експеримента.



Фиг. 4.19. Избрани позиции от проведения експеримент.

Позицията на препятствието в равнината и съответната позиция в изображението за показаната извадка е: В първата позиция (Фиг. 4.19.а.) координатите на препятствието в равнината

са (0.34, 0) и координатите в изображението ще бъдат (157, 403). Във втората позиция (Фиг. 4.19.b.) координатите на препятствието в работното пространство на робота са (0.29, 0.02), а в изображението: (203, 380). Съответно в третата (Фиг. 4.19.c.) и четвъртата (Фиг. 4.19.d.) позиция координатите в равнината са: (0.27, 0.11) и (0.35, -0.1), които се съпоставят на координати: (221, 288) и (148, 505) в изображението.

Всяка една от изобразените позиции се състои от три полета. В първото поле е дадено изображение от камерата, която се използва за разпознаване на препятствието (камерата на управляващата система). Жълтата окръжност върху препятствието означава, че то е открито от управляващата система и са извършени успешно всички описани стъпки по неговото разпознаване. Второто поле от всяка една изобразена позиция илюстрира целевата траектория (синята дъга), препланирания път (зелената линия), частта от желаната траектория, която е блокирана от препятствието (обозначена с червен цвят) и позицията на препятствието (синият кръг).

Проведеният експеримент е изпълнен успешно. Управляващата система успешно разпознава появило се препятствие в работната зона на робота и планира нова траектория с цел избягване на сблъсък с него и възможно най-бързо връщане на хващача на робота към целевата траектория. По време на планирането е възможно препятствието да смени своята позиция, управляващата система следи за това и ако е необходимо планира нова траектория. За планирането на траекторията се използва алгоритъм за търсене в широчина в граф, което прави задачата изчислително ефективна и успешна за изпълнение в реално време.

Заклучение

В дисертационния труд са анализирани предимствата и недостатъците на роботите с допълнителни степени на свобода и е мотивиран изборът за използване на робот с допълнителни степени на свобода, при задачи свързани с преодоляване на препятствия в работната зона на робота. Изготвен е математически модел на робот с допълнителни степени на свобода. Правата задача на кинематиката е решена с помощта на конвенцията на Денавит-Хартенберг, а обратната задача, посредством геометричен метод за решение. Решенията на обратната задача на кинематиката на такъв тип робот могат да се класифицират на няколко типа, в зависимост от знака на втората и третата ставна координата. В дисертацията са разгледани тези типове решения и как те разделят работното пространство на робота на различни зони. Преходът от една зона в друга може да изисква смяна на типа решение, което може да доведе до отклонение от желаното движение. Затова са изследвани преходни точки, в които смяната на типа решение може да се извърши без отклонение от желания път.

Проектиран и реализиран е прототип на равнинен робот с 4 ротационни стави, посредством методите на 3D принтирането. Създадена е хардуерна и софтуерна система за управление на антропоморфния робот, която се базира на платформите с отворен код Arduino Mega 2560 и Wi-Fi модул ESP8266. Създаден е потребителски интерфейс, през който операторът може да следи състоянието на робота и да изпраща команди към него и е предложен алгоритъм за планиране на движение. Алгоритъмът взима предвид преходните точки и се базира на теорията на графите.

В дисертацията са предложени и създадени алгоритми за планиране на траектория при наличие на статични и динамични препятствия. Алгоритмите са валидирани, чрез създаване на компютърни експерименти и провеждане на реални експерименти с 3D принтирания прототип. При изпълнение на дадена задача при наличие на статични препятствия, управляващата система на робота има предварителна информация за позицията на препятствията. Задачата става по-сложна, когато роботът манипулира в среда с динамични препятствия. В този случай управляващата система трябва в реално време да следи за позицията на препятствията и когато дадено препятствие блокира планираното движение е необходимо препланиране на движението

на работа и максимално бързо връщане към целевата траектория. Проведените и описани експерименти доказаха, че предложените алгоритми за планиране на траектория при наличие на статични и динамични препятствия са подходящи за управление на такъв тип робот, изчислително ефективни са и това позволява да бъдат приложени в задачи, изискващи работа в реално време.

Всички задачи, които са поставени в началото на дисертационния труд са изпълнени. Целта за създаване на математически модел и прототип на равнинен антропоморфен робот с допълнителни степени на свобода и създаване на методи за неговото управление е постигната.

Перспективи за развитие

Изследванията върху алгоритмите за планиране на траектория при наличие на статични или динамични препятствия ще продължат и за в бъдеще. Може да се изследва по какъв начин изборът на параметрите на алгоритъма за планиране на траектория при наличие на статични препятствия влияе на изпълнението на алгоритъма. Също така, как фазата по построяване на графа може да се реализира с по-висока изчислителна ефективност или дори да се извърши в реално време при наличие на динамични препятствия.

Проведените експерименти показват, че проектираната хардуерна система е подходяща за управление на такъв тип робот. Но вътрешната памет на контролера Arduino е ограничена. Затова би било добре, за в бъдеще, да се добави външна памет към хардуерната система на робота.

Цитирани източници

- [1] V. Potkonjak, M. Popović, M. Lazarević and J. Sinanović, "Redundancy problem in writing: From human to anthropomorphic robot arm," in *Systems and Cybernetics, Part B: Cybernetics, IEEE Transactions*, 1999.
- [2] T. Petric, A. Gams, N. Likar and L. Zlajpah, "Obstacle avoidance with industrial robots," in *Springer International Publishing*, 2015.
- [3] St. Chiaverini, G. Oriolo and A. Maciejewski, "Redundant Robots," 2016.
- [4] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*, 2020.
- [5] G. Oriolo, M. Cefalo and M. Vendittelli, "Repeatable motion planning for redundant robots over cyclic tasks," *IEEE Transactions on Robotics*, pp. 1-14, 2017.
- [6] Ch. A. Klein and Br. E. Blaho, "Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators," *The International Journal of Robotics Research*, vol. 6, pp. 72-83, 1987.
- [7] V. Chembuly and H. Voruganti, "Trajectory planning of redundant manipulators moving along constrained path and avoiding obstacles," in *Procedia Computer Science*, 2018.
- [8] J. Hollerbach and Ki Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Journal on Robotics and Automation*, vol. 3, pp. 308-316, 1987.
- [9] A. R. Hirasawa and A. Kawamura, "Trajectory planning of redundant manipulators for minimum energy consumption without matrix inversion," in *Proceedings of International Conference on Robotics and Automation*, Albuquerque, 1997.
- [10] H. Voruganti and V. Chembuly, "An optimization based inverse kinematics of redundant robots avoiding obstacles and singularities," in *Advances in robotics Conference*, 2017.
- [11] T. Yoshikawa, "Manipulability of robotic mechanisms," in *IEEE International Conference on Robotics and Automation*, 1985.
- [12] Ch. Zhou, B. Huang and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, pp. 387-424, 2022.
- [13] R. Venkat, "Path Finding - Dijkstra's Algorithm," 2014.
- [14] S. Skiena, "Dijkstra's algorithm," in *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Addison-Wesley, Boston, Mass, USA, 1990, pp. 225-227.

- [15] Z. -c. Du, G. -Y. Ouyang, J. Xue and Y. -b. Yao, "A Review on Kinematic, Workspace, Trajectory Planning and Path Planning of Hyper-Redundant manipulators," in *10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2020.
- [16] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, 1999.
- [17] L. E. Kavraki, P. Svestka, J. C. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *IEEE Transactions on Robotics and Automation*, 2002.
- [18] A. Gupta, S. Agrawal, A. Deshmukh and P. Bhargava, "A Geometric Approach to Inverse Kinematics of a 3 DOF Robotic Arm," 2018.
- [19] Br. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control* (1st. ed.), Springer Publishing Company, Incorporated., 2009.
- [20] I. Chavdarov, V. Nikolov, B. Naydenov and G. Boiadjiev, "Design and Control of an Educational Redundant 3D Printed Robot," in *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 2019.
- [21] D. E. Orin and W. W. Schrader, "Efficient computation of the Jacobian for robot manipulators," *International Journal of Robotics Research*, pp. 66-75, 1984.
- [22] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *IEEE International Conference on Robotics and Automation*, 1985.
- [23] Buss, R. Samule , "Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods," 2009.
- [24] M. Kalasariya, V. Patel, A. Thakkar, "Comparative Study of Iterative Inverse Kinematics Methods for Serial Manipulators," *International Journal of Engineering and Technical Research*, 2018.
- [25] R. B. L. Barinka, "Inverse Kinematics - Basic Methods".
- [26] M. R. Gier, "Control of a robotic arm: Application to on-surface 3Dprinting," 2015.
- [27] E. Krustev, "Passing Through Jacobian Singularities in Motion Path Control of Redundant Robot Arms," in *Proceedings of the 27th International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2018)*, 2018.
- [28] Wampler and W. Charles, "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," in *IEEE Transactions On Systems, Man, And Cybernetics*, 1986.
- [29] I. Duleba, M. Opalka, "A Comparison Of Jacobian–Based Methods Of Inverse Kinematics For Serial Robot Manipulators," *International Journal of Applied Mathematics in Computer science*, vol. 23, pp. 373-382, 2013.
- [30] A. Mueller, "Modern Robotics: Mechanics, Planning, and Control," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 100-102, 2019.
- [31] Boiadjiev G, Krastev E, Chavdarov I, Miteva L, "A Novel, Oriented to Graphs Model of Robot Arm Dynamics," *Robotics*, vol. 10, no. 128, 2021.
- [32] Vinogradov, A. Kobrinski, Y. Stephenko, L. Tives, "Details of Kinematics of Manipulators with the Method of Volumes," *Mekanika Mashin*, vol. I, no. 5, pp. 5-16, 1971.
- [33] "Smart Servo Motors HerkuleX DRS-0101," [Online]. Available: <https://www.robotshop.com/en/herkulex-drs-0101-robot-servo.html>. [Accessed 01. 11. 2022].
- [34] "Homepage FeeTech,," [Online]. Available: <http://www.feetechrc.com/>. [Accessed 01 11 2022].
- [35] "Arduino Mega 2560," [Online]. Available: <https://store.arduino.cc/arduino-mega-2560-rev3>. [Accessed 01. 11. 2022].
- [36] "Wemos D1 mini," [Online]. Available: https://docs.zerynth.com/latest/official/board.zerynth.wemos_d1_mini/docs/index.html. [Accessed 01. 11. 2022].
- [37] "Webots," [Online]. Available: <https://cyberbotics.com/doc/guide/index>. [Accessed 25. 03. 2022].
- [38] "OpenCV," [Online]. Available: <https://opencv.org/>. [Accessed 20. 11. 2022].

Приноси в дисертационния труд

Взимайки предвид работата по дисертацията и резултатите получени от проведените изследвания и представени в дисертационния труд, мога да се формулират следните приноси:

Научноприложни приноси

- Създаден е подход за класифициране по тип на решенията на обратната задача на кинематиката за равнинен робот с допълнителни степени на свобода. (Глава 2, [П1], [Д3]).
- Извършен е анализ на работното пространство на равнинен робот с допълнителни степени на свобода в зависимост от наличните препятствия (Глава 2, [П4], [Д4]).
- Изследван е ъгълът на сервис в работното пространство на равнинен робот с допълнителни степени на свобода (Глава 2, [П2]).
- Създаден е алгоритъм за планиране на траектория на равнинен робот с допълнителни степени на свобода и ограничено ставно пространство, базиран на теорията на графите (Глава 3, [П1], [Д3]).
- Създаден е подход за планиране на движение при наличие на статични препятствия за равнинен робот с допълнителни степени на свобода (Глава 3, [П4], [Д4]).
- Създаден е алгоритъм за избягване в реално време на динамични препятствия в работното пространство на равнинен робот с допълнителни степени на свобода (Глава 3, [П5]).

Приложни приноси

- Проектирана е хардуерна и софтуерна система за управление на равнинен робот с допълнителни степени на свобода (Глава 3, [П3], [Д2]).
- Създаден е компютърен експеримент на разработените методи за планиране на траектория с помощта на симулационен софтуер Webots (Глава 4, [П6]).
- Проведени са реални експерименти с 3D принтиран прототип на равнинен робот с допълнителни степени на свобода с цел верификация на алгоритмите за планиране на траектория при наличие на статични или динамични препятствия в работното пространство на робота (Глава 4, [П4], [П5], [Д4]).

Публикации, доклади и участие в проекти, свързани с темата на дисертацията

Публикации

- П1. **Lyubomira Miteva**, Ivan Chavdarov, Kaloyan Yovchev, *Trajectory Planning for Redundant Robotic Manipulators with Constrained Joint Space*, 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2020, DOI: 10.23919/SoftCOM50211.2020.9238296, ISBN: 978-953-290-099-6, ISSN: 1847-358X, Ref Scopus, Ref IEEE Xplore, <https://ieeexplore.ieee.org/document/9238296>.
- П2. **Lyubomira Miteva**, Galya Pavlova, Roumen Trifonov, Kaloyan Yovchev, *Manipulability Analysis of Redundant Robotic Manipulator*, ACM International Conference Proceeding Series, 21st International Conference on Computer Systems and Technologies, CompSysTech 2020, Pages: 135-140, DOI: 10.1145/3407982.3407987, ISBN 978-145037768-3, Ref Scopus, SJR (0.182 – 2020), <https://dl.acm.org/doi/10.1145/3407982.3407987> (Наградена за най-добър доклад в секцията).
- П3. **Lyubomira Miteva**, Kaloyan Yovchev, *Hardware and Software Design for Redundant Robotic Manipulators*, CEUR Workshop Proceedings, 2020, Volume: 2656, Pages: 167-180, ISSN: 1613-0073, Ref Scopus, SCOPUS, SJR (0.177 – 2020), <https://ceur-ws.org/Vol-2656/paper17.pdf>.
- П4. **Lyubomira Miteva**, Kaloyan Yovchev, Ivan Chavdarov, *Point-to-point Motion Planning with Obstacle Avoidance for Hard Constrained Redundant Robotic Manipulators*, 2021 XXX International Scientific Conference Electronics (ET), 2021, DOI: 10.1109/ET52713.2021.9579820, ISBN: 978-166544518-4, Ref Scopus, Ref IEEE Xplore, <https://ieeexplore.ieee.org/document/9579820>.
- П5. Kaloyan Yovchev, **Lyubomira Miteva**, *Real-time Trajectory Replanning for Dynamic Obstacles Avoidance for Robotics Manipulators*, ACM International Conference Proceeding Series, 23rd International Conference on Computer Systems and Technologies, 2022, Pages: 45–50, DOI: 10.1145/3546118.3546135, ISBN 978-145039644-8, Ref Scopus, SJR (0.232 – 2021), <https://dl.acm.org/doi/abs/10.1145/3546118.3546135> (Наградена за най-добър доклад в секцията).
- П6. **Lyubomira Miteva**, Kaloyan Yovchev, Denis Chikurtev, *Software and Hardware Infrastructure for Research and Development of Intelligent Control for Robotic Manipulators*, 2022 XXXI International Scientific Conference Electronics (ET), 2022, DOI: 10.1109/ET55967.2022.9920270, ISBN: 978-166549878-4, Ref Scopus, Ref IEEE Xplore, <https://ieeexplore.ieee.org/document/9920270>.

Доклади

- Д1. **Manipulability Analysis of Redundant Robotic Manipulator**, международна конференция CompSysTech 2020. http://www.compsystech.org/_cst20
- Д2. **Hardware and Software Design for Redundant Robotic Manipulators**, международна конференция ISGT 2020. <https://isgt.fmi.uni-sofia.bg>
- Д3. **Trajectory Planning for Redundant Robotic Manipulators with Constrained Joint Space**, международна конференция SoftCom 2020. <http://softcom2020.fesb.unist.hr>
- Д4. **Point-to-point motion planning with obstacle avoidance for hard constrained redundant robotic manipulator**, международна конференция Electronics – ET 2021, гр. Созопол. <https://e-university.tu-sofia.bg/e-conf/?konf=24>

Участие в проекти, свързани с темата на дисертацията

- **Изследване и развитие на алгоритми със самообучение за взаимодействие между индустриални роботи и обекти в работната зона, № КП-060М47/5 от 27.11.2020 г. Фонд научни изследвания, МОН.**
- **Изследване и моделиране на нови роботи чрез нетрадиционни технологии и материали, ФНИ № 17/10 от 12.12.2017 г. Фонд научни изследвания, МОН.**
- **Разработка на методи и алгоритми за управление на манипулационни роботи, ФНИ на СУ № 80-10-23 от 18.03.2020 г.**
- **Разработка на методи и алгоритми за управление на манипулационни роботи, ФНИ на СУ № 80-10-89/2021.**
- **Разработка на методи и алгоритми за управление на манипулационни роботи, ФНИ на СУ № 80-10-70/10.05.2022**

Декларация за оригиналност

Декларирам, че представената във връзка с провеждането на процедура за придобиване на образователна и научна степен „доктор“ в Софийски университет „Св. Климент Охридски“ дисертация на тема: „Моделиране и управление на антропоморфен модел на робот“ е мой труд.

Цитиранията на всички източници на информация, текст, илюстрации, таблици, изображения и други са обозначени според стандартите.

Резултатите и приносите на проведеното дисертационно изследване са оригинални и не са заимствани от изследвания и публикации, в които нямам участие.

Подпис:

/Любомира Митева/

Биографични данни за автора на дисертационния труд

Любомира Митева е родена през 1994 г. в гр. Пазарджик. През 2017 г. завършва бакалавър по Софтуерно инженерство във факултета по математика и информатика (ФМИ) на Софийски университет „Св. Климент Охридски“. През 2019 г. завършва магистратура в направление информатика и компютърни науки, специалност „Мехатроника и роботика“ във ФМИ. Дипломната работа е в областта на антропоморфните роботи. През 2020 г. е зачислена като редовен докторант към катедра „Компютърна информатика“ на ФМИ, Софийски университет „Св. Климент Охридски“. Темата на дисертацията е „Моделиране и управление на антропоморфен модел на робот“.

От 2020 г. е хоноруван преподавател във ФМИ и води упражнения към курсовете „Въведение в програмирането“ и „3D моделиране, принтиране и приложения в роботиката“.

Научните интереси на Любомира Митева са свързани с проектиране, създаване и управление на манипулационни и мобилни роботи.

Благодарности

Бих искала да изкажа специални благодарности на научните ми ръководители проф. д-р Евгений Кръстев и доц. д-р Иван Чавдаров за градивните съвети, коментари и забележки.

Изказвам благодарности и към членовете на катедри „Компютърна информатика“ и „Мехатроника, роботика и механика“ за съветите и напътствията.

Благодаря и на Институт по роботика – БАН за подкрепата и предоставената възможност за участие в научни проекти.