

UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET



Nikola Milić

**Sistem za eksperimentalno ispitivanje koncepta servo-upravljanog
industrijskog manipulacionog robota sa četiri stepena slobode**

-MASTER RAD-

Podgorica, 2023.

PODACI I INFORMACIJE O STUDENTU

Ime i prezime: Nikola Milić

Datum i mjesto rođenja: 05.05.1999., Nikšić, Crna Gora

Naziv završenog osnovnog studijskog programa i godina završetka studija: Energetika i automatika, 2020.

INFORMACIJE O MASTER RADU

Naziv master studija: Automatika i industrijska elektrotehnika

Naslov master rada: Sistem za eksperimentalno ispitivanje koncepta servo-upravljanog industrijskog manipulacionog robota sa četiri stepena slobode

Fakultet/Akademija na kojem je rad odbranjen: Elektrotehnički fakultet, Podgorica

UDK, OCJENA I ODBRANA MASTER RADA

Datum prijave master rada: 13.02.2023.

Datum sjednice Vijeća univerzitetske jedinice na kojoj je prihvaćena tema: 11.07.2023.

Komisija za ocjenu teme i podobnosti kandidata: Prof. dr Božo Krstajić
Vanr. prof. dr Žarko Zečević
Prof. dr Milovan Radulović

Mentor:

Prof. dr Milovan Radulović

Komisija za ocjenu rada:

Prof. dr Božo Krstajić

Vanr. prof. dr Žarko Zečević

Prof. dr Milovan Radulović

Komisija za odbranu rada:

Prof. dr Božo Krstajić

Vanr. prof. dr Žarko Zečević

Prof. dr Milovan Radulović

Datum odbrane:

Datum promocije:

Ime i prezime autora: Nikola Milić, BSc

ETIČKA IZJAVA

U skladu sa članom 22 Zakona o akademskom integritetu i članom 18 Pravila studiranja na master studijama, pod krivičnom i materijalnom odgovornošću, izjavljujem da je master rad pod naslovom

„Sistem za eksperimentalno ispitivanje koncepta servo-upravljanog industrijskog manipulacionog robota sa četiri stepena slobode”,

moje originalno djelo.

Podnosilac izjave,



Nikola Milić, BSc

U Podgorici, dana 11.10.2023. godine

Predgovor

U današnjem globalnom okruženju proizvodnje i industrije, brza i precizna manipulacija materijalima postaje sve važnija komponenta za postizanje konkurentske prednosti. Industrijski manipulacioni roboti igraju ključnu ulogu u automatizaciji različitih procesa, a razvoj naprednih tehnologija servo upravljanja otvara nova vrata za preciznost i efikasnost ovih sistema. Ovaj master rad predstavlja rezultate istraživanja i razvoja sistema za eksperimentalno ispitivanje koncepta servo-upravljanog industrijskog manipulacionog robota sa četiri stepena slobode.

U svijetu industrijskih robota, stepen slobode predstavlja osnovnu karakteristiku koja određuje sposobnosti robota za izvođenje različitih zadataka. Dodavanje servo motora za svaki stepen slobode omogućava izuzetnu preciznost i brzinu u izvođenju pokreta, otvarajući nove mogućnosti za primjenu u preciznoj montaži, obradi materijala i drugim granama industrije gdje su potrebne visoke performanse.

Shodno navedenom, u ovom radu je obrađeno razvijanje i eksperimentalno testiranje prototipa servo-upravljanog manipulacionog robota sa četiri stepena slobode. Ovaj rad može poslužiti kao koristan materijal studentima osnovnih i master studija tehničkih fakulteta, kao i svima onima koji se bave ili proučavaju industrijsku robotiku. Rad ima kako teorijski, tako i praktični značaj, jer su teorijski koncepti robotike preslikani na realni sistem. Autor se zahvaljuje svom mentoru, profesoru Milovanu Raduloviću koji je bio podrška tokom pisanja rada, Inovaciono-preduzetničkom centru Tehnopolis i Montenegro Space Research organizaciji za pružanje tehničke podrške u procesu izrade robotskog manipulatora.

Izvod rada

Jedan od principa na kojem počivaju Treća industrijska revolucija, koja je započela 70-ih godina prošlog vijeka, i Četvrta industrijska revolucija, koja se trenutno implementira, je automatizacija industrijskih procesa, tj. njihovo odvijanje bez intervencija od strane čovjeka. Jedan od primjera automatizacije u industriji su roboti koji izvršavaju programirane sekvence bez ljudske intervencije. Industrijski roboti su automatizovani sistemi koji koriste računar kao inteligentni dio upravljanja. Glavni faktori širenja robotske tehnologije u industrijskoj proizvodnji su smanjenje troškova, povećanje produktivnosti i najbitnije, udaljavanje čovjeka od opasnih zadataka u industriji. Mogućnosti primjene robota u industriji su praktično neograničene. Na bazi prethodno opisane aktuelnosti robotike u industriji, predmet istraživanja ovog rada je problematika dizajna i izrade jednostavnog prototipa industrijskog robotskog manipulatora, programiranje manipulatora i upravljanje njime korišćenjem osnovnih principa kinematike.

U radu je najprije predstavljen proces dizajna robotskog manipulatora koji podrazumijeva odabir komponenti i 3D modelovanje segmenata robota i radne platforme. Nakon završenog dizajna i odabira komponenti, vrši se izrada segmenata i sklapanje svih djelova u jednu funkcionalnu cjelinu. Nakon dizajna, izrade segmenata i sklapanja, vrši se kinematičko modelovanje manipulatora. Kinematički model robota se koristi za rješavanje zadatka inverzne kinematike, odnosno prevođenje koordinata zadatah u radnom prostoru u koordinate zglobova. Rezultati dobijeni rješavanjem inverznog problema se zatim koriste za generisanje referentnih trajektorija zglobova manipulatora. Kako je fokus rada, osim na dizajnu, i na upravljanju manipulatorom, u radu se vrši analiza primjene različitih principa i algoritama upravljanja, kao i tehnika generisanja trajektorije.

Kako izrađeni manipulator predstavlja niskobudžetni prototip industrijskog manipulatora, izvršen je odabir i prilagođavanje izloženih principa i algoritama upravljanja upravljanja i tehnika generisanja trajektorije. Iako nije postignuto precizno praćenje referentnih trajektorija, odabrane metode upravljanja su se pokazale kao efikasne za izvršavanje radnih zadataka koji uključuju premještanje radnog predmeta iz jednog položaja u drugi, što predstavlja jednu od najčešćih radnji koje robotski manipulatori izvršavaju u industrijskim pogonima.

Za razliku od većine radova na ovom polju u literaturi, ovaj rad uključuje i eksperimentalni doprinos. U radu je pokazano na koji način se teorijski principi mogu preslikati na realni sistem, uz ograničenja koja su nametnuta od strane komponenti sistema.

Ključne riječi: industrija, upravljanje, robotski manipulator, kinematika robota

Abstract

One of the basic principles of the Third Industrial Revolution, which began in the 1970s, and the Fourth Industrial Revolution, which is currently being implemented, is the automation of industrial processes, i.e. their running without human intervention. One example of automation in industry is robots, which execute programmed sequences without human intervention. Industrial robots are automated systems that use a computer as an intelligent control system part. The main factors of the spread of robotic technology in industrial production are cost reduction, increase of productivity and, most importantly, removal of humans from dangerous tasks in industry. The possibilities of applying robots in industry are practically unlimited. Based on the previously described presence of robotics in industry, the subject of research in this paper is the problem of designing and making a simple prototype of an industrial robotic manipulator, manipulator programming and control using the basic principles of kinematics.

Firstly, this paper presents the process of designing a robot manipulator, which includes the selection of components and 3D modeling of robot segments and the work platform. After design completion and selection of components, the segments are made and all parts are assembled into one functional unit. After design, segment manufacturing and assembly, kinematic modeling of the manipulator is performed. The kinematic model of the robot is used to solve the task of inverse kinematics, i.e. calculating joint coordinates using the given the workspace coordinates. The results obtained by solving the inverse problem are then used to generate the reference trajectories of the manipulator joints. As the focus of the work, apart from design, is also on manipulator control, the paper analyzes the application of various control principles and algorithms, as well as trajectory generation techniques.

As the manufactured manipulator represents a low-budget prototype of an industrial manipulator, the selection and adjustment of the described principles and control algorithms and the trajectory generation techniques was carried out. Although precise tracking of reference trajectories was not achieved, the selected control methods proved to be effective for performing tasks that include moving a workpiece from one position to another, which is one of the most common actions performed by robotic manipulators in industrial plants.

Unlike most papers in the field of robotics, this paper also includes an experimental contribution. The paper shows how the theoretical principles can be mapped to a real system, with respect to the limitations imposed by the system components.

Keywords: industry, control, robotic manipulator, robot kinematics

Sadržaj

1. Uvod	1
2. Robotski manipulator	4
2.1. Pojam robotskog manipulatora	4
2.1.1. Mehanički sistem.....	5
2.1.2. Energetski sistem	6
2.1.3. Mjerni sistem.....	6
2.1.4. Upravljački sistem	7
2.2. Dizajn robotskog manipulatora	7
2.2.1. Dizajn i izrada mehaničke strukture manipulatora.....	8
2.3. Elementi upravljačkog sistema	16
2.3.1.1. Arduino mikrokontroler.....	16
2.3.1.2. Servo motori.....	18
2.3.2 Softver	22
2.4. Izbor aktuatora robotskog manipulatora	23
2.4.1. Moment kao fizička veličina.....	23
2.4.2. Obrtni moment zglobova.....	24
2.5. Laboratorijska postavka manipulacionog robota.....	26
3. Kinematika robota	30
3.1. Osnovni teorijski pojmovi	30
3.1.1. Kinematički lanci	32
3.1.2. Stepeni slobode kinematičkog lanca	33
3.2. Radni prostor	36
3.3. Osnovni pojmovi kinematike robota	38
3.3.1. Spoljašnje i unutrašnje koordinate.....	38
3.3.2. Homogene transformacije	40
3.3.3. Transformacija translacije.....	43
3.3.4. Transformacije rotacije.....	44
3.4. Denavit – Hartenbergova notacija.....	47
3.4.1. Definisane homogenih matrica transformacija pomoću D – H parametara	50
3.4.2. Određivanje orijentacije završnog uređaja	51
3.5. Direktni kinematički problem.....	53
3.5.2. Rješenje direktnog kinematičkog problema za SPT – 3000	54
3.6. Inverzna kinematika robotskog manipulatora.....	56
3.6.1. Analitičko rješenje inverznog kinematičkog problema	57
3.6.2. Numeričko (iterativno) rješenje inverznog kinematičkog problema.....	58
3.7. Rješavanje inverznog kinematičkog problema	60
3.7.1. Planarni manipulator sa dva stepena slobode.....	60
3.7.2. Rješenje inverznog kinematičkog problema za SPT – 3000	63
4. Planiranje kretanja i koncepti upravljanja.....	66

4.1. Upravljanje robotom	66
4.2. Principi upravljanja robotom	68
4.2.1. PID regulator	68
4.2.2. Servo upravljanje jednosmjernim motorom	71
4.2.3. Nivoi i strategije upravljanja robotom	74
4.2.4. Tipovi upravljanja	77
4.3. Planiranje trajektorije	77
4.3.1. Put i trajektorija	78
4.3.2. Trajektorije u prostoru zglobova	78
4.3.2.1. Kretanje od tačke do tačke	79
4.3.2.2. Kretanje kroz niz tačaka	83
4.3.3. Trajektorija u radnom prostoru	87
4.3.3.1. Definicija puta u prostoru	87
4.3.3.2. Pozicija	90
4.3.3.3. Orijehtacija	92
5. Eksperiment.....	94
5.1. Definicija i rješavanje radnog zadatka	97
5.2. Tok i rezultati eksperimenta	101
6. Zaključak.....	108
Literatura.....	109
Prilog	112

1. Uvod

Robotski manipulator ili robotska ruka je reprogramabilni i multifunkcionalni elektromehanički uređaj koji se može koristiti za pomjeranje materijala, djelova, objekata ili alata kroz set programiranih pokreta zarad izvršenja radnog zadatka. Definiciju robota-reprogramabilne i multifunkcionalne mehaničke strukture daje Internacionalna organizacija za standarde: „Robot je mašina koja se sastoji od mehanizama sa više stepeni slobode kretanja, a sposoban je da vrši manipulaciju sa alatom, radnim predmetom ili nekim drugim sredstvom.“

Izraz „robot“ potiče iz češkog jezika. Prvi put se pojavio u predstavi iz 1921. godine. Koristio se za opisivanje vještačkih ljudi proizvedenih u fabrici da služe ljudima. Izraz je izveden od riječi robota što znači „kmetovski rad“. U Evropi u osamnaestom vijeku kmet je bio osoba vezana za imanje i primorana da obrađuje zemlju prema uputstvima gospodara. Kmetovi su posmatrani kao najniža društvena klasa, u suštini robovi. Njihov stav prema poslu koji su obavljali bio su beznačajan njihovim gospodarima, pri čemu su najčešće obavljali najnepoželjnije i najteže poslove. Dakle, izraz „robot“ je poprimio značenje mašina sa ljudskim karakteristikama, koja izvršava rad neprikladan za tipičnog čovjeka – u suštini mašina u ropstvu.

Komercijalna primjena industrijskih robota sa računskim upravljanjem – kompjuterizovanih industrijskih robota počinje 70-ih godina XX vijeka. Automatizacija procesa i mašina nalazi primjenu prvenstveno kod izvođenja proizvodnih procesa i upravljanjem mašinama, a manje kod drugih takođe važnih proizvodnih aktivnosti kao što su: opsluživanje radnog mjesta, pozicioniranje radnog komada i slično.

Industrijski roboti nalaze primjenu prilikom:

- opsluživanja radne stanice,
- držanja radnog predmeta u radnom položaju u raznim fazama izrade,
- operacionog transporta,
- tehnološke operacije,
- automatske montaže i
- predprocesne, procesne i postprocesne kontrole.

Industrijski roboti su idealni za poslove koji se smatraju teškim i nepogodnim za ljude. Koriste se za poslove koji se ponavljaju više puta i kao takvi se smatraju monotoni. U onim procesima gdje se traži visok kvalitet i velika produktivnost takođe se koriste industrijski roboti. Savremena industrijska proizvodnja u većini svojih grana uspješno koristi robotske sisteme. Kada je u pitanju pokretljivost pojedinih članova robota, mogućnost izvođenja različitih putanja, sposobnost dosezanja u bilo koju tačku radnog prostora sa postizanjem određene orijentacije, može se reći da su mogućnosti primjene robota u proizvodnji praktično neograničene.

Ono što ograničava primjenu robota u pojedinim operacijama jeste pitanje ekonomičnosti. Nije rentabilno da jedna robotska struktura velikog volumena radnog prostora, velikih brzina i snage, obavlja radne zadatke za koje u potpunosti ne iskorišćava svoje sposobnosti. Iz tog su razloga se industrijski roboti dizajniraju tako da mogu obavljati određenu vrstu radnih zadataka.

Jedan od bitnih razloga primjene robotskih sistema u industriji jeste i dehumanizacija rada, pogotovo na poslovima štetnim po ljudsko zdravlje (rad u zagađenoj sredini, prašini, visokoj temperaturi, rad na monotonim i zamarajućim poslovima).

Proces dizajna industrijskog manipulacionog robota je veoma komplikovan i spor. Iz tog razloga je dio istraživanja u ovoj oblasti posvećen razvoju modularnih manipulatora za generalnu industrijsku primjenu, kao i što lakšoj i bržoj integraciji manipulatora u sajber-fizičkim proizvodnim sistemima [1, 2]. Kao što je već prethodno navedeno, da bi se upravljalo robotom, neophodno je napraviti kinematički model robota. Na osnovu kinematičkog modela se vrši proračun direktne i inverzne kinematike manipulatora [3]. Osim za samo upravljanje, koncepti direktne i inverzne kinematike su esencijalni prilikom razvoja novih koncepta dizajna manipulatora [4], kao i analize postojećih industrijskih rješenja u cilju njihovog poboljšanja [5] – [7].

Dizajn i izrada strukture robota predstavlja veoma složen proces koji zahtijeva mehaničke i dinamičke proračune, izbor pogona robota, određivanje geometrije i izrada segmenata. Pojavom 3D štampe je značajno olakšan i ubrzan proces dizajna i testiranja mehaničkih djelova robota, kao i za izradu kompletnog prototipa manipulatora [8].

Osim računara, koji predstavlja mozak robota, veoma bitnu ulogu u upravljanju robotom će imati mikrokontroler, tj. Arduino platforma. Arduino razvojna ploča predstavlja alat za brzi razvoj prototipa sistema i uređaja koji mogu vršiti interakciju sa svojom okolinom. Upravo zbog toga je Arduino mikrokontroler našao ogromnu primjenu u praktičnoj edukaciji studenata inženjerskih nauka širom svijeta [9, 10]. Raznovrsnost funkcija koje Arduino platforma može obavljati ga čini prikladnim za razvoj prototipa robotskih manipulatora. Arduino može u realnom vremenu upravljati motorima manipulatora, vršiti očitavanje pozicija zglobova i komunicirati sa računarom, dok se kinematsko modelovanje i proračuni, zbog svoje kompleksnosti, najčešće moraju vršiti na računaru. Dakle, dizajn i razvoj manipulacionog robota je napratičnije vršiti umrežavanjem računara i Arduino platforme, koji vrši testiranje predloženog koncepta upravljanja [11] – [13].

Za planiranje kretanja manipulatora, kao i samo upravljanje njime, je neophodno riješiti problem inverzne kinematike. Ovaj problem nije jednostavno riješiti kao direktni problem, jer se njegovo rješenje svodi na rješavanje sistema nelinearnih jednačina. Rješenje inverznog kinematičkog problema se može dobiti analitičkim i numeričkim putem. Pristupi rješavanja opisanog problema se najčešće baziraju na direktnom rješavanju sistema jednačina ili na algoritmima koji koriste matrice homogenih transformacija koje opisuju međusobni položaj zglobova robota [14]. Jedan od savremenih pristupa je baziran na tzv. mekom računarstvu (*soft computing approach*), koji uključuje neuralne mreže, fazi logiku i evolucione algoritme. U radu [15] je opisano rješavanje inverzne kinematike pomoću neuralnih mreža zbog njihove osobine da veoma dobro mapiraju nelinearni odnos između ulaznih i izlaznih podataka (ulaz – položaj završnog organa, izlaz – koordinate zglobova). Robotska ruka se može smatrati redundantnim manipulatorom ukoliko robot posjeduje veći broj stepeni slobode od onog koji je potreban da bi se izvršio radni zadatak. Međutim, kod redundantnih konfiguracija se može javiti veći broj rješenja za isti krajnji položaj, od kojih neka mogu imati nelinearnosti ili singularitete što inverzni kinematički problem čini komplikovanim za rješavanje. Upravo iz tog razloga se za rješavanje inverzne kinematike sve više koriste algoritmi mekog računarstva, koji tolerišu nepreciznosti i nelinearnosti prilikom rješavanja problema, kao što su genetski algoritam (eng. *GA – Genetic Algorithm*), teorija optimizacije roja (eng. *PSO – Particle Swarm Optimization*) i diferencijalna evolucija (eng. *DE – Differential Evolution*) [16] – [18].

Od robotskog manipulatora u industriji se očekuje da prati zadatu trajektoriju što je moguće bliže za veoma širok raspon kretanja u radnom prostoru, uprkos promjenama parametara sistema – greške u parametrima, različite vrste tereta i različite vrste radnih zadataka. Da bi se ostvarila preciznost u radu robota neophodno je uvesti povratnu spregu, tj. vršiti poređenje zadate i stvarne pozicije robota. Minimizacija razlike između zadate i stvarne vrijednosti pozicije zglobova se vrši pomoću upravljačkog algoritma. Algoritam upravljanja koji predstavlja regulacije industrijskih procesa i koji se takođe nerijetko primjenjuje u robotici je PID (eng. *Proportional, Integral, Derivative*) koji, nakon podešavanja koeficijenata, minimizuje grešku i samim tim omogućava praćenje referentne pozicije [19].

Senzori u robotici služe za procjenu stanja i okoline robota. Senzori u robotima su zasnovani na funkcijama ljudskih čulnih organa. Jedna od novijih inovacija u robotici je sistem vizije, koji omogućava bolju navigaciju i identifikaciju objekata u radnom prostoru robota. U radu [20] je upotrijebljen upravo sistem vizije, sa više kamera i sistemom za obradu slike, za koji je pokazano da poboljšava preciznost pozicioniranja završnog organa u radnom prostoru.

Rad je podijeljen na sljedeća poglavlja.

U drugom poglavlju je opisan robotski manipulator koji se koristi u radu. Izložen je proces izrade 3D modela, izrade segmenata, opis svih komponenti robota, proces sklapanja i način funkcionisanja manipulatora. Takođe je detaljno opisana geometrija robota, na osnovu koje se vrši kinematičko modelovanje robota.

U trećem poglavlju je izložen kinematički model robota koji se koristiti prilikom planiranja kretanja i upravljanja. Takođe su izloženi problemi direktne i inverzne kinematike, kao i objašnjena njihova primjena u upravljanju manipulatorom.

U četvrtom poglavlju je opisano planiranje kretanja robota, odnosno koncept generisanja trajektorija završnog organa i zglobova robota. Izloženi su različiti načini generisanja trajektorija, kako u prostoru unutrašnjih, tako i u prostoru spoljašnjih koordinata. Takođe su detaljno opisani principi upravljanja robotskim manipulatorom i upravljački algoritmi.

U petom poglavlju ovog rada je opisan eksperiment i izvršena analiza rezultata. Najprije je definisan radni zadatak koji manipulator treba da izvrši. Nakon toga se definišu tačke puta za završni organ, na osnovu kojih se izračunavaju trajektorije zglobova. Radni zadatak se izvršava uz korišćenje odabranih koncepta i algoritama upravljanja. Na kraju su prikazani i analizirani rezultati eksperimenta. Nakon analize rezultata se se izvode zaključci o radu sistema radi potvrde istraživačkih pitanja.

U zaključku je napravljen osvrt na cjelokupan rad, uz odgovarajuće komentare o potencijalnim unaprijeđenjima i daljem istraživanju.

Svi programski kodovi koji su izvršeni u MATLAB-u i na Arduino mikrokontroleru dati su u prilogu na kraju rada.

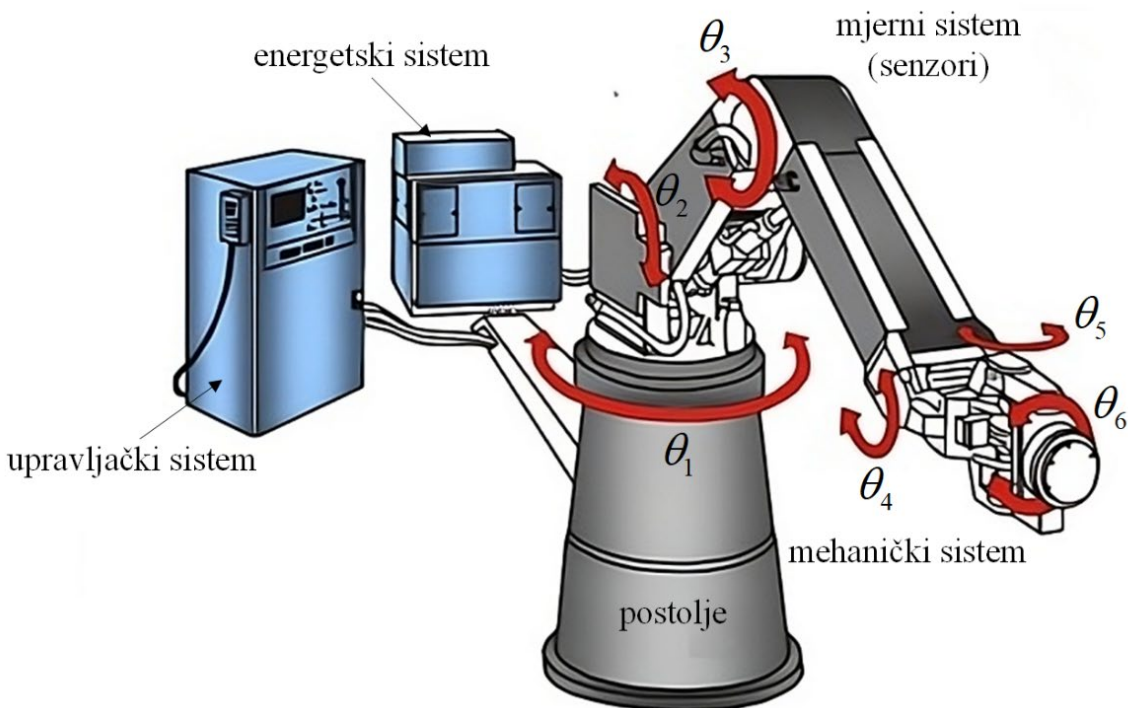
2. Robotski manipulator

U ovoj glavi je izložen pojam robotskog manipulatora kao jednog složenog elektromehaničkog sistema. Uz to je opisan proces dizajna manipulatora koji se koristi u radu i proces izrade njegovih segmenata. Takođe su dati detaljni opisi komponenti sistema, procesa sklapanja i načina funkcionisanja manipulatora.

2.1. Pojam robotskog manipulatora

Industrijski robotski manipulator je robot koji se koristi za obavljanje različitih zadataka, poput obrade i transporta materijala, sastavljanja proizvoda, pakovanja i drugih procesa u industriji. Za industrijske primjene se najčešće koristi antropomorfna mehanička ruka, nalik na ljudsku, kao na slici 2.1, koja može da donosi odluke na osnovu mjernih signala sa senzora i da vrši interakciju sa svojim okruženjem. Svaki manipulator, a samim tim i manipulator koji se koristi u ovom radu, sadrži sljedeće neophodne sisteme:

- mehanički sistem,
- energetski sistem,
- mjerni sistem,
- upravljački sistem.



Slika 2.1. Robotski manipulator [21]

2.1.1. Mehanički sistem

Mehanički sistem robotskog manipulatora sastoji se od skupa povezanih elemenata koji omogućavaju kretanje i pozicioniranje robotske ruke u radnom prostoru. Glavni elementi robotskog manipulatora su kruti segmenti i mehanički zglobovi koji ih povezuju.

Zglobovi ili kinematički parovi nastaju međusobnim povezivanjem dva ili više segmenata, od kojih je bar jedan pokretan. Kod većine industrijskih robota, zglobovi dozvoljavaju isključivo rotaciono ili linearno kretanje, pa se samim tim takvi zglobovi nazivaju *rotacioni* i *translatorni* zglobovi. Mehanizam zgloba robota se sastoji od najmanje četiri glavne komponente: strukture ose zgloba, aktuatora, prenosa i senzora stanja (najčešće pozicije, ali su česti i senzori brzine i sile). Kod prototipa manipulatora niskih performansi, koji se koristi u ovom radu, inercija i dinamika sistema nisu toliko značajne, pa samim tim aktuatori mogu biti postavljeni blizu osa zglobova, jer njihova inercija i težina neće značajno uticati na performanse robota. U slučaju robota visokih performansi, koji se mogu zateći u industrijskim postrojenjima, minimizovanje inercije sistema je veoma bitno. Iz tog razloga se kod ovih manipulatora aktuatori postavljaju blizu ose prvog zgloba, dok se zglobovi pokreću putem sistema kaiševa, kablova ili zupčanika. Međutim, dok duža distanca prenosa smanjuje gravitacioni moment i inerciju, ona istovremeno povećava fleksibilnost, odnosno redukuje krutost sistema. Dakle, pozicioniranje aktuatora i prenosa za svaki zglob predstavlja balansiranje između inercije i krutosti sistema [23].

Najbitnija mjera kvaliteta jednog zgloba je krutost ili otpor zgloba na sve neželjene kretnje. Odavde proizilazi da je poželjno da zglob ima što manje stepeni slobode. Upravo iz tog razloga se rotacioni i translatorni zglobovi smatraju *osnovnim* zglobovima, jer imaju po jedan stepen slobode kretanja.

Rotacioni zglobovi su dizajnirani da vrše isključivo rotaciono kretanje, uz minimizaciju drugih pomjeraja i kretanja. Da bi se zglob klasifikovao kao rotacioni, osa rotacije mora biti normalna na ose segmenata. Translatorni ili linearni zglob omogućava translatorno pomjeranje jednog segmenta u odnosu na drugi [21].

Segmenti ili linkovi su djelovi robotskog manipulatora koji se povezuju pomoću zglobova i zajedno sa zglobovima čine mehaničku strukturu robotske ruke. Segmenti mogu biti različitih dužina, oblika i izrađeni od raznih materijala, zavisno od funkcije koju robot treba da obavlja. Segmenti se najčešće izrađuju od metala, plastike ili nekih drugih materijala i mogu se spojiti na mnoštvo različitih načina, zavisno od dizajna i primjene robota.

Ključni element mehaničkog sistema robotskog manipulatora je pogonski (aktuatorski) sistem koji omogućava kretanje. Aktuatorski sistem može biti hidraulički, pneumatski ili električni. Najčešće se koristi električni pogonski sistem, koji sastoji se od elektromotora i mehaničkih prenosnika. Električni motori se koriste za pokretanje zglobova i završnog alata manipulatora, pri čemu se najčešće koriste koračni motori i servo motori. Svrha mehaničkog prenosnika je da prenese mehaničku snagu sa osovine motora na ose samih zglobova.

2.1.2. Energetski sistem

Energetski sistem robotskog je sistem čija je uloga snabdijevanje energijom preostalih sistema manipulatora. U zavisnosti od vrste manipulatora, mogu se koristiti različiti izvori energije kao što su električni, hidraulički, pneumatski ili njihova kombinacija.

U industriji se najčešće se koriste električni izvori energije, odnosno naizmjenični jednofazni ili trofazni izvori napajanja. Ovi izvori napajanja omogućuju konstantnu snagu i napon, što je ključno za stabilan rad pogonskog, mjernog i upravljačkog sistema. Pored toga, neki manipulatori koriste i hidraulične ili pneumatske pogonske sisteme. Hidraulični pogon koristi hidrauličnu tečnost za pokretanje zglobova, dok pneumatski pogon koristi vazduh pod pritiskom.

U slučaju manipulatora upotrebljenog u radu, koristi električni jednosmjerni (DC) izvor napajanja, iz razloga što sve električne komponente manipulatora funkcionišu na jednosmjernom naponu. Da bi se dobile željena pozicija i brzina robotske ruke, energija iz DC napajanja se mora precizno usmjeravati. Usmjeravanje energije, odnosno upravljanje se vrši pomoću upravljačke jedinice koja usmjerava energiju ka jedinicama pogonskog sistema.

2.1.3. Mjerni sistem

Mjerni sistem robotskog manipulatora je sistem senzora i uređaja koji se najčešće koriste za mjerenje položaja, brzine, ubrzanja i sila, kao i parametara koji opisuju njegovo radno okruženje. Preciznost mjernog sistema je ključna za postizanje visoke tačnosti pozicioniranja i pouzdanosti manipulatora.

Senzori koji se najčešće koriste za mjerenje položaja i brzine manipulatora su enkoderi ili potencimetri. Enkoderi su uređaji koji mjere ugaoni pomjeraj zglobova robotske ruke. Postoje inkrementalni enkoderi koji mjere relativni ugaoni pomjeraj ili apsolutni enkoderi koji mjere apsolutni ugaoni položaj. Potencijometar je promjenljivi otpornik koji funkcioniše kao djelitelj napona, promjena otpornosti odnosno napona na njegovom izlazu je proporcionalna promjeni pozicije.

Za mjerenje sile i pritiska koriste se senzori sile koji se mogu ugraditi u završne uređaje ili bazu manipulatora. Senzori sile se koriste za mjerenje sile koju manipulator primjenjuje na objekat kojim se manipuliše, a takođe se mogu koristiti za mjerenje sila koje djeluju na samu robotsku ruku.

Senzori koji se koriste za mjerenje različitih parametara radne sredine robota su senzori temperature, senzori vlage, senzori optičke detekcije i drugi.

Mjerni podaci se prikupljaju i obrađuju na računaru uz pomoć algoritama koji omogućuju precizno pozicioniranje robotske ruke, uz poštovanje propisanih mjera sigurnosti. Na taj način manipulator može precizno izvršavati radne zadatke i upravljati teškim ili opasnim teretima.

2.1.4. Upravljački sistem

Upravljački sistem robotskog manipulatora sastoji se od hardverskih i softverskih komponenti koji su potrebni za upravljanje pokretima robotske ruke.

Hardverski dio se sastoji od upravljačkog računara, pogonskih motora, upravljačkih jedinica i senzora. Upravljački računar je srce upravljačkog sistema koje prima naredbe od korisnika i obrađuje ih pomoću softverskih algoritama kako bi se upravljalo pokretima robota. Pogonski motori generišu potrebne sile za pokretanje zglobova robota, dok upravljačke jedinice regulišu brzinu, smjer i poziciju motora. Senzori prikupljaju podatke o stanju robota i šalju ih upravljačkom računaru kako bi se mogao osigurati precizan nadzor nad pokretima robota.

Softverski dio upravljačkog sistema uključuje algoritme za planiranje kretanja, upravljačke algoritme, praćenje pokreta robota, kao i algoritme za izbjegavanje sudara i testiranje sigurnosti u toku obavljanja radnog zadatka. Algoritam za planiranje pokreta omogućava planiranje putanje robota u zavisnosti od radnog zadatka, dok algoritmi upravljanja i praćenje pokreta osiguravaju precizno kretanje robota u realnom vremenu. Algoritmi za izbjegavanje sudara i testiranje sigurnosti osiguravaju da se robot kreće na siguran način, bez opasnosti po okolinu i ljude.

Upravljački sistem robotskog manipulatora može biti programiran na različite načine, zavisno od potreba korisnika. Najčešće se za razvoj upravljačkog softvera koriste programski jezici kao što su C++ ili Python, dok se u drugim slučajevima mogu koristiti vizuelni programski alati poput LabVIEW-a.

2.2. Dizajn robotskog manipulatora

Djelovi fizičke strukture kao što segmenti, osovine, prenosnici i ležajevi robota koji stvaraju njegov pokretni skelet naziva se mehanička struktura ili mehanizam robota. U ovom poglavlju je opisan proces dizajna mehanizma manipulatora koji se koristi u radu, izbor aktuatora (servo motora) i upravljačkog sistema i njihovo povezivanje u elektromehanički sistem koji transformiše kompjuterske komande u kretanje.

Robot se dizajnira sa određenim skupom zadataka na umu. Dizajn robota se fokusira na broj zglobova, fizičku veličinu, nosivost i zahtjeve za kretanje završnog organa. Konfiguracija mehanizma i ukupna veličina robota su određeni zahtjevima zadataka za doseg, radni prostor i sposobnost reorijentacije. Ove karakteristike utiču na preciznost upravljanja putanjom hvataljke, kao i sposobnost pozicioniranja u prostoru, ponovljivost pokreta i finu rezoluciju koja omogućava precizne pokrete u realnom vremenu zasnovane na informacijama dobijenih sa senzora. Kritična stavka u dizajnu robotskog sistema je broj zadataka za koje je robot namijenjen. Robot treba da bude dizajniran tako da ima fleksibilnost koja mu je potrebna da obavlja niz zadataka za koje je namenjen. Informacija o broju i vrsti zadataka za koje je robot namijenjen određuje topologiju mehanizma robota i aktuatorski sistem.

2.2.1. Dizajn i izrada mehaničke strukture manipulatora

Mehanička struktura industrijskog robota se može podijeliti na: postolje robota, mehanizam robota i završni organ, koji može da bude u vidu hvataljke, senzora ili alata.

Postolje robota je slično postolju alatnog uređaja i može biti izvedeno na više različitih načina. Kod izbora načina izvođenja postolja bitnu ulogu ima namjena robota i specifičnost zadataka koje robot treba da obavlja.

Mehanizam robotskog manipulatora predstavlja sistem zglobno povezanih segmenata, čija je namjena dovodenje završnog organa u zadati položaj.

Da bi se moglo dalje govoriti o dizajnu i izradi mehaničke strukture robota, najprije je potrebno reći nešto više o procesima 3D modelovanja i 3D štampe.

U 3D računarskoj grafici, 3D modelovanje je process kreiranja matematičke reprezentacije bilo koje površine ili objekta u tri dimenzije korišćenjem specijalizovanih softvera putem manipulacija ivicama, tačkama i poligonima u simuliranom trodimenzionalnom prostoru. Ono što nastane se naziva 3D modelom, koji je u svojoj srži kolekcija podataka o tačkama u trodimenzionalnom prostoru koje su povezane raznim geometrijskim oblicima (linijama, trouglovima, zakrivljenim površinama, itd.).

Trodimenziionalna ili 3D štampa je vrsta aditivne proizvodnje koja se koristi za pravljenje trodimenzionalnog objekta na osnovu 3D modela. Aditivna proizvodnja podrazumijeva slaganje i spajanje slojeva materijala. Veoma je brza, jeftina i može da kreira značajno kompleksnije geometrije u odnosu na tradicionalne tehnologije. Postoje različiti tipovi 3D štampe, pri čemu je najčešće korišćeni FDM – *Fused Deposition Modeling*. FDM je proces koji koristi filament od termoplastike, koji se konstatno dovodi do glave ekstrudera 3D štampača, gdje se isti zagrijava, a zatim gura kroz mlaznicu i nanosi na podlogu u slojevima. Glava štampača je upravljana od strane upravljačke jedinice kako bi se dobio željeni oblik.

Upravo su opisani procesi 3D modelovanja i 3D štampe esencijalni u dizajnu i izradi segmenata robotskog manipulatora koji se koristi u radu, čiji je naziv SPT – 3000. U nastavku poglavlja će biti opisan proces dizajna SPT – 3000 manipulacionog robota sa četiri stepena slobode.

Prvi korak je izrada 3D modela manipulatora. Segmenti robota se dizajniraju u alatu za 3D modelovanje Solid Edge. Osnovno pravilo prilikom dizajna segmenata je da se odabrani servo motori oblikom i dimenzijama uklapaju u tijela segmenata. Nakon uklapanja servo motora sa segmentima, posvećena je i pažnja međusobnom uklapanju segmenata. Dakle, oblik svakog segmenta mora biti takav da se:

- njemu pridruženi servo motor savršeno uklapa sa njim,
- servo motor može pravilno povezati sa narednim segmentom,
- segmenti mogu neometano kretati jedni u odnosu na druge.

Slika 2.2 prikazuje 3D model, odnosno 3D sklop SPT – 3000 manipulatora u programskom paketu Solid Edge. Na slici su označeni svi zglobovi robota. Industrijska konvencija nalaže da se zglobovi identifikuju počevši od baze robota, preko tijela robota do završnog organa.

Zglob baze ili bazni zglob povezuje segment ramena sa dijelom koji se može nazvati bazni segment. Trodimenzionalni model baznog segmenta je prikazan na slici 2.4. Bazni segment je fiksiran i povezuje konstrukciju robota sa radnom platformom. Unutar baznog segmenta se nalazi

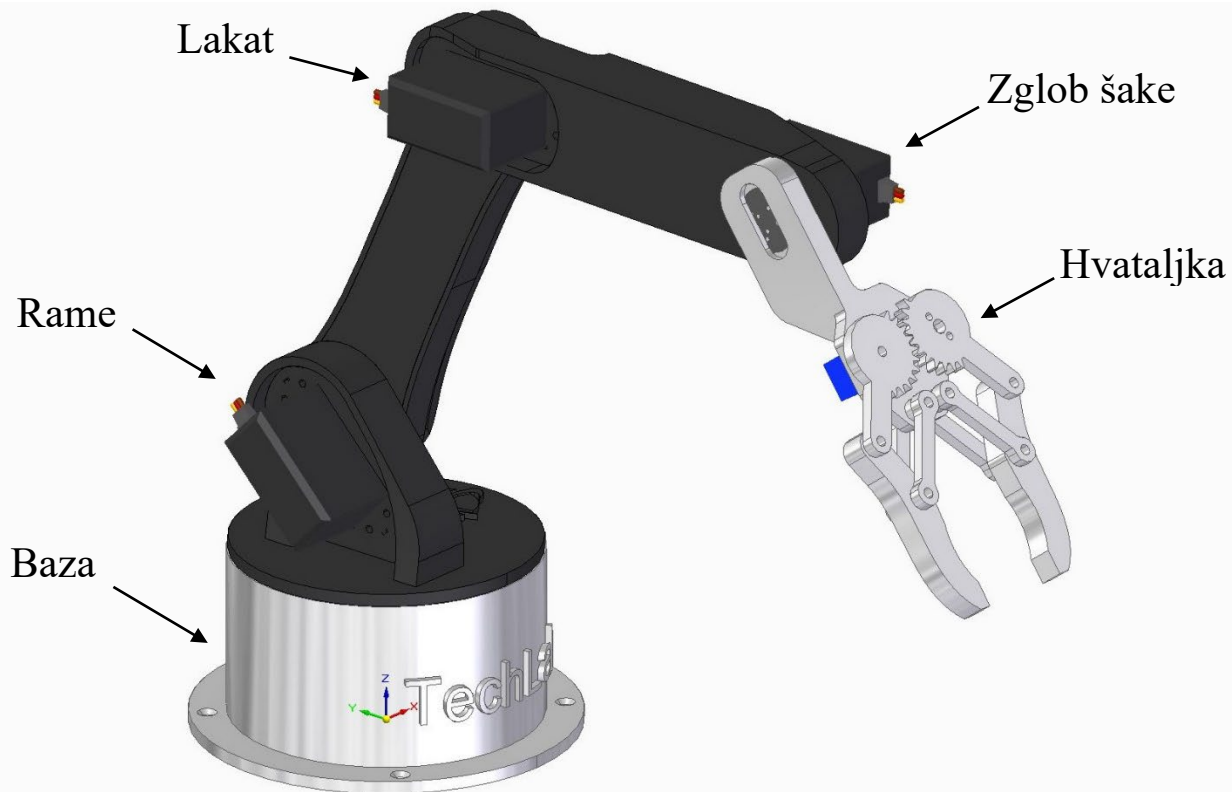
servo motor koji vrši rotaciju ramenog segmenta oko ose rotacije. Osa rotacije baznog zgloba je normalna u odnosu na radnu platformu manipulatora.

Sljedeći zglob SPT – 3000 manipulatora je zglob ramena. Rameni segment je prikazan na slici 2.5. Unutar njega se nalazi ležište servo motora koji pogoni zglob ramena. Rameni segment se u mehaničkoj strukturi manipulatora nalazi između baznog segmenta i segmenta koji se može nazvati nadlaktica i ujedno predstavlja ležište servo motora.

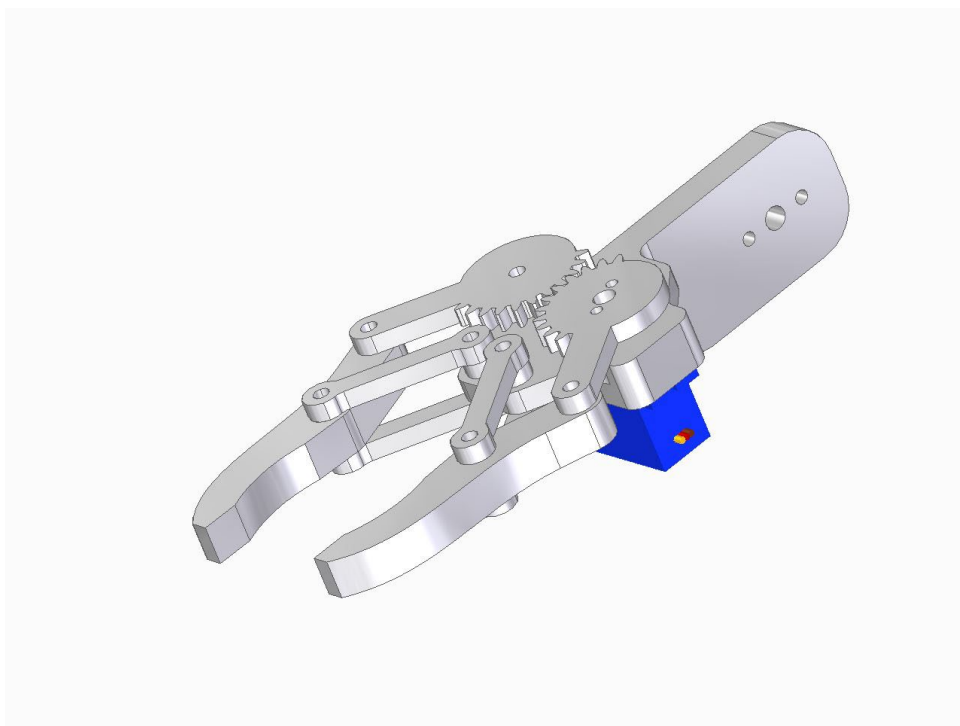
Zglob lakta povezuje dva segmenta – segment nadlaktice i segment podlaktice. Trodimenzionalni model segmenta nadlaktice je dat na slici 2.6. Segment nadlaktice je dizajniran za povezivanje sa dva servo motora – motora zgloba ramena i zgloba lakta. U samom tijelu nadlaktice je takođe napravljen procjep koji služi za lakše vođenje kablova robota. Poslednji segment manipulatora je podlaktica, čiji je 3D model prikazan na slici 2.7. Na segment podlaktice se nadovezuje završni uređaj i u njemu se nalaze kućišta za servo pogone zgloba lakta i zgloba šake.

Na mehanički sklop sastavljen od servo motora i segmenata se nadovezuje završni uređaj, odnosno hvataljka. Orijehtacija hvataljke se vrši pomoću samo jednog rotacionog zgloba, tj. zgloba šake.

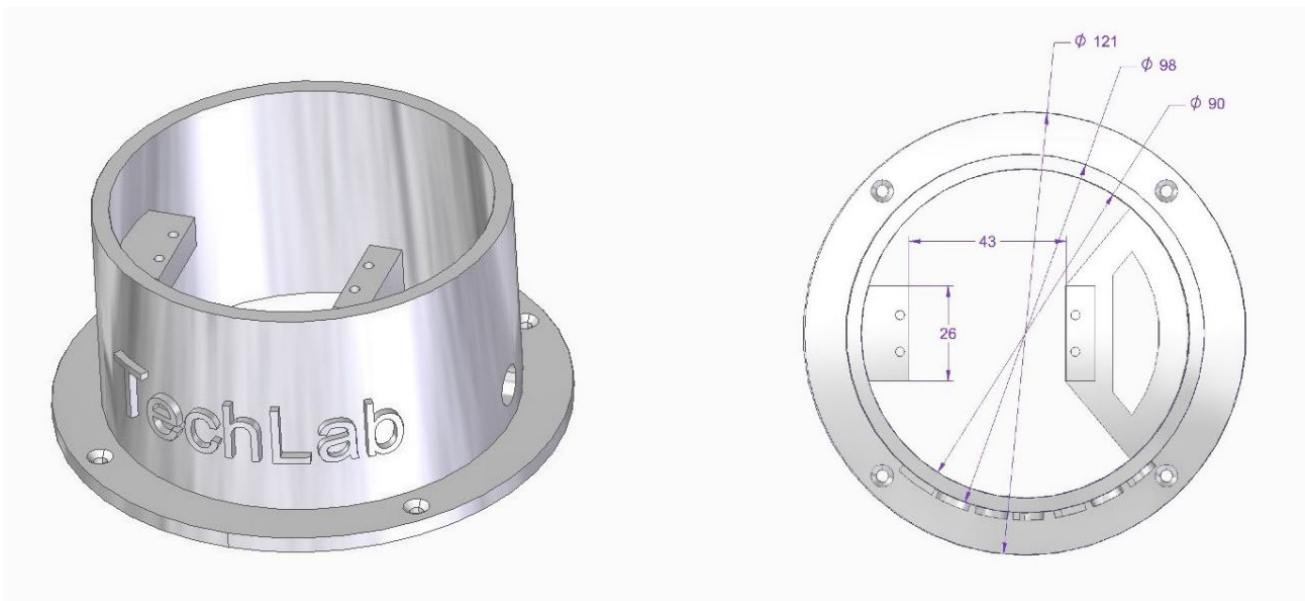
Završni uređaj - hvataljka je samostalna jedinica koja se nadovezuje na korijen šake i u direktnoj je vezi između predmeta nad kojim se vrši odgovarajuća operacija i kinematičke strukture samog robota. Namjena hvataljke je prenošenje objekta iz početne u krajnju poziciju i reorijentacija istog u radnom prostoru. Na slici 2.3 je prikazan 3D model hvataljke robotskog manipulatora. Hvataljka ima oblik kliješta, pri čemu se njen mehanizam pogoni korišćenjem mikro servo motora. Prikazana hvataljke se jedino može koristiti za podizanje lakih predmeta pravilnog oblika, upravo zbog male snage aktuatora koji je pogoni.



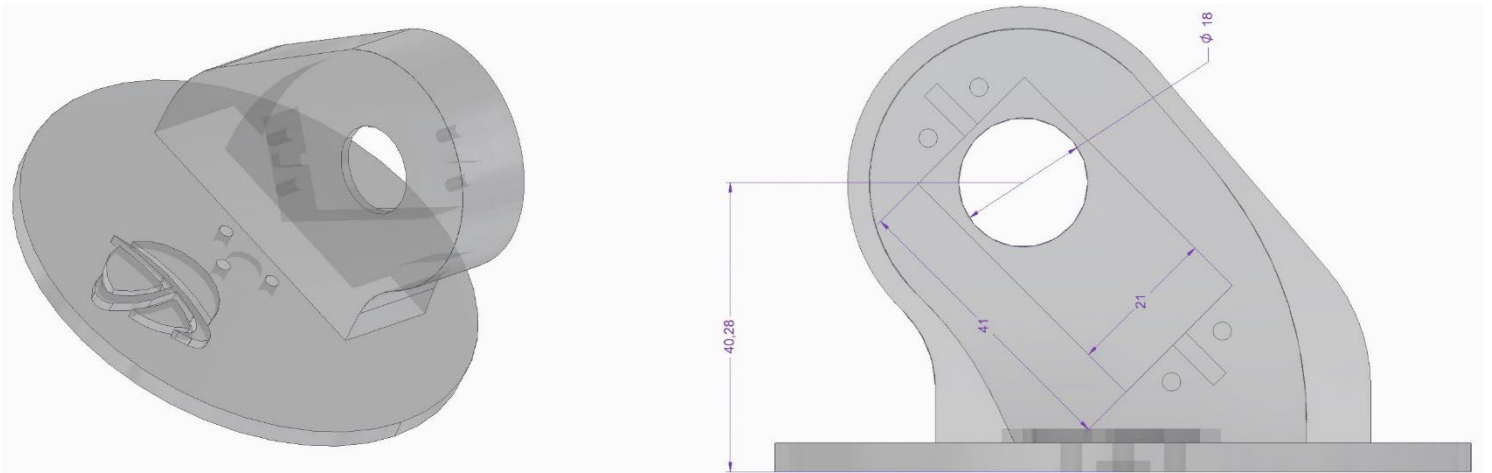
Slika 2.2. 3D sklop SPT – 3000 robotskog manipulatora



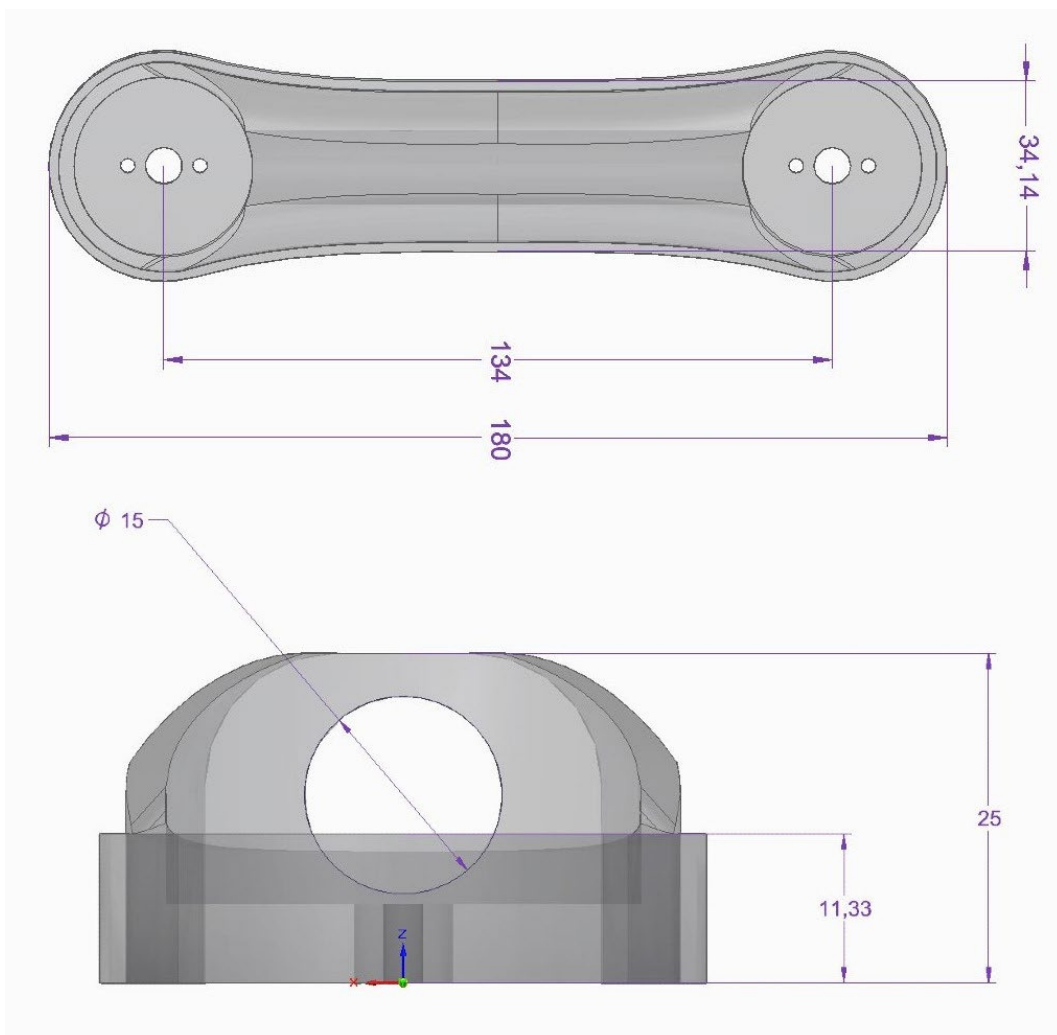
Slika 2.3. 3D model hvataljke



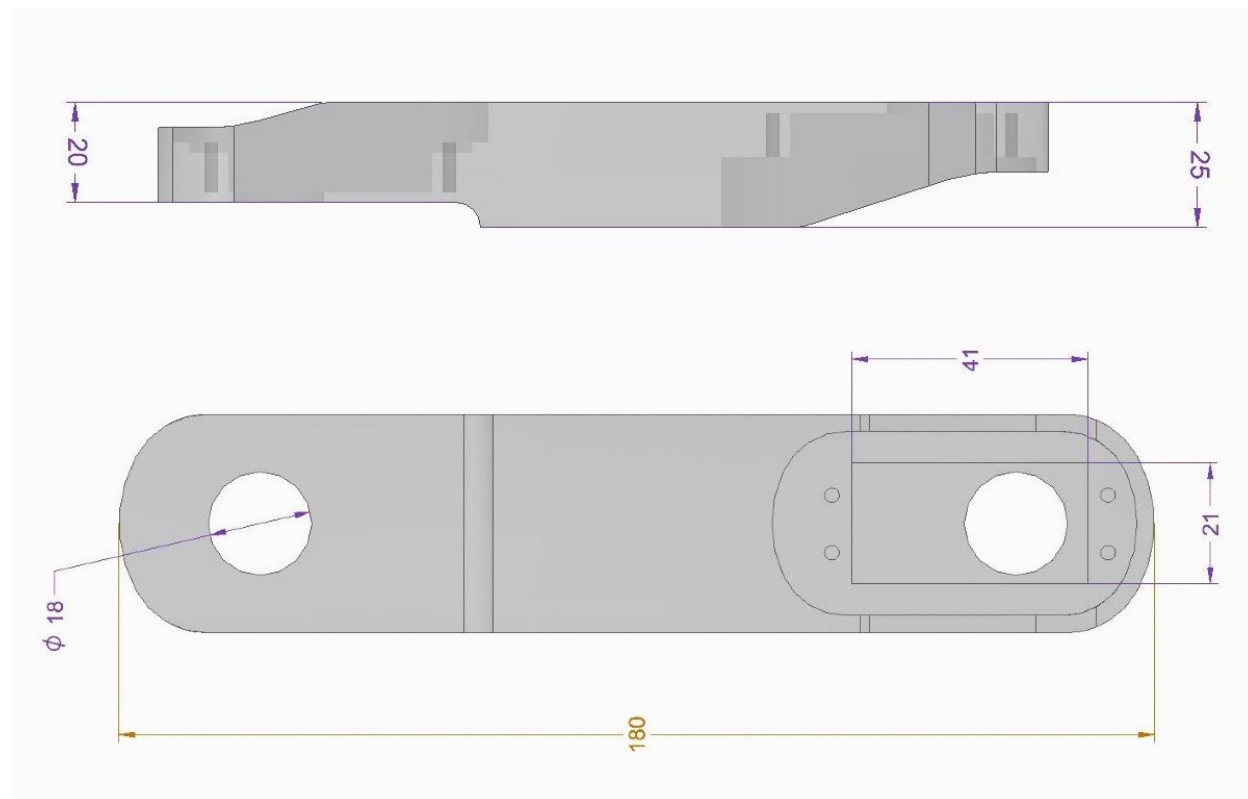
Slika 2.4. 3D model baznog segmenta



Slika 2.5. 3D model ramenog segmenta



Slika 2.6. 3D model segmenta nadlaktice

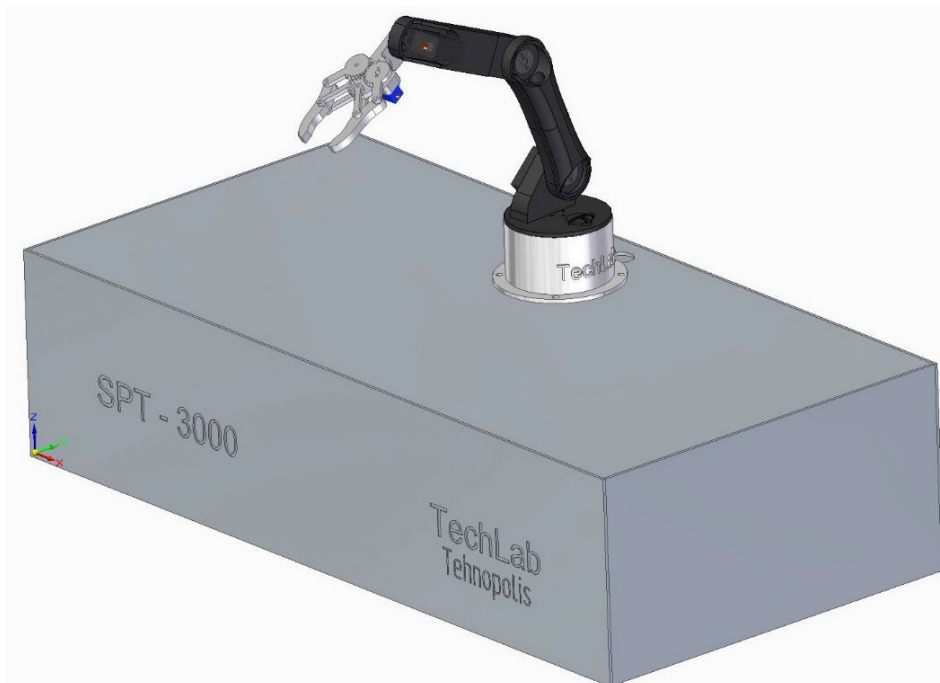


Slika 2.7. 3D model segmenta podlaktice

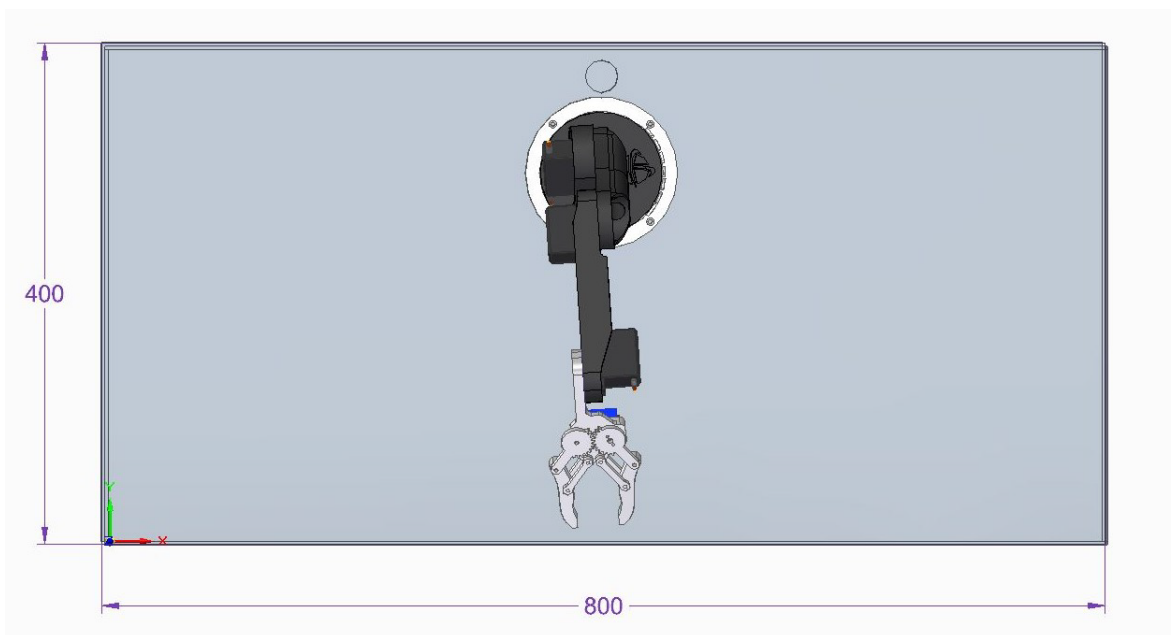
Kao što je već navedeno, u mehanički dio robotskog manipulatora spada i postolje robota. Za postolje robota su vezani ostali djelovi mehaničke strukture robota - izvršni mehanizam i hvataljka. Kako je fokus rada na ispitivanju principa kinematike antropomorfnog, servo-upravljanog manipulacionog robota, odlučeno je da postolje robota bude nepokretno, ali i prenosivo, kako bi se cjelokupna struktura robota mogla naći u bilo kom laboratorijskom okruženju.

Postolje robota je napravljeno tako da, osim što je na njemu smješten manipulator, se u njemu sadrži i cjelokupni elektronski dio manipulatora – mikrokontroler zajedno sa razvodnom pločom i kablovima koji vode do servo motora. Trodimenzionalni model postolja robota, zajedno sa robotom, je prikazan na slici 2.8.

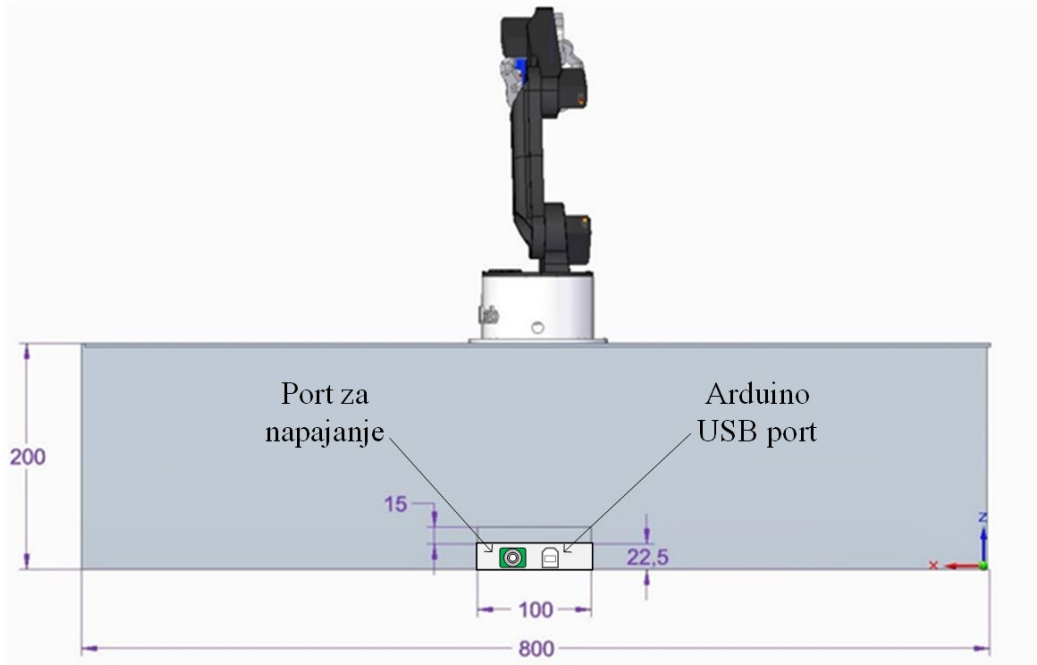
Na gornjoj strani postolja se nalaze rupe za montažu manipulatora, kao i rupa za uvođenje kablova servo motora unutar postolja (slika 2.9). Sa zadnje strane postolja se nalazi procjep koji je napravljen radi priključenja na napajanje i povezivanja sa računarom (slika 2.10).



Slika 2.8. Prikaz robotskog manipulatora sa postoljem



Slika 2.9. Gornja strana postolja



Slika 2.10. Zadnja strana postolja

2.3. Elementi upravljačkog sistema

2.3.1. Hardver

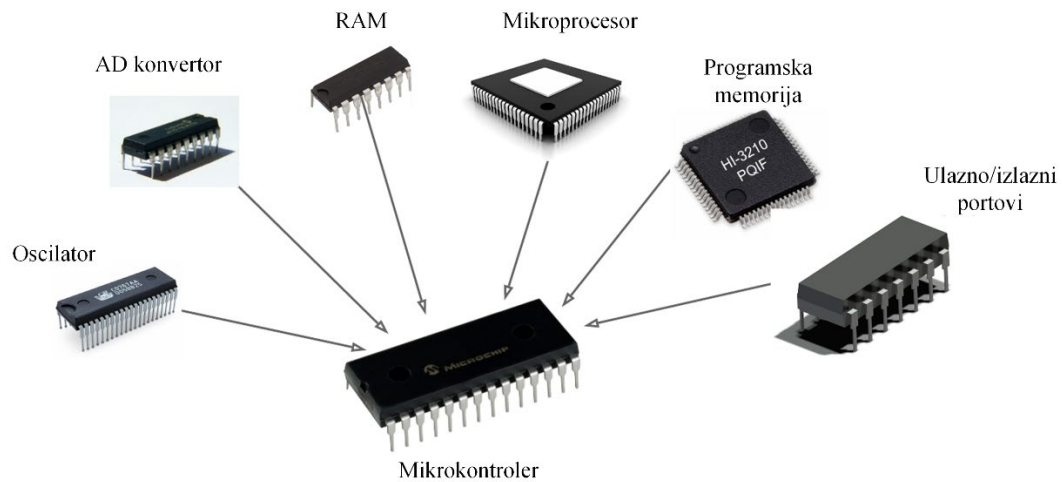
Da bi se robotski manipulator mogao koristiti za izvršavanje određenog skupa radnih zadataka, neophodno je temeljno poznavati karakteristike elemenata istog. U računarskoj tehnici termin hardver znači vidljivi, odnosno opipljivi dio računara. U slučaju robotskog manipulatora, koji predstavlja složeni elektromehanički sistem, hardverski elementi su:

1. Arduino mikrokontroler, koji sam po sebi predstavlja mikroracunar,
2. Servo motori, koji predstavljaju kombinaciju jednosmjernih motora, senzora pozicije i upravljačke elektronike,
3. Racunar, koji vrši kompleksne kinematičke proračune i komunicira sa Arduino mikrokontrolerom putem USB kabla, korišćenjem asinhronne serijske komunikacije.

2.3.1.1. Arduino mikrokontroler

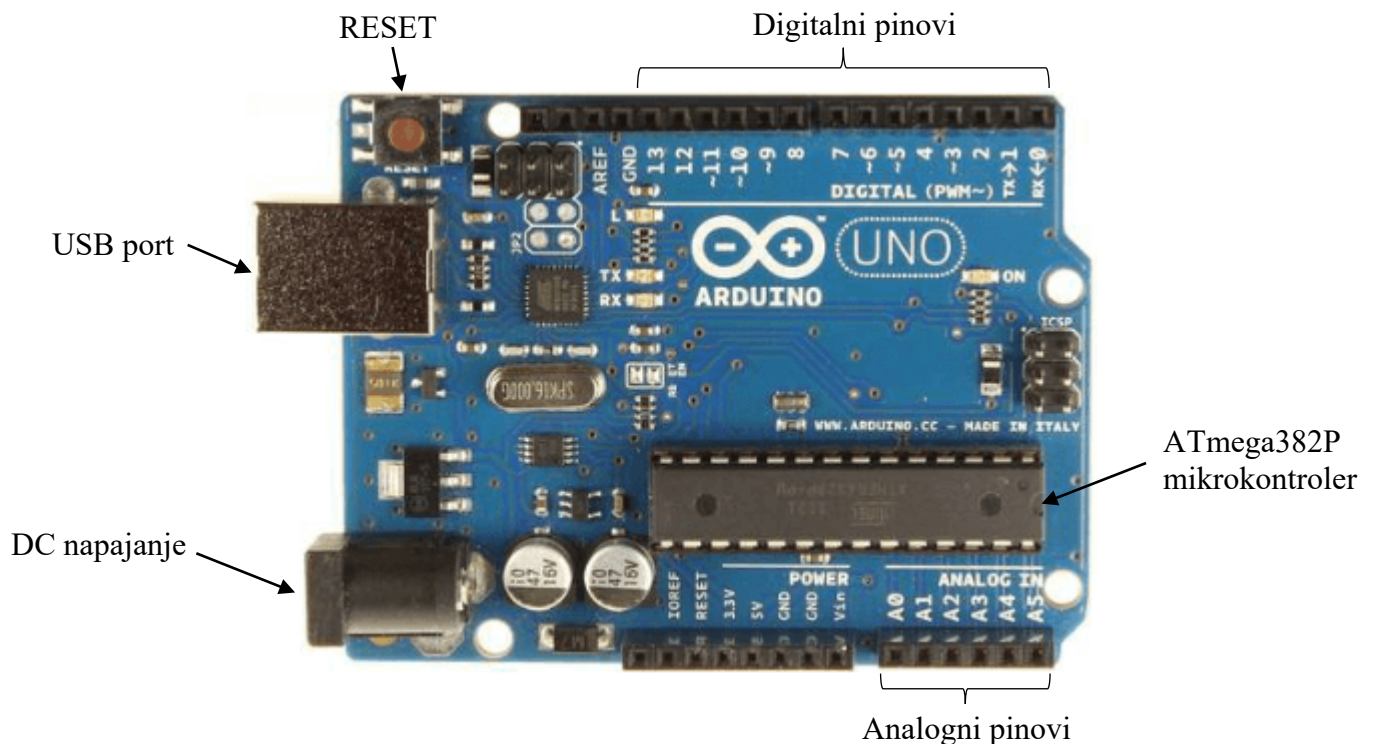
Mikrokontroler predstavlja kompaktno integrisano kolo dizajnirano da upravlja određenim procesom u ugrađenom sistemu (eng. *embedded system*). Tipični mikrokontroler uključuje procesor, memoriju i ulazne/izlazne periferije i to na jednom čipu. Mikrokontroler ili ugrađeni računar ili mikrokontrolerska jedinica se može naći u vozilima, kancelarijskim mašinama, medicinskim uređajima, kućnim aparatima ili , kao što je u slučaju ovog master rada, u robotima. Oni u suštini predstavljaju mini računare dizajnirane za kontrolu pojedinačnih funkcija u velikim, složenim sistemima, bez upotrebe kompleksnog operativnog sistema. Elementi mikrokontrolera su (slika 2.11):

1. Mikroprocesor (eng. *CPU – Central Processing Unit*) – može se posmatrati kao mozak sistema. Njegova uloga je vršenje osnovnih aritmetičkih, logičkih i ulaz/izlaz operacija na osnovu instrukcija, odnosno programa koji upravlja radom mikrokontrolera.
2. Memorija – koristi se za skladištenje podataka koje mikroprocesor prima i koristi za odgovaranje na instrukcije programa samog kontrolera. Prvi tip memorije je programska memorija, koja skladišti dugoročne informacije o instrukcijama koje izvršava CPU. Drugi tip memorije je radna memorija, koja je potrebna za privremeno skladištenje podataka u toku izvršavanja instrukcija.
3. Ulazno/izlazni portovi – ostvaruju komunikaciju procesora sa spoljašnjim svijetom. Ulazni portovi primaju informacije i šalju je procesoru u binarnoj formi. Procesor prima i obrađuje te informacije i šalje neophodne instrukcije putem izlaznih portova do spoljašnjih uređaja koji vrše zadatke izvan mikrokontrolera.



Slika 2.11. Mikrokontroler

Arduino Uno mikrokontrolerska ploča je bazirana na ATmega328P mikrokontrolerskom čipu. Ploča je opremljena setom digitalnih i analognih I/O pinova koji se mogu povezati sa spoljašnjim uređajima. Arduino ploča sadrži 14 digitalnih I/O pinova, od kojih je šest sposobno za širinsko-impulsnu modulaciju (eng. *PWM – Pulse Width Modulation*), 6 analognih I/O pinova i može se programirati u Arduino razvojnom okruženju (*Arduino IDE*), putem USB konekcije sa računarom. Ploča se može napajati putem USB kabela ili DC konektora koji dozvoljava napon u opsegu 7 – 20V. Prikaz Arduino ploče, sa označenim komponentama, je dat na slici 2.12.



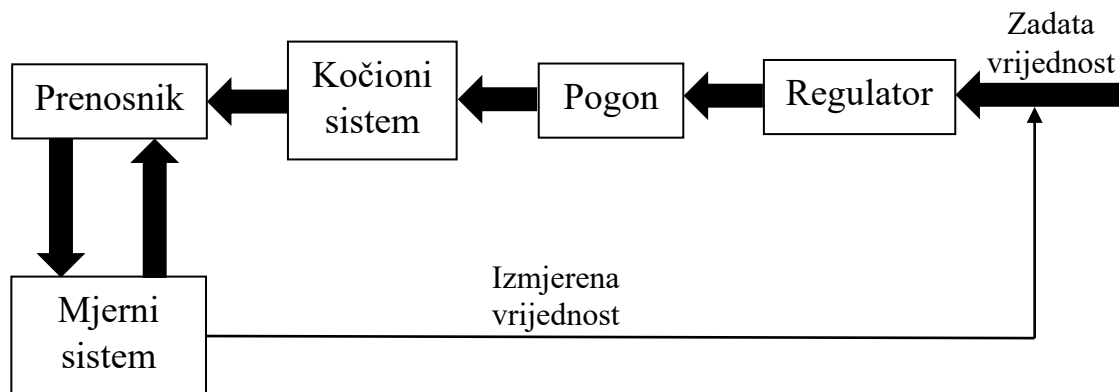
Slika 2.12. Arduino mikrokontrolerska ploča

2.3.1.2. Servo motori

Motori koji omogućavaju kretanje u zglobovima robota nazivaju se pogoni ili aktuatori. Pogoni robota imaju zadatak da prate zadatu trajektoriju, te pozicioniraju i orijentišu završni uređaj. Najvažniji zahtjevi koje moraju zadovoljavati pogoni su:

- mala težina i zapremina motora,
- dovoljno veliki obrtni momenti,
- dovoljno veliko područje ugaonog zaokreta,
- visoka tačnost pozicioniranja,
- visoki stepen pouzdanosti, itd.

Šematski prikaz pogonskog sistema dat je na slici 2.13.

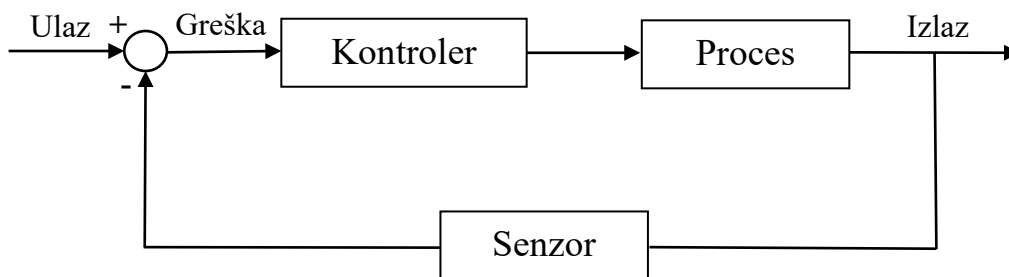


Slika 2.13. Šematski prikaz pogonskog sistema

Električni pogon kod robota je prvi put uvela švedska firma ASEA 1974. godine. Od tada se elektromotori sve češće koriste kao pogoni robota. Prednost korišćenja elektromotora je sveopšta prisutnost električne energije, jednostavnost priključaka, upravljanje koje je jednostavno, precizno i pouzdano. Glavni nedostatak ovih motora je nepovoljan odnos snage i težine, jer je najčešće neophodna primjena određenih magnetskih materijala koji značajno povećavaju težinu motora. Može se koristiti bilo koji tip motora, ali danas se u robotima koriste sljedeći tipovi motora:

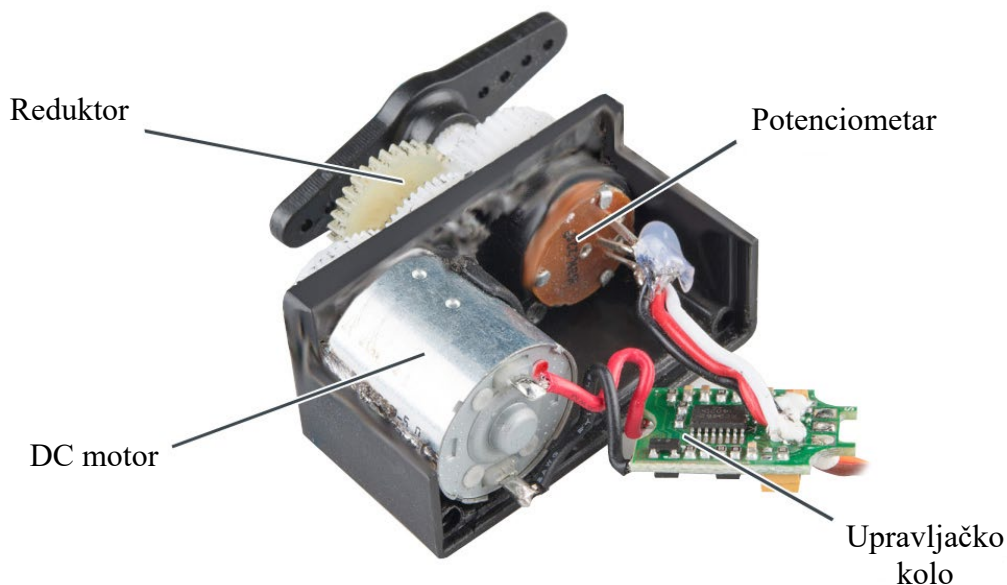
- jednosmjerni (DC) motori,
- naizmjenični (AC) motori i
- koračni ili step motori.

Konkretno, kod SPT – 3000 manipulatora upotrijebljeni su jednosmjerni servo motori. Servo motor je rotacioni aktuator koji omogućava precizno upravljanje ugaonom brzinom ili pozicijom. Sastoji se od motora, seta zupčanika, upravljačke elektronike i senzora koji daje povratnu informaciju o poziciji/brzini. Upravljačka elektronika koristi informaciju sa senzora za precizno upravljanje brzinom/pozicijom motora, tj. vrši upravljanje na principu povratne sprege (slika 2.14). Set zupčanika ili reduktor takođe igra bitnu ulogu – povećava moment i snižava brzinu na izlaznoj osovini motora. Odnosno, reduktor smanjuje brzinu kojom se pomjera teret na osovini motora, dok istovremeno povećava moment, odnosno povećava silu kojom se djeluje na teret.

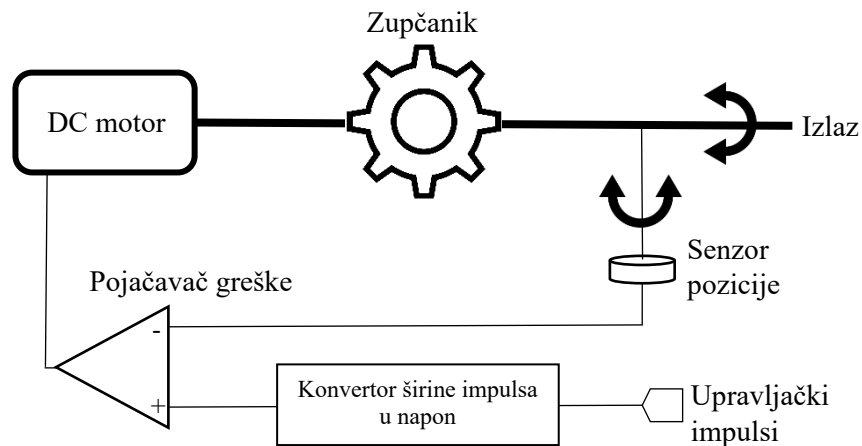


Slika 2.14. Princip povratne sprege

Za potrebe ovog rada se koriste jednosmjerni, hobi servo motori malih snaga zbog jednostavnog načina upravljanja, malih gabarita i pristupačnih cijena. Četiri osnovne komponente hobi servo motora su: DC motor, reduktor, potencijometar i upravljačko kolo (slika 2.15). DC motor na svom izlazu daje visoku brzinu i mali moment, ali se upotrebom reduktora brzina motora smanjuje na 60 obrtaja u minutu, dok se istovremeno povećava obrtni moment. Potencijometar je zakačen na poslednji zupčanik reduktora, odnosno izlaznu osovinu servo motora. Sa rotacijom, odnosno promjenom ugla na izlazu servo motora se rotira i osovinu potencijometra, pri čemu se na izlazu potencijometar stvara napon proporcionalan apsolutnom uglu izlazne osovine. U kontrolnom kolu se izlazni napon potencijometra poredi sa kontrolnim (referentnim) naponom. Ukoliko postoji razlika između izlaznog napona potencijometra i referentnog napona, dolazi do generisanja upravljačkog signala koji omogućava odgovarajuću rotaciju motora kako bi se razlika napona, odnosno greška pozicije svela na nulu. Grafički prikaz principa rada servo motora je dat na slici 2.16.

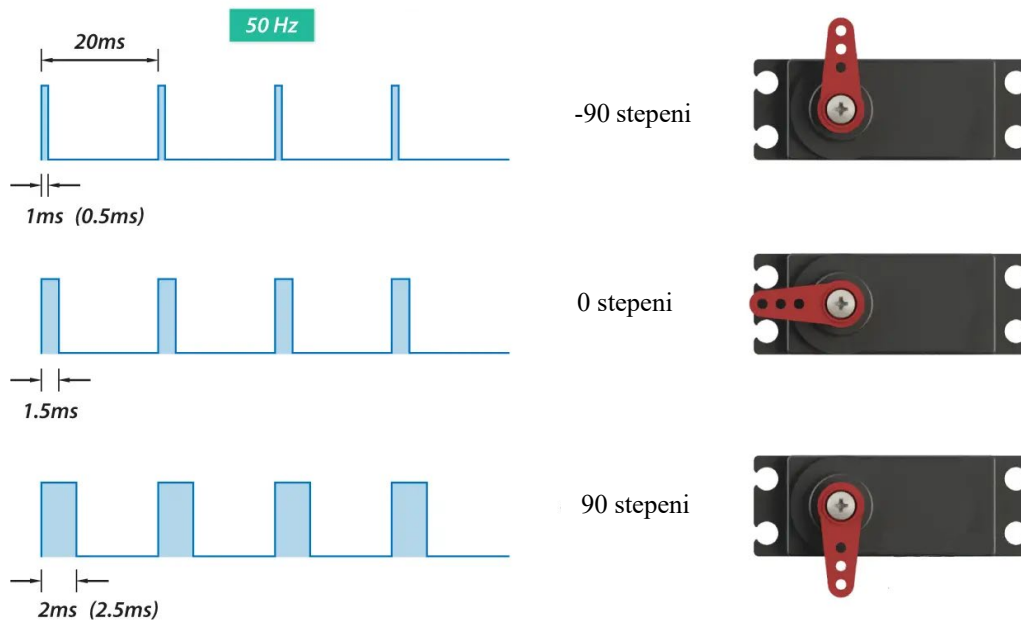


Slika 2.15. Unutrašnjost hobi servo motora



Slika 2.16. Princip rada servo motora

Servo motorom se upravlja slanjem serije impulsa putem signalne linije, odnosno putem širinsko-impulsne modulacije. Frekvencija upravljačkog signala je 50Hz , odnosno pulsevi se dešavaju na svako 20ms . Širina impulsa određuje ugaonu poziciju servo motora. Zavisno od tipa motora, opseg širine impulsa može varirati. Na slici 2.17 je dat primjer korelacije između širine impulsa i ugla zaokreta servo motora.



Slika 2.17. Upravljački signal servo motora

Modeli servo motora koji se koriste za izradu SPT – 3000 su: *SG90 Micro Servo*, *Rev Robotics Smart Robot Servo* i *Miuzei Digital Servo*. Isti su prikazani na slici 2.18, dok su njihove karakteristike date u tabeli 2.1.



Slika 2.18. Servo motori SPT – 3000 manipulatora

Servo motori	Moment kratkog spoja	Radni napon	Struja u praznom hodu	Struja kratkog spoja	Maks. brzina	Opseg širine upr. impulsa	Težina
SG90	1,2kg·cm (4,8V) 1,6kg·cm (6V)	3,5 – 6V	~100mA	650mA	60°/0,12s	500 – 2500 μs (-90 – 90°)	9g
Smart Robot Servo	13,5kg·cm (6V)	4,8 – 7,4V	~250mA	2000mA (6V)	60°/0,13s	500 – 2500 μs (-135 – 135°)	71g
Miuzei Digital Servo	20,5kg·cm (4,8V) 22,8kg·cm (6,8V)	4,8 – 6,8V	~300mA	2100mA (4,8V) 2700mA (6,8V)	60°/0,18s (4,8V) 60°/0,14s (6,8V)	500 – 2500 μs (-135 – 135°)	82g

Tabela 2.1. Karakteristike servo motora

2.3.2 Softver

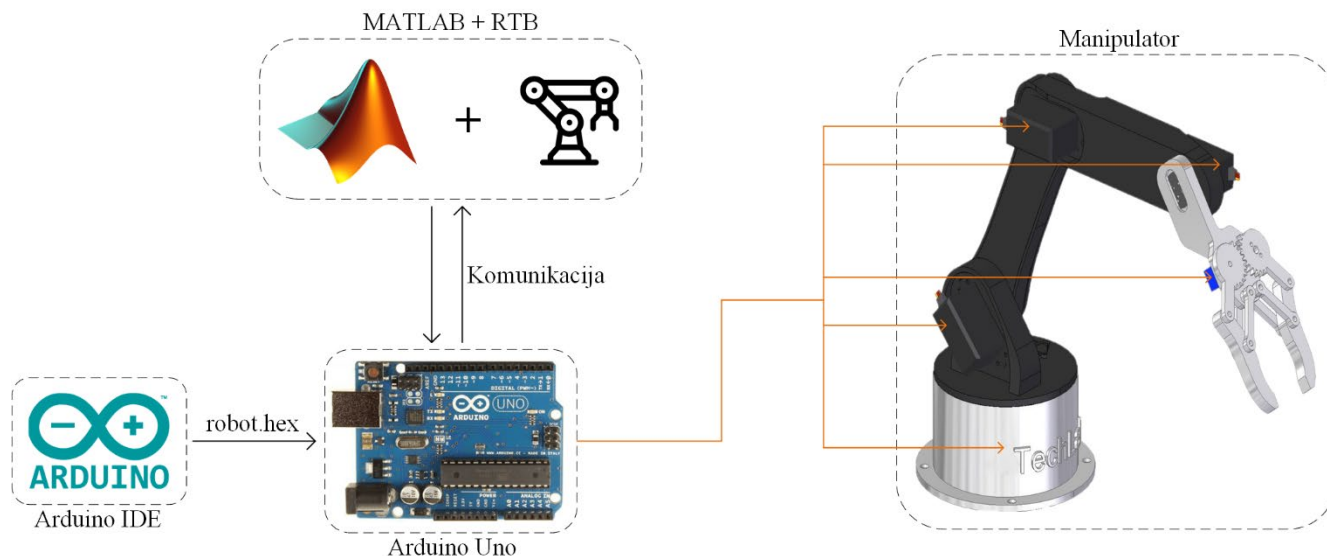
Imajući u vidu složenost zadataka koje robot obavlja (farbanje, zavarivanje, montaža dijelova i slično), kao i potrebu brzog prilagođavanja novim radnim zadacima, osnovni preduslov za korišćenje robota u industrijskim proizvodnim sistemima je mogućnost njegovog reprogramiranja.

Da bi robot obavio radni zadatak, on mora biti programiran na određeni način. Programom robota se najčešće definiše trajektorija koju slijedi završni organ robota i sve radnje robota koje omogućavaju obavljanje radnog zadatka.

U slučaju manipulatora koji se koristi u radu, vrši se programiranje Arduino mikrokontrolera, kao i pisanje izvršnog fajla u MATLAB-u. Dakle, može se zaključiti da se robotom upravlja pomoću cjeline koju čine dvije upravljačke jedinice, koje se zasebno programiraju (slika 2.19).

Arduino mikrokontrolerska ploča upravlja pogonskim sistemom, odnosno servo motorima robota. Izlazni pinovi mikrokontrolera generišu upravljačke signale. Svaki upravljački signal predstavlja povorku impulsa čija je širina direktno proporcionalna uglu rotacije servo motora, odnosno zgloba manipulatora. Manipulator sadrži ukupno pet servo motora (četiri zgloba i hvataljka), pri čemu svaki od njih mora da prati svoju zadatu trajektoriju. Informacije o trajektorije motora, tj. zglobova mikrokontroler dobija od računara sa kojim je povezan putem USB kabla. Arduino izvršava instrukcije dobijene od računara, tj. vrši računanje pozicija i upravljanje servo motorima. Iste se takođe prosljeđuju računaru gdje se vrši njihov prikaz i analiza performansi robota. Programiranje Arduino mikrokontrolera se vrši u Arduino razvojnom okruženju (*Arduino IDE*) u programskom jeziku Arduino C, koji je baziran na C++ programskom jeziku.

Kinematički model manipulatora i planiranje kretanja su implementirani na računaru u programskom paketu MATLAB, uz pomoć MATLAB Robotic Toolbox-a. Program napisan u MATLAB-u ima niz funkcionalnosti, koje će biti detaljno izložene u glavi posvećenj eksperimentu.



Slika 2.19. Programiranje robotskog manipulatora

2.4. Izbor aktuatora robotskog manipulatora

Robotska ruka se sastoji od segmenata povezanih zglobovima, pri čemu svaki zglob sadrži motor i set zupčanika (reduktor). Svaki mogući pokret manipulatora je određen od strane aktuatora u zglobovima, što znači da su performanse određene momentom koji motor, odnosno sami zglob generiše. Dakle, da bi manipulator pratio zadatu trajektoriju i izvršio radni zadatak, neophodno je adekvatno odabrati motore u zglobovima. Da bi se odabrali aktuatori, potrebno je odrediti opterećenja, odnosno momente u zglobovima manipulatora. Bitno je napomenuti da se odabir motora u ovom radu bazira isključivo na slučaju gdje robot održava teret (nije u pokretu), kada motori trpe isključivo opterećenja koja su posljedica težina samih djelova manipulatora i predmeta koji se prenosi. Prilikom kretanja manipulatora, zavisno od generisanih trajektorija i težine predmeta, na zglobove manipulatora, osim gravitacione, djeluju i dodatne sile koje su posljedica ubrzanja zglobova i samog opterećenja. Razlog za zanemarivanje sila koje nastaju prilikom kretanja robota je taj što je sami fokus rada kinematika robotskih manipulatora, pri čemu su gabariti samog manipulatora mali u odnosu na industrijske manipulatore, pa samim tim opterećenja koja nastaju kretanjem manipulatora se neće značajno razlikovati od opterećenja u mirovanju. Ipak, značajno je dati osnovni uvid u dinamiku manipulatora kao jedan od osnovnih kriterijuma prilikom dizajna bilo kog industrijskog manipulatora.

2.4.1. Moment kao fizička veličina

Moment sile ili obrtni moment je veličina u mehanici rotacionog kretanja koja je analogna ulozi sile kod translacionog kretanja. Obrtni moment, pri dejstvu sile, izaziva rotaciono kretanje tijela. Intenzitet momenta sile je jednak proizvodu intenziteta sile i normalne udaljenosti linije djelovanja sile od ose rotacije. Vektor obrtnog momenta se dobija kao vektorski proizvod vektora udaljenosti i vektora sile. Vektor momenta sile i njegov intenzitet se mogu izraziti sljedećim relacijama:

$$\vec{M} = \vec{r} \times \vec{F}, \quad (2.1)$$

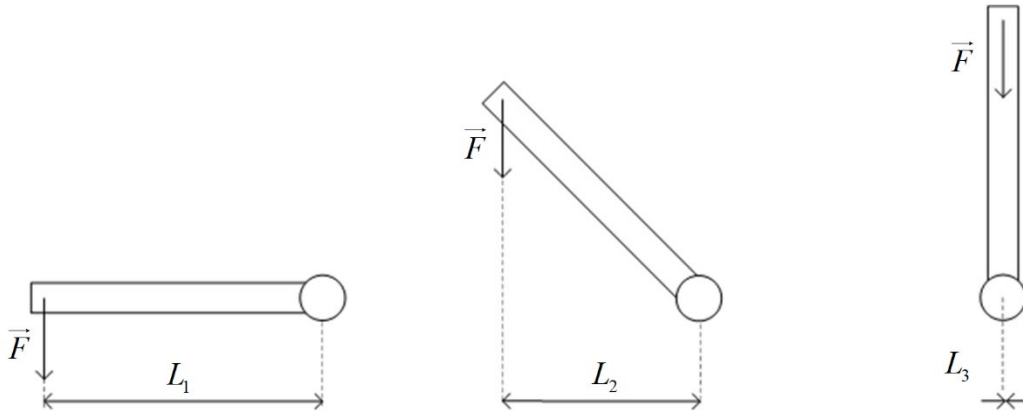
$$M = r \cdot F \cdot \sin \angle(\vec{r}, \vec{F}). \quad (2.2)$$

U slučaju robotskog manipulatora, za slučaj mirovanja, moment koji aktuatori treba da savladaju potiče od težine samog manipulatora i težine tereta. Samim tim, intenzitet pojedinačnih sila koje djeluju na zglobove manipulatora se može izraziti kao:

$$F = G = m \cdot g, \quad (2.3)$$

gdje je G težina, m masa, a g gravitaciono ubrzanje. Standardna jedinica SI sistema za moment je Nm . Međutim, obrtni momenti servo motora SPT – 3000 manipulatora su dati u $kg \cdot cm$, koja ne predstavlja standardnu SI jedinicu za moment. Na primjer, ukoliko je maksimalni moment servo motora $5kg \cdot cm$, to znači da je na rastojanju $1cm$ od osovine motora, motor u stanju da podigne teret od $5kg$. Radi upoređivanja izračunatih momenta zglobova, čija je jedinica Nm , i maksimalnog momenta odgovarajućih aktuatora, obrtne momente zglobova je po izračunavanju potrebno preskalirati, odnosno podijeliti gravitacionim ubrzanjem g .

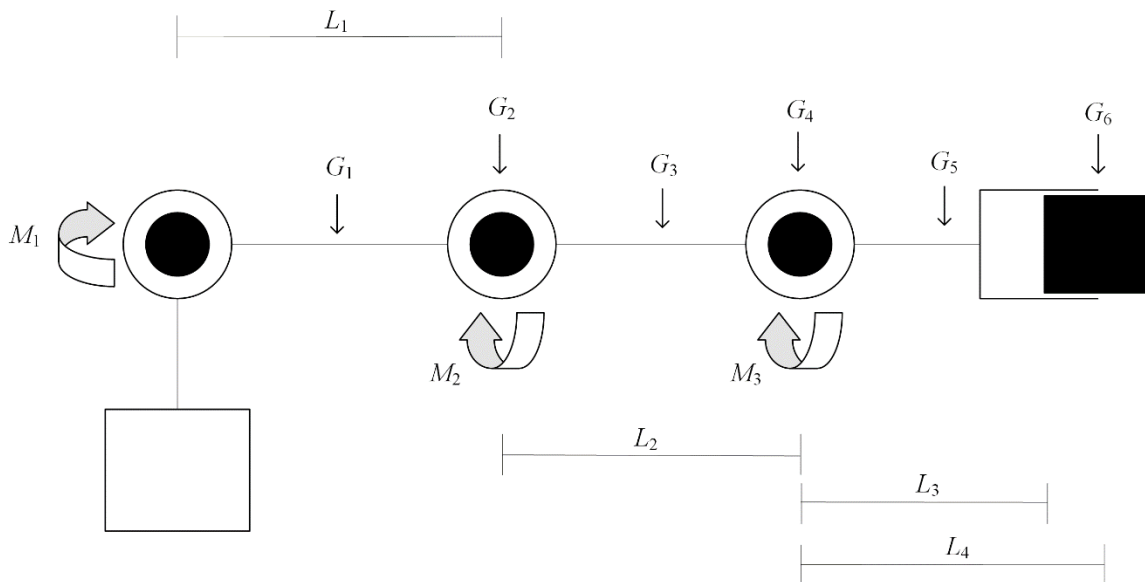
Da bi se procijenio moment potreban za svaki zglob, neophodno je izabrati najgori scenario. Slika 2.20 prikazuje polugu dužine L koja se rotira u smjeru kazaljke na satu i na čijem kraju djeluje sila intenziteta F . Moment se, kao što je već objašnjeno, može izraziti kao proizvod intenziteta sile i normalne udaljenosti od linije njenog djelovanja. Sa slike se može uočiti da se udaljenost linije djelovanja sile smanjuje, a samim tim se smanjuje i moment, jer je sila konstantna. Ovo znači da zglob čija je osa rotacije vertikalna ne učestvuje u podizanju tereta, kao i da su zglobovi koji podižu teret najopterećeniji kada su segmenti u horizontalnom položaju [28].



Slika 2.20. Rotacija poluge

2.4.2. Obrtni moment zglobova

Slika 2.21 prikazuje konfiguraciju manipulatora – zglobove, dužine i mase segmenata, radi izračunavanja obrtnih momenata za svaki pojedinačni zglob. Prilikom dizajna robotskih manipulatora težina robotske ruke i predviđena težina tereta se razmatraju prilikom izbora motora da bi se postigao dobar dizajn koji može da podrži težinu manipulatora i tereta koji se prenosi.



Slika 2.21. Obrtni momenti manipulatora

Računanje momenta se vrši pojedinačno za svaki zglob koji vrši podizanje tereta. Manipulator koji se koristi u radu ima tri stepena slobode koja vrše podizanje, dok preostali stepen slobode (bazni zglob) ne vrši podizanje. Smatra se da je centar mase svakog segmenta u njegovoj sredini. Za svaki zglob manipulatora mora važiti da ukupni obrtni moment zgloba mora biti manji od ili jednak momentu koji stvara aktuator tog zgloba. Na osnovu slike 2.18 vrši se određivanje obrtnih momenata zglobova na sljedeći način:

1) obrtni moment za zglob ramena:

$$M_1 = \frac{L_1}{2} G_1 + L_1 G_2 + \left(L_1 + \frac{L_2}{2} \right) G_3 + (L_1 + L_2) G_4 + (L_1 + L_2 + L_3) G_5 + (L_1 + L_2 + L_4) G_6, \quad (2.4)$$

2) obrtni moment za zglob lakta:

$$M_2 = \frac{L_2}{2} G_3 + L_2 G_4 + (L_2 + L_3) G_5 + (L_2 + L_4) G_6, \quad (2.5)$$

3) obrtni moment za zglob šake:

$$M_3 = L_3 G_5 + L_4 G_6, \quad (2.6)$$

gdje je G_1 – težina segmenta nadlaktice, G_2 – težina aktuatora lakta, G_3 – težina segmenta podlaktice, G_4 – težina aktuatora zgloba šake, G_5 – masa sklopa hvataljke, G_6 – masa tereta, L_1 – dužina nadlaktice, L_2 – dužina podlaktice, L_3 – rastojanje od zgloba šake do centra mase hvataljke, L_4 – rastojanje od zgloba šake do centra mase tereta.

Parametri za računanje obrtnih momenata zglobova su sljedeći: $m_1 = 52g$, $m_2 = 68g$, $m_3 = 45g$, $m_4 = 64g$, $m_5 = 71g$, $m_6 = 100g$, $g = 9,81 \text{ m/s}^2$, $L_1 = 134mm$, $L_2 = 132mm$, $L_3 = 71mm$, $L_4 = 118mm$.

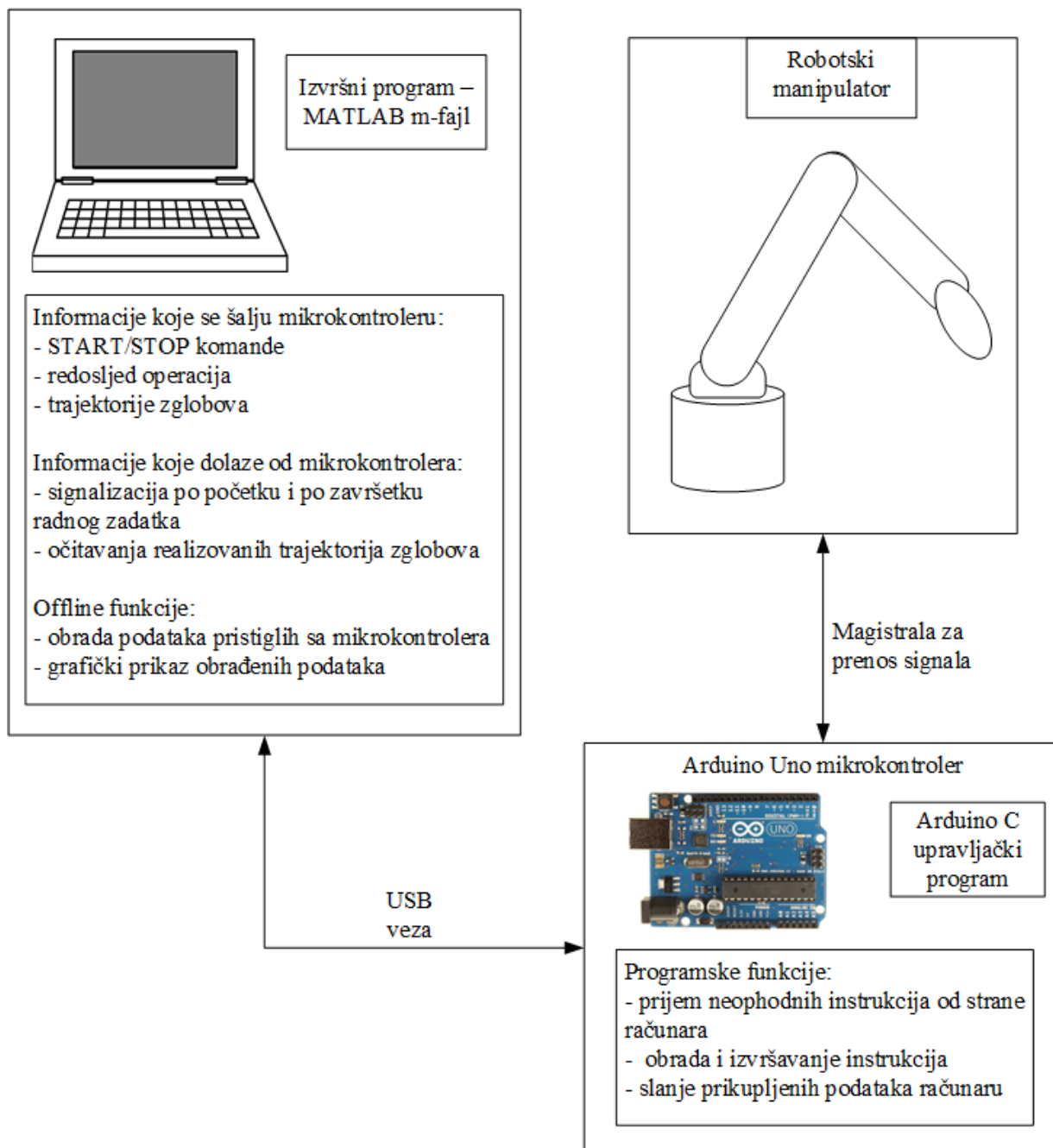
Na osnovu jednačina (2.4), (2.5) i (2.6) se računaju obrtni momenti u zglobovima koji vrše podizanje tereta. Masa tereta za dati primjer iznosi 100g. Momenti zglobova i momenti odabranih aktuatora su dati u tabeli 2.2.

Zglob	Obrtni moment zgloba	Obrtni moment izabranog motora
Zglob ramena	10,094kg·cm	20kg·cm
Zglob lakta	5,083kg·cm	20kg·cm
Zglob šake	1,684kg·cm	13kg·cm

Tabela 2.2. Momenti zglobova i aktuatora

2.5. Laboratorijska postavka manipulacionog robota

Laboratorijska postavka manipulacionog robota SPT – 3000 se sastoji od tri zasebna podsistema (slika 2.22): elektromehaničkog (robotski manipulator sa svim svojim elementima), nižeg upravljačkog nivoa (Arduino mikrokontrolera) i višeg upravljačkog nivoa (MATLAB program koji se izvršava na računaru).



Slika 2.22. Struktura sistema

Arhitektura i funkcionisanje sistema sa slike 2.22 se može zaključiti na osnovu praćenja toka podataka. Najprije, operator definiše radni zadatak u izvršnom programu na računaru. Na osnovu unesenih podataka, MATLAB program definiše redosljed sekvencu operacija koje robot treba da izvrši, kao i trajektorije zglobova. Neophodni podaci i instrukcije se zatim šalju mikrokontroleru putem USB konekcije. Nakon prijema podataka, mikrokontroler izvršava instrukcije zadatim redosljedom.

Trajektorije zglobova zapravo predstavljaju širine impulsa upravljačkih signala koji se šalju servo motorima u zglobovima manipulatora. Frekvencija upravljačkih signala je 50Hz, što znači da svako se na svakih 20 milisekundi vrši novo zadavanje pozicija servo motora. Između zadavanja pozicija, mikrokontroler prikuplja informacije sa o trenutnim pozicijama servo motora. Algoritam upravljanja servo motorima je PID i on je implementiran direktno unutar servo motora. Frekvencija oscilatora unutar Arduino mikrokontrolera je 16Mhz, što znači da mikrokontroler izvršava po jednu instrukciju na svakih 62 nanosekunde.

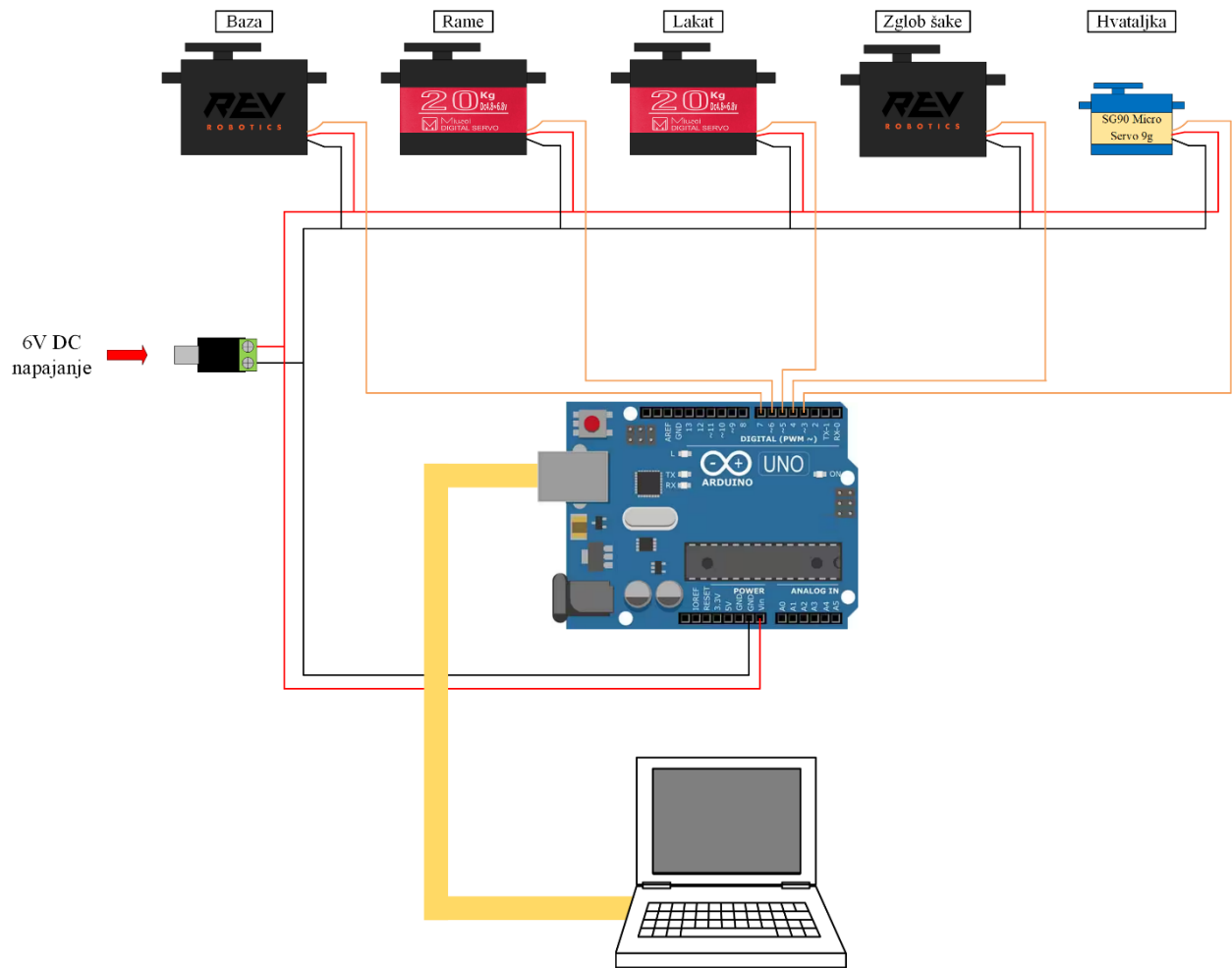
U toku izvršavanja radnog zadatka, Arduino šalje prikupljene informacije računaru. Kako ograničenje programske memorije Arduino mikrokontrolera ne dozvoljava skladištenje informacija o trajektorijama zglobova u svakom trenutku vremena, Arduino na osnovu instrukcija dobijenih od MATLAB-a samostalno vrši računanje trajektorija zglobova u realnom vremenu i na osnovu istih vrši generisanje impulsa kojima se upravlja servo motorima. Nakon završetka radnog zadatka i obrade informacija dobijenih od Arduina, na računaru vrši se prikaz zadatih i realizovanih trajektorija i analiza performansi sistema.

Povezivanje komponenti sistema se može podijeliti na dvije faze – mehanička i električna. Mehanička faza podrazumijeva povezivanje segmenata manipulatora sa servo motorima i montiranje manipulatora na platformu. Unutar platforme se nalazi Arduino mikrokontroler i štampana ploča koja služi za električno povezivanje svih komponenti. Prikaz laboratorijske postavke robota je dat na slici 2.23.

Postolje robota je napravljeno tako da, osim što je na njemu smješten manipulator, se u njemu sadrži i cjelokupni elektronski dio manipulatora – mikrokontroler zajedno sa razvodnom pločom i kablovima koji vode do servo motora. Električna faza podrazumijeva povezivanje svih komponenti na napajanje, povezivanje svih servo motora sa mikrokontrolerom i povezivanje mikrokontrolera sa računarom. Šema električnog povezivanja komponenti je data na slici 2.24.



Slika 2.23. Laboratorijska postavka robotskog manipulatora



Slika 2.24. Električna šema sistema

3. Kinematika robota

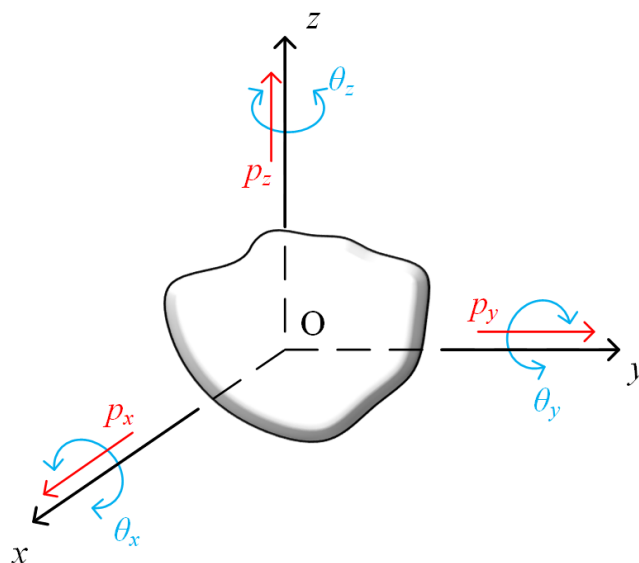
Ova glava je posvećena proučavanju kinematike robotskog manipulatora. Kinematika predstavlja fundamentalni aspekt u razumijevanju kretanja i pozicioniranja robotskih manipulatora. Ovaj konceptualni i teorijski okvir omogućava analizu i predviđanje kretanja svakog segmenta manipulatora, kao i relacija među njima. U ovom poglavlju će biti detaljno izloženi osnovni elementi kinematike robotskog manipulatora, uključujući geometriju, koordinatne sisteme, matrice transformacije i Denavit – Hartenbergovu (D – H) konvenciju [30]. Korišćenjem ovih alata se vrši matematičko modelovanje i analiza kinematike manipulatora.

Na kinematički model robotskog manipulatora se nadovezuju problemi direktne i inverzne kinematike, koji su od ključnog značaja za određivanje položaja i orijentacije završnog alata manipulatora. Proučavanjem direktnog i inverznog kinematičkog problema otvoriće se prostor za razumijevanje i implementaciju različitih algoritama i metoda upravljanja manipulatorom. Kroz ovo poglavlje će se steći dublji uvid u kinematiku robotskog manipulatora i njenu ulogu u njegovom pozicioniranju u prostoru. Proučavanje kinematike robotskog manipulatora takođe predstavlja osnovu za planiranje trajektorije i upravljanja manipulatorom, o čemu će biti više riječi u narednoj glavi.

3.1. Osnovni teorijski pojmovi

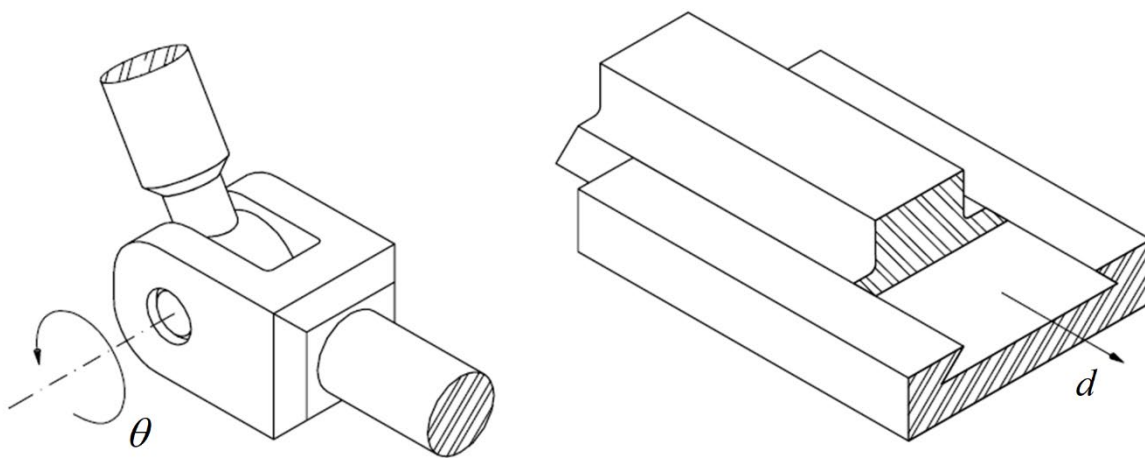
Kinematička određenost robota podrazumijeva određenost pozicije i orijentacije hvataljke u odnosu na predmete u radnom prostoru robota, kao i u odnosu na neki nepomični referentni koordinatni sistem. Sa druge strane, položaj robota je određen relativnim ugaonim, odnosno relativnim translatorskim pomjerajima u zglobovima robota. Da bi robot obavio ispravno radni zadatak, u svakom trenutku moraju biti određeni pozicija i orijentacija završnog organa u prostoru.

Položaj tijela u prostoru je određen sa 6 nezavisnih parametara – tri translacije i tri rotacije. Dakle, tijelo u prostoru ima šest stepeni slobode kretanja – $f = 6$. Drugim riječima, tijelo se može kretati na šest različitih, nezavisnih načina: *translatorno*, za pomjeraje p_x , p_y i p_z duž osa x , y i z , čime se postiže pozicioniranje određene tačke tijela u prostoru i *rotaciono* oko sve tri ose za vrijednosti uglova θ_x , θ_y i θ_z (slika 3.1).



Slika 3.1. Kretanje tijela u prostoru

Kada se dva tijela međusobno povežu, pri čemu je bar jedno od njih pokretno, nastaje zglob, odnosno kinematički par. Nastajanjem zgloba dolazi do smanjenja mogućnosti kretanja, pa je samim tim i stepen slobode kretanja tijela manji, tj. $f < 6$. Osnovni zglobovi su *rotacioni* i *translatorni* zglob i oni imaju $f = 1$ stepen slobode kretanja. Svi ostali zglobovi sa $f > 1$ stepeni slobode kretanja svode na ova dva, osnovna zgloba. Kako se svi zglobovi mogu svesti na njih, rotacioni i translatorni zglob su najvažniji kinematički parovi u robotici. Primjeri rotacionog i translatornog zgloba su dati na slici 3.2.



Slika 3.2. Rotacioni i translatorni zglob [27]

3.1.1. Kinematički lanci

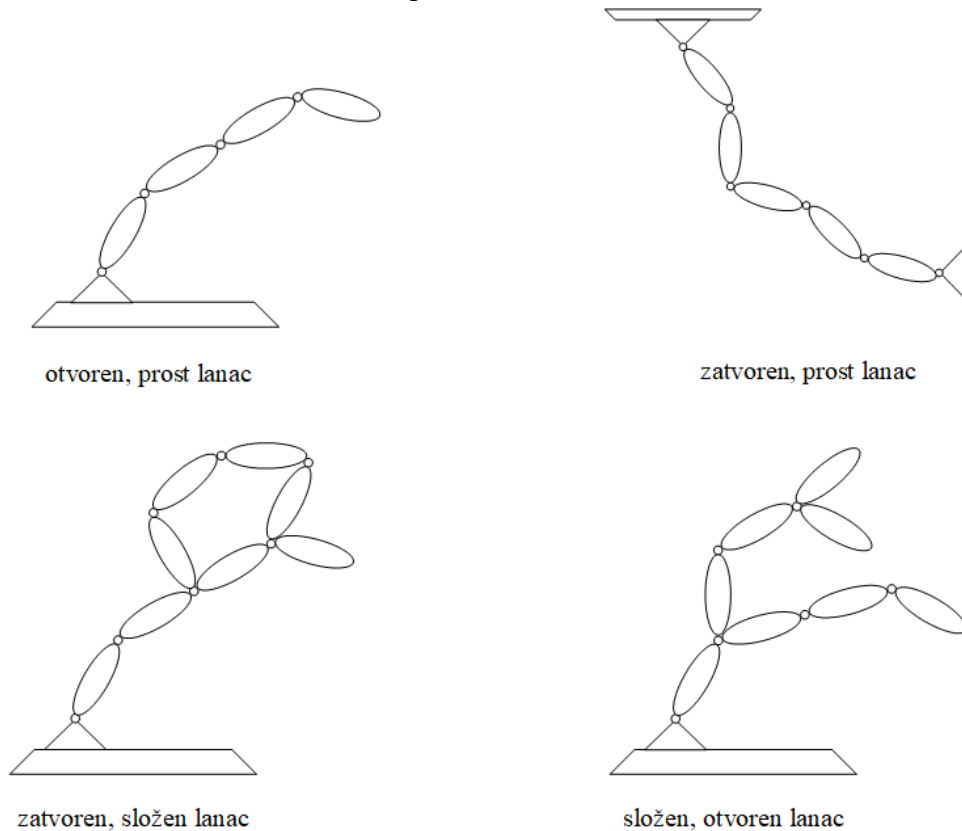
Kinematička struktura robota u suštini predstavlja *kinematički lanac*. Kinematički lanac se definiše kao skup od n međusobno povezanih kinematičkih parova. Konfiguracija industrijskog robota se sastoji od n nedeformabilnih tijela koja predstavljaju segmente, koji su međusobno povezani zglobovima. Svaki zglob industrijskog manipulatora najčešće ima jedan stepen slobode kretanja, odnosno zglobovi industrijskog robota su najčešće rotacioni ili translatorni.

Zavisno od strukture veza koje su ostvarene u kinematičkom lancu, razlikuju se prosti i složeni kinematički lanci. Lanac kod kojeg svaki segment u nizu ima samo dva zgloba, prethodni i naredni, naziva se *prosti* kinematički lanac. *Složeni* kinematički lanac je onaj kod kojeg barem jedan segment ima tri ili više zglobova.

Još jedna način podjele kinematičkih lanaca je na *otvorene* i *zatvorene*. Otvoreni kinematički lanac je onaj kod kojeg ne postoji zatvoreni niz, tj. iz jedne tačke lanca u drugu se može doći samo jednim putem. U slučaju otvorenog kinematičkog lanca početni segment je vezan za čvrstu podlogu, dok je uloga zadnjeg segmenta nošenje završnog uređaja. Kako svaki zglob ima po jedan stepen slobode kretanja ($f = 1$), cjelokupni kinematički lanac će imati

$$f_L = n \cdot f = n \cdot 1 = n \quad (3.1)$$

stepeni slobode kretanja, pri čemu je n broj zglobova. Zatvoreni kinematički lanac ima najmanje jedan zatvoreni niz segmenata, kod kojeg se od jedne do druge tačke iz tog niza može doći na bar dva načina. Vrste kinematičkih lanaca su prikazane na slici 3.3.



Slika 3.3. Kinematički lanci



Slika 3.4. Promjena kinematičke strukture u postupku montaže

Za robotske mehanizme karakteristično je da u toku rada kinematički lanac mijenja svoju strukturu i to od otvorene ka zatvorenoj i obrnuto. Karakterističan primjer promjene strukture kinematičkog lanca je u postupku montaže, gdje robot u fazi prenosa predmeta predstavlja otvoren lanac, dok u toku montaže predmeta predstavlja zatvoren lanac. Slika 3.4 prikazuje primjer promjene strukture kinematičkog lanca.

U analizi, sintezi i upravljanju robotima izuzetan značaj ima kinematičko opisivanje robota kao otvorenog kinematičkog lanca. Kod zglobnog kinematičkog lanca broj stepeni slobode je jednak broju nezavisnih parametara koji su potrebni da bi se jednoznačno odredio položaj cjelokupnog lanca.

Da bi se definisao položaj kod jednog otvorenog kinematičkog lanca, sa osnovnim kinematičkim parovima, neophodno je znati promjene koordinata u svim zglobovima lanca, odnosno potrebno je znati za koju vrijednost ugla je zarotiran svaki rotacioni zglob i koliko je pomjeranje u svakom translatorsnom zglobu. Kod ovakvih lanaca zglobovi su međusobno nezavisni i pomjeranja u zglobovima mogu biti proizvoljna. Broj stepeni slobode kod prostih, otvorenih kinematičkih lanaca jednak je broju zglobova.

3.1.2. Stepeni slobode kinematičkog lanca

Broj stepeni slobode kinematičkog lanca zavisi od njegove topološke strukture, tj. od toga da li je on otvoren ili zatvoren i od vrste primjenjenih zglobova. Translatorsni i rotacioni zglobovi kao najčešće primjenjivani u robotici imaju jedan stepen slobode kretanja ($f_z = 1$) jer dopuštaju samo jednu vrstu kretanja, kako je to prikazano na slici 3.2. Nezavisna promjenljiva veličina kod rotacionog zgloba je njegov ugao rotacije θ , dok je kod translatorsnog zgloba nezavisna promjenljiva iznos njegove translacije d .

Ukoliko se slijedi definicija broja stepeni slobode za jedan kinematički par dolazi se do broja stepeni slobode cjelokupnog zglobnog kinematičkog lanca, koji je jednak broju mogućih nezavisnih kretanja tog lanca, odnosno broju nezavisnih parametara potrebnih da bi se jednoznačno odredio položaj tog lanca.

U opštem slučaju, broj stepeni slobode kretanja za zglobne sisteme po Grübler – Kutzbach kriterijumu se izračunava na sljedeći način:

$$f = \sum_{i=1}^{n_z} f_{zi} - 6n_p, \quad (3.2)$$

gdje je f_{zi} broj stepeni slobode kretanja i -tog zgloba, a n_p broj nezavisnih petlji unutar kinematičkog lanca. Broj nezavisnih petlji n_p se računa po formuli

$$n_p = n_z - n_s, \quad (3.3)$$

gdje je n_z broj zglobova, a n_s broj tijela ili segmenata.

Na primjer, posmatra se kinematička struktura manipulacionog robota SPT – 3000, koji se koristi u radu (slika 2.2). Za izračunavanje broja stepeni slobode kretanja manipulatora sa slike koristi se obrazac (3.2), polazeći od činjenice da je u pitanju struktura sa rotacionim kinematičkim parovima, sa po jednim stepenom slobode ($f_{zi} = 1$).

Na osnovu obrasca (3.3) koji definiše broj nezavisnih petlji n_p , pri čemu se sa slike 2.2 može uočiti da je broj zglobova $n_z = 4$ i broj segmenata $n_s = 4$ (bazni segment se posmatra kao dio osnove), dobija se sljedeće:

$$n_p = n_z - n_s = 4 - 4 = 0. \quad (3.4)$$

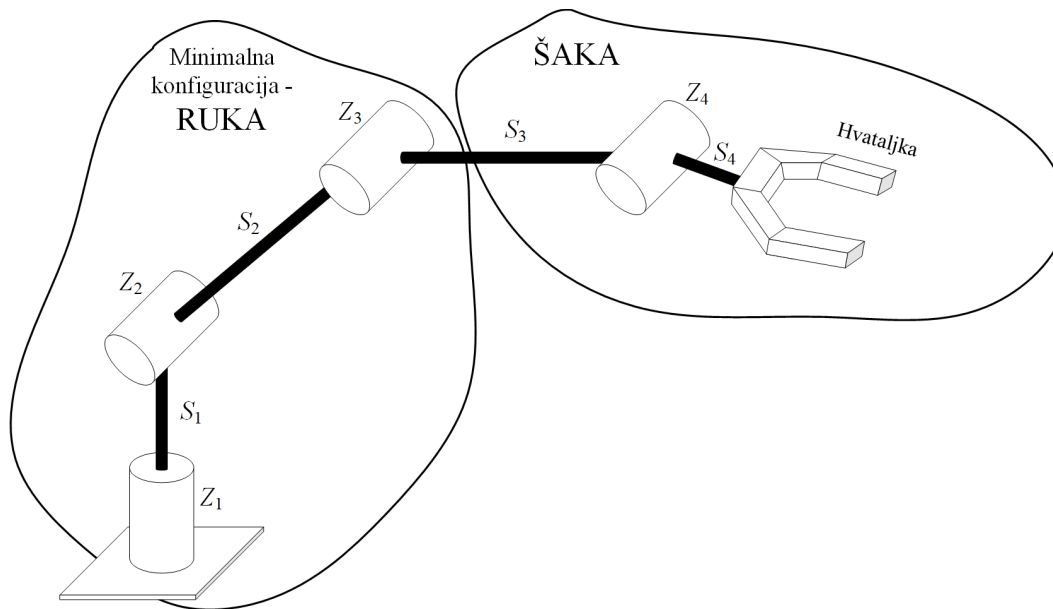
Iz izraza (3.4) se može zaključiti da kinematička struktura na slici 2.2 ne sadrži nezavisne petlje, upravo jer se radi o prostom, otvorenom kinematičkom lancu. Na osnovu obrasca (3.2), broj stepeni slobode SPT–3000 iznosi:

$$f = \sum_{i=1}^{n_z} f_{zi} - 6n_p = 4 - 6 \cdot 0 = 4. \quad (3.5)$$

Dakle, robotski manipulator koji se koristi u radu ima četiri stepena slobode, odnosno za definisanje položaja manipulacionog robota SPT – 3000 potrebna su četiri nezavisna parametra!

3.1.3. Minimalna konfiguracija robota

Da bi jedan robotski manipulator bio u upotrebi, on mora imati barem mogućnost pozicioniranja u prostoru. To znači da treba da se sastoji od makar tri zgloba, odnosno mora imati minimum tri stepena slobode kretanja. Struktura koja sadrži tri zgloba (stepena slobode) se naziva *minimalna konfiguracija robota* [21]. Na ovakvu minimalnu konfiguraciju se nadovezuje završni uređaj (hvataljka). Manipulator koji se koristi u radu ima četiri stepena slobode, pa se samim tim može podijeliti na dva dijela: *ruku* i *šaku*. On u prostoru može postići potpuno pozicioniranje pomoću tri ose ruke i djelimičnu orijentaciju pomoću ose šake, što je i prikazano na slici 3.5. Da bi se postigla potpuna orijentacija, neophodno je da šaka posjeduje tri stepena slobode, odnosno da kompletan robot posjeduje šest stepeni slobode.



Slika 3.5. Ruka i šaka robota

Minimalna konfiguracija se često definiše kao mehanizam sa tri stepena slobode kretanja koji vrši pozicioniranje, tj. dovođenje šake na željeno mjesto u radnom prostoru, pa se samim tim može reći da minimalna konfiguracija definiše poziciju korijena šake, koji predstavlja mjesto njenog priključka. Normalno, ova minimalna konfiguracija robota može se sastojati od bilo koje kombinacije zglobova (translatorni ili rotacioni). Broj mogućih varijacija strukture robota se može odrediti na osnovu jednakosti:

$$V = n^f, \quad (3.6)$$

gdje je V broj varijacija, n broj zglobova i f broj stepeni slobode kretanja. Za tri stepena slobode kretanja i dva osnovna zgloba ruke (R i T) postoji $V = 2^3 = 8$ minimalnih konfiguracija robota (TTT , TTR , TRT , RTT , TRR , RTR , RRT , RRR). U slučaju manipulatora SPT – 3000, kako su svi zglobovi rotacioni, u pitanju je RRR minimalna konfiguracija.

Definicijom minimalne konfiguracije robota se određuje njegov radni prostor u kojem on može obavljati radne zadatke, odnosno skup položaja u prostoru u koje on može dovesti završni organ. Svaka od navedenih konfiguracija ima svoje dobre i loše strane, pri čemu rješenja sa rotacionim zglobovima imaju jednostavniju mehaničku konstrukciju, složenije programiranje kretanja i brži su od konfiguracija sa translatorskim zglobovima.

3.2. Radni prostor

Na osnovu definisane minimalne konfiguracije robota je moguće odrediti i njegov *manipulacioni prostor*. Manipulacioni prostor je skup tačaka u koje robot može dovesti hvataljku (centar prihvata, vrh alata), odnosno u kojem se može obaviti radni zadatak. Izvan manipulacionog prostora robot nije u stanju da obavlja radne zadatke.

Radni prostor robotskog manipulatora čine samo one tačke manipulacionog prostora koje robot može dostići uz proizvoljnu orijentaciju. Drugim riječima, radni prostor je dio manipulacionog prostora u kojem je moguć zahvat iz svih smjerova, dok je u preostalom manipulacionom prostoru moguće zahvat moguće izvesti samo pod određenim uglom [21]. Radni i manipulacioni prostor su određeni kinematičkom strukturom i geometrijom kinematičkih parova robota, kao i ograničenjima postavljenim na pojedine stepene slobode kretanja pojedinih zglobova. Dakle, veličina radnog prostora zavisi od tipa i broja zglobova, dužine segmenata i postojećih fizičkih ograničenja koja su neposredno povezana sa samom građom i izgledom manipulatora. Kako prva tri zgloba određuju položaj ručnog zgloba, a preostale ose određuju orijentaciju hvataljke, može se zaključiti da minimalna konfiguracija određuje geometriju radnog prostora robota. Radni prostor može biti *uređen* ili *neuređen*. Za radni prostor se smatra da je uređen ako je definisan položaj i orijentacija proizvoda (radnog predmeta) i ukoliko nema prepreka u radnom prostoru. Što je uređenost radnog prostora veća, to je potrebni nivo inteligencije robota manji.

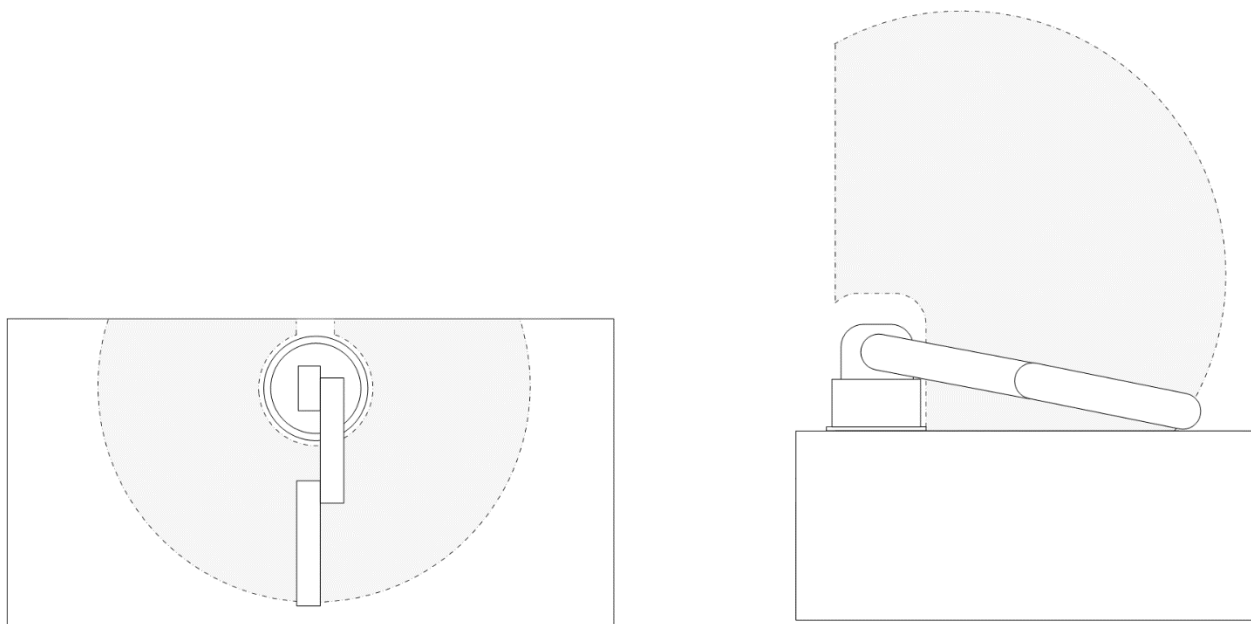
Konfiguracija manipulatora koji se koristi u radu je rotaciona (*RRR*), koja se još naziva i laktasta, antropomorfna ili zglobna konfiguracija. Radni prostor za ovu konfiguraciju, za slučaj kada ne postoje ograničenja rotacionih kretanja, je sfera sa određenim ograničenjima nametnutim samom konfiguracijom manipulatora (slika 3.6). U slučaju kada postoje dodatna ograničenja u kretanju, radni prostor je dio kugle složenog oblika čiji presjek sa bočne strane ima oblik polumjeseca.

Veličinu radnog prostora robota je moguće približno odrediti pomoću dohvata i hoda. Horizontalan dohvat je maksimalna udaljenost koju ručni zglob može dosegnuti. Mjeri se od vertikalne ose oko koje robot rotira. Horizontalan hod predstavlja ukupnu udaljenost od vertikalne ose po kojoj se ručni zglob može kretati. Razlika između horizontalnog dohvata i hoda je minimalna udaljenost ručnog zgloba od glavne vertikalne ose. S obzirom da je ta veličina pozitivna, tada je dohvat uvijek veći ili jednak hodu. Vertikalni dohvat robota predstavlja maksimalnu udaljenost ručnog zgloba robota od baze. Vertikalni hod robota je ukupna vertikalna udaljenost po kojoj se ručni zglob može kretati i manji je ili jednak vertikalnom dohvatu.

Kako je SPT – 3000 postavljen na platformi pravogaonog oblika, uzimajući u obzir poziciju manipulatora na platformi (slika 2.7) i ograničenja uzrokovana samom konfiguracijom manipulatora, može se zaključiti da je radni prostor u ovom slučaju sfera složenog oblika. Ravanski prikaz radnog prostora SPT – 3000 dat je na slici 3.7.



Slika 3.6. Radni prostor SPT – 3000 manipulatora



Slika 3.7. Ravanski prostor radnog prostora SPT – 3000

3.3. Osnovni pojmovi kinematike robota

Robot se može modelovati kao lanac krutih tijela-segmenata, međusobno povezanih zglobovima, gdje se na početku lanca nalazi nepomična baza robota, a na kraju lanca je završni uređaj (hvataljka). Da bi se robot kretao u trodimenzionalnom prostoru, gdje obavlja određene poslove, potrebno je upravljati pozicijom i orijentacijom hvataljke. Zbog toga je neophodno odrediti vezu između parametara zglobova robota i pozicije i orijentacije hvataljke.

Da bi se adekvatno opisala kinematika robota, najprije je neophodno upoznati se sa osnovnim kinematičkim pojmovima, tj. Denavit – Hartenbergovom notacijom, unutrašnjim i spoljašnjim koordinatama i homogenim transformacijama.

U robotici postoje dva osnovna pristupa u opisivanju kinematike robota:

- Denavit – Hartenbergov analitički pristup i
- numerički pristup na bazi Rodrigezove formule [31].

Denavit – Hartenbergov pristup (D – H notacija) je prikladniji kada se formiraju kinematičke jednačine robota, dok je pristup na bazi Rodrigezove formule prikladniji za formiranje dinamičkih jednačina robota. D – H notacija se zasniva na homogenim transformacijama, koje definišu položaj i orijentaciju jednog koordinatnog sistema u odnosu na drugi.

Da bi se dostigao željeni položaj robota, on se najprije se mora zadati. Razlikuju se dva načina zadavanja položaja robota:

- preko unutrašnjih koordinata robota (koordinata zglobova) i
- preko spoljašnjih koordinata robota.

3.3.1. Spoljašnje i unutrašnje koordinate

Unutrašnje koordinate jednog robotskog manipulatora predstavljaju skalarne veličine koje opisuju relativni položaj jednog segmenta u odnosu na drugi segment kinematičkog para. Kod rotacionog kinematičkog para unutrašnja koordinata je ugao zakretanja u zglobu, dok je kod translacionog zgloba unutrašnja koordinata linearni pomjeraj duž ose zgloba. Unutrašnje koordinate se najčešće obilježavaju sa q_i ($i = 1, \dots, n$) i čine *vektor unutrašnjih koordinata*:

$$\mathbf{q} = [q_1 \quad q_2 \quad \dots \quad q_n]^T. \quad (3.7)$$

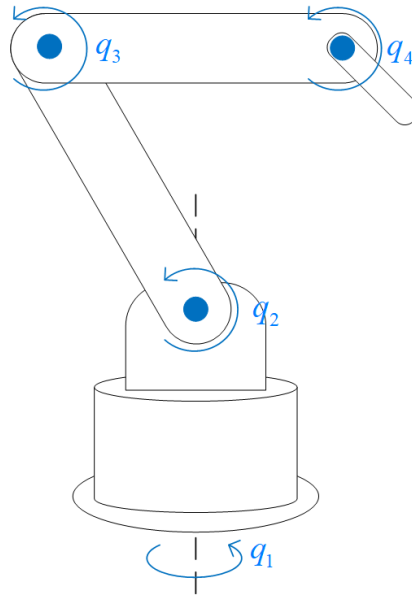
Broj unutrašnjih koordinata zavisi od broja stepeni slobode kretanja manipulatora. Kako manipulator koji se koristi u radu ima četiri stepena slobode, samim tim će se i njegov položaj opisivati sa četiri unutrašnje koordinate (slika 3.8).

Pod spoljašnjim koordinatama robota se podrazumijevaju veličine koje opisuju položaj završnog alata robota u odnosu na nepokretni koordinatni sistem, koji je vezan za bazu robota. Pozicija završnog organa (hvataljke) se najčešće opisuje Dekartovim koordinatama – p_x , p_y i p_z , dok se orijentacija hvataljke opisuje pomoću Ojlerovih uglova [32] - θ , φ i ψ između osa koordinatnog sistema vezanog za hvataljku i nepokretnog (baznog) koordinatnog sistema. Ugao skretanja θ predstavlja ugao rotacije oko z – ose nepokretnog koordinatnog sistema, ugao posrtanja

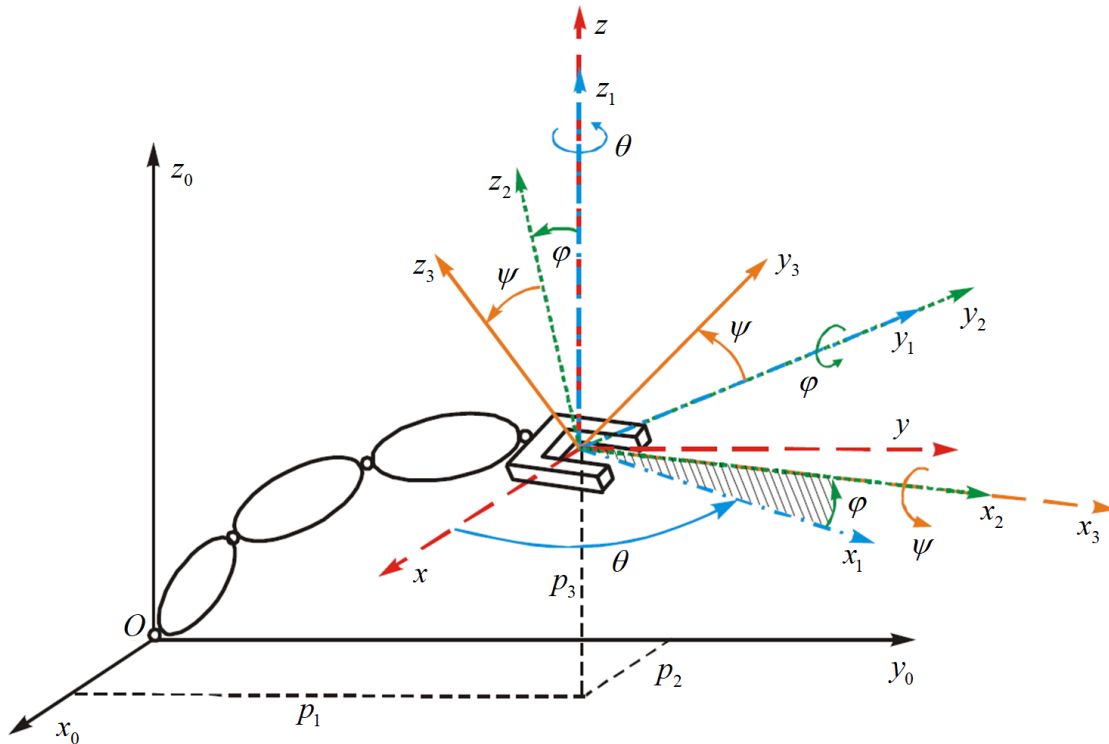
φ odgovara rotaciji oko novodobijene ose y_1 , dok ugao valjanja ψ opisuje rotaciju oko novodobijene ose x_2 . Prikaz spoljašnjih koordinata završnog alata robota su dati na slici 3.9.

Konačno, vektor spoljašnjih koordinata se može zapisati kao:

$$\mathbf{r} = [p_x \quad p_y \quad p_z \quad \theta \quad \varphi \quad \psi]^T. \quad (3.8)$$



Slika 3.8. Unutrašnje koordinate



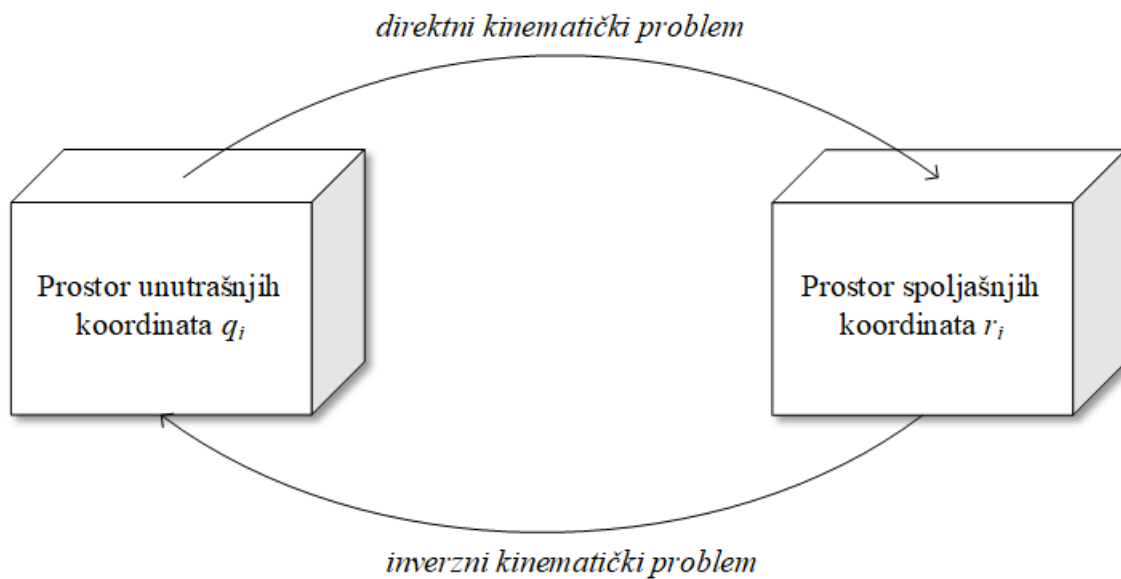
Slika 3.9. Spoljašnje koordinate [21]

Pri promjeni unutrašnjih koordinata robota q_i ($i=1, \dots, n$), dolazi do promjene i spoljašnjih koordinata robota r_i ($i=1, \dots, m$). Veza između spoljašnjih i unutrašnjih koordinata se može opisati relacijom:

$$\mathbf{r} = \mathbf{f}(\mathbf{q}), \quad (3.9)$$

gdje je $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ nelinearna, neprekidna, diferencijabilna vektorska funkcija koja vrši preslikavanje iz prostora unutrašnjih koordinata u prostor spoljašnjih koordinata.

Postupak određivanja vektora spoljašnjih koordinata \mathbf{r} na osnovu zadatog vektora unutrašnjih koordinata je poznat kao *direktni kinematički problem*. Inverzni postupak, odnosno određivanje vektora koordinata zglobova na osnovu zadate pozicije i orijentacije hvataljke, se naziva *inverzni kinematički problem*. Veza između unutrašnjih i spoljašnjih koordinata je data na slici 3.10.

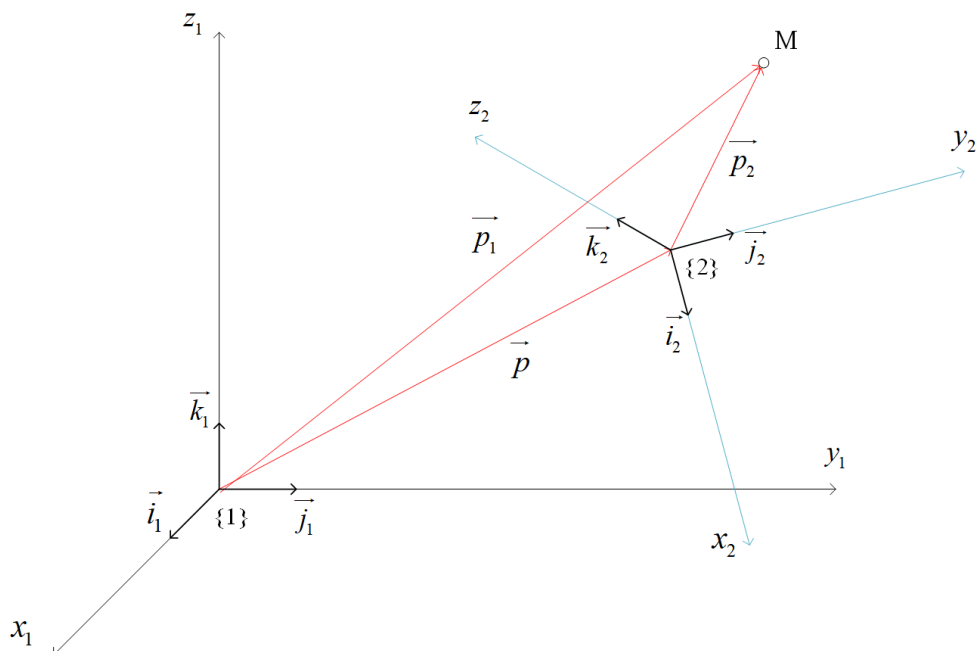


Slika 3.10. Veza između spoljašnjih i unutrašnjih koordinata

3.3.2. Homogene transformacije

Rješavanjem direktnog kinematičkog problema određuje se relativni položaj hvataljke manipulatora u odnosu na nepokretni, referentni koordinatni sistem, u funkciji unutrašnjih koordinata u zglobovima robota. Pri rješavanju ovog problema koriste se *matrice homogenih transformacija*.

Homogenim transformacijama se definišu pozicija i orijentacija jednog koordinatnog sistema u odnosu na drugi. Drugim riječima, matricama homogenih transformacija se karakteristike zadatog vektora, odnosno tačke preračunavaju iz jednog koordinatnog sistema u drugi.



Slika 3.11. Transformacija koordinata iz jednog koordinatnog sistema u drugi

Na slici 3.11 su prikazana dva koordinatna sistema, pri čemu je koordinatni početak sistema {2} pomjeren u odnosu na koordinatni početak sistema {1} za vektor \vec{p} . Sistem {2} je takođe zarotiran za određeni ugao u odnosu na sistem {1}. Vektor položaja \vec{p}_1 proizvoljne tačke M u odnosu na sistem {1} je

$$\vec{p}_1 = \vec{p} + \vec{p}_2, \quad (3.10)$$

gdje je \vec{p}_2 vektor položaja tačke M u odnosu na sistem {2}.

Izraz (3.10) se može zapisati i u obliku

$$x_1 \vec{i}_1 + y_1 \vec{j}_1 + z_1 \vec{k}_1 = p_x \vec{i}_1 + p_y \vec{j}_1 + p_z \vec{k}_1 + x_2 \vec{i}_2 + y_2 \vec{j}_2 + z_2 \vec{k}_2, \quad (3.11)$$

gdje su:

- $\vec{i}_1, \vec{j}_1, \vec{k}_1$ - jedinični vektori prvog koordinatnog sistema,
- $\vec{i}_2, \vec{j}_2, \vec{k}_2$ - jedinični vektori drugog koordinatnog sistema,
- x_1, y_1, z_1 - koordinate tačke M u prvom sistemu,
- p_x, p_y, p_z - koordinate koordinatnog početka sistema {2} u sistemu {1},
- x_2, y_2, z_2 - koordinate tačke M u sistemu {2}.

Ukoliko se jednačina (3.11) pomnoži skalarno sa setom jediničnih vektora $\vec{i}_1, \vec{j}_1, \vec{k}_1$, dobija se sljedeći set od tri jednačine:

$$\begin{aligned}
x_1 &= p_x + x_2 (\vec{i}_1 \cdot \vec{i}_2) + y_2 (\vec{i}_1 \cdot \vec{j}_2) + z_2 (\vec{i}_1 \cdot \vec{k}_2), \\
y_1 &= p_y + x_2 (\vec{j}_1 \cdot \vec{i}_2) + y_2 (\vec{j}_1 \cdot \vec{j}_2) + z_2 (\vec{j}_1 \cdot \vec{k}_2), \\
z_1 &= p_z + x_2 (\vec{k}_1 \cdot \vec{i}_2) + y_2 (\vec{k}_1 \cdot \vec{j}_2) + z_2 (\vec{k}_1 \cdot \vec{k}_2).
\end{aligned} \tag{3.12}$$

Kada se set jednačina (3.12) prikaže u matičnom obliku dobija se

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} (\vec{i}_1 \cdot \vec{i}_2) & (\vec{i}_1 \cdot \vec{j}_2) & (\vec{i}_1 \cdot \vec{k}_2) \\ (\vec{j}_1 \cdot \vec{i}_2) & (\vec{j}_1 \cdot \vec{j}_2) & (\vec{j}_1 \cdot \vec{k}_2) \\ (\vec{k}_1 \cdot \vec{i}_2) & (\vec{k}_1 \cdot \vec{j}_2) & (\vec{k}_1 \cdot \vec{k}_2) \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \tag{3.13}$$

ili, u obliku matrice dimenzija 4×4

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} (\vec{i}_1 \cdot \vec{i}_2) & (\vec{i}_1 \cdot \vec{j}_2) & (\vec{i}_1 \cdot \vec{k}_2) & p_x \\ (\vec{j}_1 \cdot \vec{i}_2) & (\vec{j}_1 \cdot \vec{j}_2) & (\vec{j}_1 \cdot \vec{k}_2) & p_y \\ (\vec{k}_1 \cdot \vec{i}_2) & (\vec{k}_1 \cdot \vec{j}_2) & (\vec{k}_1 \cdot \vec{k}_2) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix}. \tag{3.14}$$

U skraćenom obliku, matrica iz (3.14) se može zapisati kao

$$\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}. \tag{3.15}$$

Vektor \mathbf{p} iz (3.15) predstavlja *vektor translacije* koordinatnog sistema $\{2\}$ u odnosu na koordinatni sistem $\{1\}$ sa slike 3.11. Vektor translacije je dimenzija 3×1.

Matrica

$$\mathbf{R} = \begin{bmatrix} (\vec{i}_1 \cdot \vec{i}_2) & (\vec{i}_1 \cdot \vec{j}_2) & (\vec{i}_1 \cdot \vec{k}_2) \\ (\vec{j}_1 \cdot \vec{i}_2) & (\vec{j}_1 \cdot \vec{j}_2) & (\vec{j}_1 \cdot \vec{k}_2) \\ (\vec{k}_1 \cdot \vec{i}_2) & (\vec{k}_1 \cdot \vec{j}_2) & (\vec{k}_1 \cdot \vec{k}_2) \end{bmatrix} \tag{3.16}$$

predstavlja *matricu rotacije* koordinatnog sistema $\{2\}$ u odnosu na koordinatni sistem $\{1\}$. Matrica rotacije je dimenzija 3×3.

Matrica

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.17}$$

predstavlja *matricu transformacije* koordinatnog sistema $\{2\}$ u odnosu na koordinatni sistem $\{1\}$. Matrica transformacije je dimenzija 4×4 i u sebi sadrži vektor translacije i matricu rotacije. Osim

toga, prva tri člana poslednje vrste matrice transformacije su jednaki nuli, a poslednji član te vrste je jednak jedinici.

Da bi se formirao kinematički model robota, neophodno je izvršiti više uzastopnih transformacija, kako translacija, tako i rotacija koordinatnih sistema. U suštini, najbitniji odnos je između prvog, odnosno koordinatnog sistema fiksiranog za bazu, i zadnjeg, odnosno koordinatnog sistema vezanog za završni uređaj, sistema u kinematičkom lancu jednog robota.

U opštem slučaju transformacije koordinata iz nekog n -tog sistema u neki m -ti sistem, pri čemu je $m < n$, može se pisati

$$\begin{aligned}\mathbf{p}_m &= \mathbf{T}_{m,m+1}\mathbf{p}_{m+1}, \\ \mathbf{p}_{m+1} &= \mathbf{T}_{m+1,m+2}\mathbf{p}_{m+2}, \\ &\vdots \\ \mathbf{p}_{n-1} &= \mathbf{T}_{n-1,n}\mathbf{p}_n,\end{aligned}$$

iz čega se dobija da je

$$\mathbf{p}_m = \mathbf{T}_{m,m+1}\mathbf{T}_{m+1,m+2}\cdots\mathbf{T}_{n-1,n}\mathbf{p}_n. \quad (3.18)$$

Kako je

$$\mathbf{p}_m = \mathbf{T}_{m,n}\mathbf{p}_n, \quad (3.19)$$

tada se na osnovu izraza (3.18) i (3.19) matrica transformacije sistema n u odnosu na sistem m može zapisati kao

$$\mathbf{T}_{m,n} = \mathbf{T}_{m,m+1}\mathbf{T}_{m+1,m+2}\cdots\mathbf{T}_{n-1,n}\mathbf{p}_n = \prod_{i=m}^{n-1}\mathbf{T}_{i,i+1}. \quad (3.20)$$

U slučaju kada se transformacija vrši od nekog nopomičnog koordinatnog sistema (0 -ti ili bazni sistem) do nekog n -tog sistema (sistem završnog uređaja), tada se matrica transformacije obilježava sa $\mathbf{T}_{0,n}$.

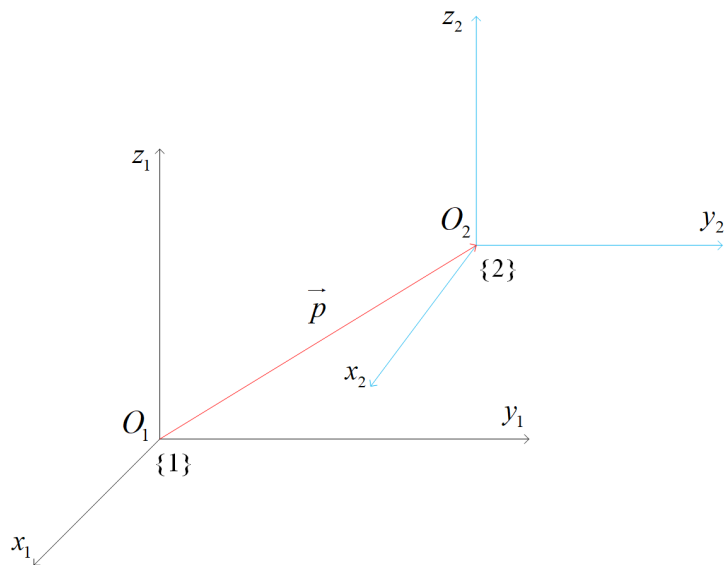
Kako se kod robotskog manipulatora javljaju translatorni i rotacioni stepeni slobode kretanja, neophodno je objasniti transformacije *translacije* i *rotacije*, koje se mogu smatrati *osnovnim transformacijama*.

3.3.3. Transformacija translacije

Ukoliko se posmatra slika 3.12, može se zaključiti da *transformacija translacije* predstavlja pomjerenost koordinatnog početka koordinatnog sistema $\{2\}$ za vektor

$$\vec{p} = [p_x \quad p_y \quad p_z]^T,$$

u odnosu na koordinatni početak sistema $\{1\}$, pri čemu su ose koordinatnih sistema $\{1\}$ i $\{2\}$ paralelne.



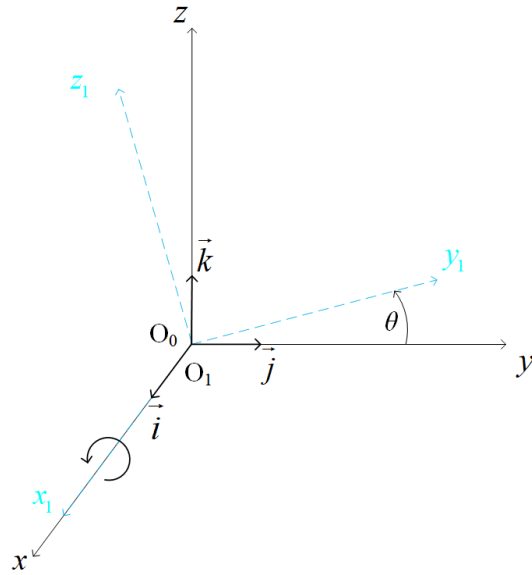
Slika 3.12. Transformacija translacije

Ukoliko se vrši samo translacija, bez rotacije koordinatnog sistema, tada se matrica transformacije \mathbf{T} pretvara u *matricu translacije*. Matrica translacije ima oblik

$$\mathbf{T}_{trans} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.21)$$

3.3.4. Transformacije rotacije

Ako pokretni koordinatni sistem rotira oko jednog jediničnog vektora nepokretnog koordinatnog sistema, dobija se matrica transformacije koordinata koja se naziva *matrica osnovne rotacije*. U trodimenzionalnom prostoru postoje tri osnovne rotacije. Na slici 3.13 prikazana je rotacija oko ose x . Nakon rotacije koja je izvršena oko x -ose koordinatnog sistema za veličinu ugla rotacije θ , koordinatni početak O_1 novog, zarotiranog koordinatnog sistema će se poklapati sa nepokretnim koordinatnim sistemom, dok će ostale ose biti zarotirane za veličinu ugla θ .

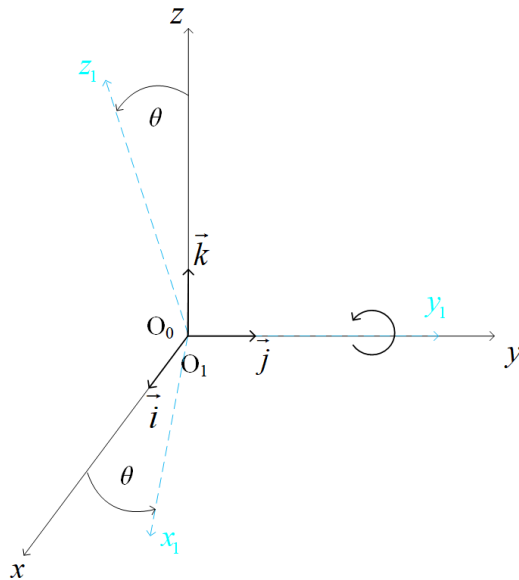


Slika 3.13. Rotacija oko x -ose

Matrica rotacije oko x -ose za ugao θ ima oblik

$$\mathbf{T}_{Rx}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

Rotacija oko y -ose za veličinu ugla θ prikazana je na slici 3.14.

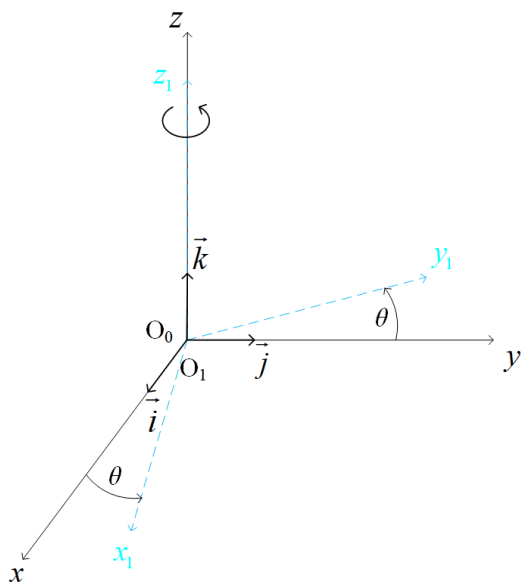


Slika 3.14. Rotacija oko y -ose

Matrica rotacije oko y -ose za ugao θ ima oblik

$$\mathbf{T}_{Ry}(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.23)$$

Rotacija oko z -ose prikazana je na slici 3.15.



Slika 3.15. Rotacija oko z -ose

Matrica rotacije oko z -ose za ugao θ ima oblik

$$\mathbf{T}_{Rz}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.24)$$

Na kraju je važno napomenuti da su transformacije translacije linearne, dok su transformacije rotacije nelinearne jer se za njihovo opisivanje koriste trigonometrijske funkcije.

3.4. Denavit – Hartenbergova notacija

Jedan od osnovnih zadataka inženjera robotike je kinematičko modelovanje robotskog manipulatora. Ovaj zadatak se sastoji od izrade modela koji se može nedvosmisleno predstaviti kontrolnoj jedinici kroz skup podataka. Svrha ovog zadatka je zadavanje manipulacionih instrukcija robota, bez osvrta na dinamiku tereta i samog robota. Najjednostavniji način kinematičkog modelovanja robotskog manipulatora je pomoću koncepta *kinematičkog lanca*.

Kao što je već prethodno objašnjeno u poglavlju 3.1.1, kinematički lanac predstavlja skup krutih tijela, odnosno segmenata, koja su povezana kinematičkim parovima. U slučaju kinematičkog lanca robota koji se koristi u radu, radi se o prostom, otvorenom kinematičkom lancu sa četiri setepena slobode, sa isključivo rotacionim kinematičkim parovima. Prvi segment takvog kinematičkog lanca se naziva *baza manipulatora*, a poslednji segment se naziva *završni organ*. Takođe, ovaj kinematički lanac se sastoji od *binarnih segmenata*, koji sadrže po dva kinematička para (srednji segmenti), i *prostih segmenata*, koji se nalaze na početku i kraju (baza i završni uređaj).

U cilju nedvosmislenog opisivanja arhitekture kinematičkog lanca, tj. relativnog položaja i orijentacije osa susjednih kinematičkih parova, uvodi se *Denavit – Hartenbergova* ili *D – H konvencija*.

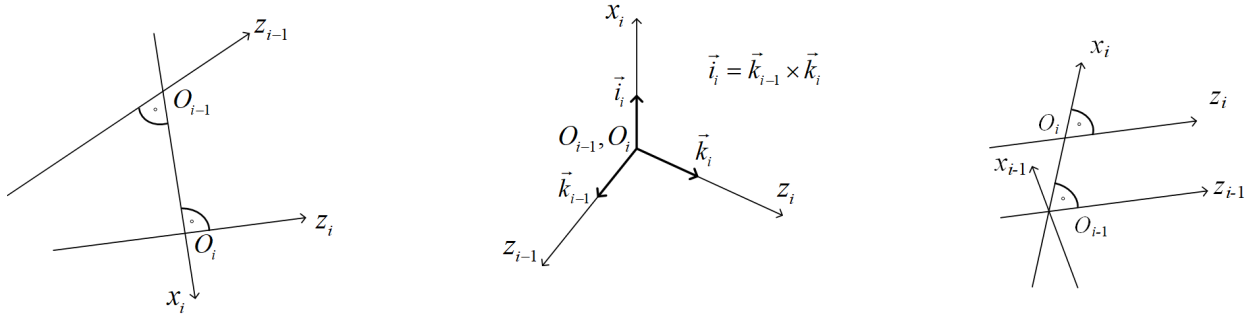
Prema D – H konvenciji, numerisanje segmenata se vrši tako da je baza robota nulti segment, prvi pokretni segment je numerisan brojem 1 i tako do slobodnog kraja, odnosno završnog uređaja, koji je zapravo n -ti segment, pri čemu se i -ti kinematički par definiše kao spajanje $(i-1)$ -vog segmenta i i -tog segmenta. Samim tim, manipulator se sastoji od $n+1$ segmenata i n zglobova.

Da bi se mogle prikazivati određene fizičke i geometrijske veličine vezane za kretanje elemenata robota, potrebno je pridružiti koordinatne sisteme pojedinačnim segmentima. Važno je napomenuti da se koordinatni sistem K_i sa koordinatnim početkom O_i i osama x_i , y_i i z_i pridružuje i -tom segmentu, a nikako $(i+1)$ -vom segmentu!

Za prvih n lokalnih koordinatnih sistema pravila postavljanja su sljedeća:

1. Osa z_{i-1} je osa i -tog kinematičkog para, tj. zgloba oko koje se vrši rotacija za ugao q_i . Postoje dvije mogućnosti za definisanje pozitivnog smjera ose z_{i-1} , jer je osa svakog para linija, a ne vektor.
2. Posmatra se slika 3.16. Osa x_i se definiše kao zajednička normala z_{i-1} i z_i , usmjerena od prethodne ka narednoj osi. Ako se ove dvije ose sijeku, pozitivan smjer ose x_i je nedefinisan i stoga može biti proizvoljno zadan. U ovom slučaju, primijenije se *pravilo desne ruke*, odnosno jedinični vektor \vec{i}_i , koji odgovara osi x_i , se dobija kao vektorski proizvod jediničnih vektora \vec{k}_{i-1} i \vec{k}_i , tj. $\vec{k}_{i-1} \times \vec{k}_i$. Takođe, ako su ose z_{i-1} i z_i paralelne, tada se x_i pozicionira tako da prolazi kroz koordinatni početak $(i-1)$ -vog koordinatnog sistema.
3. Najkraće rastojanje između z_{i-1} i z_i je upravo jedan od D - H parametara, dužina a_i .
4. Koordinata z_{i-1} tačke O_i' , koja predstavlja presjek z_{i-1} sa x_i , predstavlja parametar d_i . Kako ovaj parametar predstavlja koordinatu, isti može biti pozitivan ili negativan. Njegova apsolutna vrijednost predstavlja rastojanje između x_{i-1} i x_i i naziva se *offset* između uzastopnih zajedničkih normala.

5. Ugao između z_{i-1} i z_i se definiše kao α_i i mjeri se u pozitivnim smjeru ose x_i . Ovaj ugao se naziva *ugao uvijanja* između osa susjednih parova.
6. Ugao između x_{i-1} i x_i se definiše kao θ_i i mjeri se u pozitivnom smjeru ose z_{i-1} .

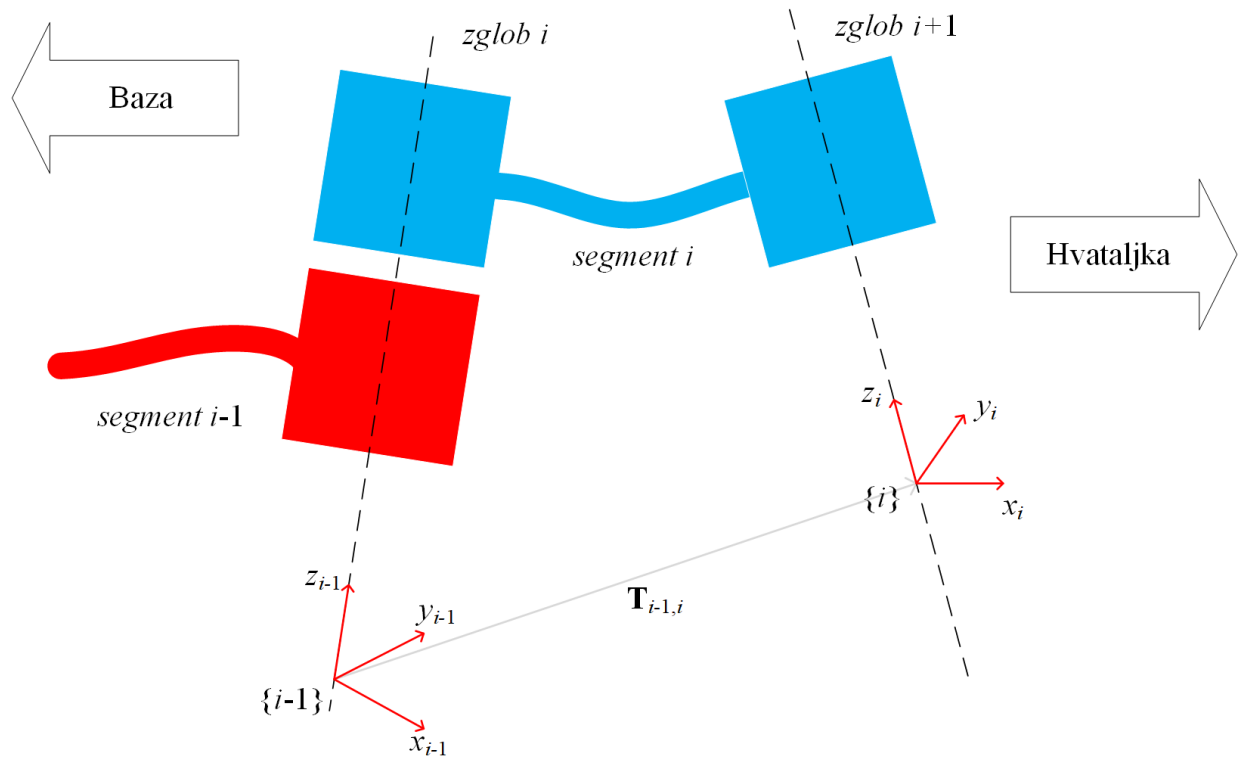


Slika 3.16. Definicija ose x_i

Poslednji, odnosno $(n+1)$ -vi koordinatni sistem se pridružuje samom kraju n -og segmenta, tj. završnog uređaja. Kako kod manipulatora ne postoji $(n+1)$ -vi segment, zadnji koordinatni sistem se definiše tako da najbolje odgovara predviđenom radnom zadatku. Dakle, može se zaključiti da se manipulator sa n stepeni slobode sastoji od $n+1$ segmenata i $n+1$ koordinatnih sistema.

Ključni aspekt D – H notacije je činjenica da se svaki zglobov robota opisuje sa samo četiri parametra. Na slici 3.17 je dat primjer pridruživanja koordinatnih sistema segmentima. Koordinatni sistem za svaki pojedinačni segment se postavlja na samom kraju segmenta i to onom koji je najbliži završnom alatu. Nakon toga, opisuje se pozicija i -tog koordinatnog sistema u odnosu na koordinatni sistem prethodnog zgloba korišćenjem *matrice homogenih transformacija*, čime se dobija relativni položaj i -tog koordinatnog sistema u odnosu na $(i-1)$ -vi koordinatni sistem. Krajnji cilj je dobijanje relativnog položaja krajnjeg koordinatnog sistema, koji je vezan za hvataljku, u odnosu na početni koordinatni sistem, koji je vezan za bazu robota.

Ukoliko je i -ti kinematički par rotacioni, tada su sve navedene veličine $(a_i, b_i, \alpha_i, \theta_i)$ konstantne, osim θ_i , koja je promjenljiva i predstavlja *zglobnu promjenljivu* i -tog kinematičkog para. Ostale veličine (a_i, b_i, α_i) predstavljaju *zglobne parametre* i -tog para. Alternativno, ukoliko je i -ti par translatorski, tada b_i predstavlja *zglobnu promjenljivu*, dok su ostale veličine konstantne. Dakle, može se zaključiti da se za svaki zglobov vezuje tačno jedna promjenljiva i tri konstantna parametra, što znači da se manipulator sa n stepeni slobode može opisati sa n promjenljivih, koje su grupisane u n -dimenzionalni vektor θ i $3n$ konstantnih parametara. Konstantni parametri definišu *arhitekturu manipulatora*, dok promjenljive definišu njegovu *konfiguraciju* ili *položaj*.



Slika 3.17. Pridruživanje koordinatnih sistema

3.4.1. Definisane homogenih matrica transformacija pomoću D – H parametara

Homogene matrice transformacije su matrice reda 4×4 , koje objedinjuju informaciju o rotaciji između dva koordinatna sistema i informaciju o rastojanju između njihovih koordinatnih početaka. Kod D – H notacije, transformacija iz jednog koordinatnog sistema u drugi se opisuje pomoću homogene matrice transformacija, koja se najčešće označava slovom \mathbf{T} i koja se sastoji od četiri elementarne transformacije – dvije rotacije i dvije translacije. Dakle, transformacija iz $(i-1)$ -og u i -ti sistem se može zapisati kao

$$\mathbf{T}_{i-1,i} = \mathbf{T}_{Rz_{i-1}}(q_i) \mathbf{T}_{trans,z_{i-1}}(d_i) \mathbf{T}_{trans,x_{i-1}}(a_i) \mathbf{T}_{Rx_i}(\alpha_i). \quad (3.25)$$

Objašnjenje jednačine (3.25) je sljedeće:

- najprije se vrši rotacija oko ose z_{i-1} za vrijednost ugla q_i (u slučaju da se radi o translatorskom zglobo, rotacija oko ose z_{i-1} bi se izvršila za vrijednost ugla θ_i),
- zatim se vrši translacija duž ose z_{i-1} za veličinu d_i (ukoliko se radi o translatorskom zglobo, za veličinu unutrašnje koordinate q_i),
- u trećem koraku se vrši translacije duž ose $x_{i-1} \equiv x_i$ za vrijednost a_i ,
- poslednji korak je rotacija oko ose x_i za vrijednost ugla α_i .

Dakle, pomjeranje od $(i-1)$ -vog do i -tog koordinatnog sistema se vrši primjenom četiri elementarne transformacije, dvije translacije i dvije rotacije. Pitanje koje se nameće je kako se položaj u prostoru, koji ima dvije komponente (pozicija i orijentacija) koje se opisuju sa po tri parametra, može opisati sa samo četiri parametra. Odgovor na ovo pitanje leži u ograničenjima koja nameće D – H konvencija. Prva ograničenje je da x osa i -tog sistema siječe osu z $(i-1)$ -vog sistema, a drugo ograničenje je da je osa x_i normalna na osu z_{i-1} . Nametanje pomenutih ograničenja omogućava da se, iako relativni položaj u prostoru ima šest stepeni slobode, isti opiše sa četiri parametra [23].

Niz rotacija i translacija predstavljen izrazom (3.25) se može zapisati kao proizvod sljedećih homogenih matrica transformacije (slučaj rotacionog kinematičkog para):

$$\mathbf{T}_{i-1,i} = \begin{bmatrix} \cos q_i & -\sin q_i & 0 & 0 \\ \sin q_i & \cos q_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.26)$$

Kada se matrice na desnoj strani jednačine (3.26) izmnože, dobija se homogena matrica transformacije između i -tog i $(i-1)$ -vog koordinatnog sistema, za rotacioni kinematički par:

$$\mathbf{T}_{i-1,i} = \begin{bmatrix} \cos q_i & -\sin q_i \cos \alpha_i & \sin q_i \sin \alpha_i & a_i \cos q_i \\ \sin q_i & \cos q_i \cos \alpha_i & -\cos q_i \sin \alpha_i & a_i \sin q_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.27)$$

Na osnovu jednačine (3.27) se može zaključiti da je homogena matrica transformacije jedino funkcija unutrašnjih koordinata. Da bi se analitički oblik transformacione matrice $\mathbf{T}_{i-1,i}$ pojednostavio, u praksi je uobičajeno da manipulatori izrađuju tako da uglovi α_i uzimaju vrijednosti 0° (ukoliko su ose zglobova paralelne) ili $\pm 90^\circ$ ili (ako su ose međusobno normalne).

Kada se izračunaju matrice homogenih transformacija između susjednih segmenata, može se izračunati i matrica transformacije između nepokretnog koordinatnog sistema u bazi robota i koordinatnog sistema hvataljke:

$$\mathbf{T}_{0,n} = \mathbf{T}_{0,1} \cdot \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,3} \cdot \dots \cdot \mathbf{T}_{n-2,n-1} \cdot \mathbf{T}_{n-1,n}. \quad (3.28)$$

Submatrica matrice $\mathbf{T}_{0,n}$, koja sadrži njene prve tri vrste i kolone, predstavlja matricu rotacije $\mathbf{R}_{0,n}$ između koordinatnog sistema hvataljke i nepokretnog koordinatnog sistema u bazi robota, a vektor koji sadrži četvrtu kolonu i prve tri vrste matrice $\mathbf{T}_{0,n}$ je vektor položaja hvataljke robota u odnosu na nepokretni koordinatni sistem, odnosno:

$$\mathbf{T}_{0,n} = \left[\begin{array}{ccc|c} & & & p_x \\ & \mathbf{R}_{0,n} & & p_y \\ & & & p_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right]. \quad (3.29)$$

3.4.2. Određivanje orijentacije završnog uređaja

Orijentacija završnog uređaja, tj. hvataljke se može ostvariti kompozicijom homogenih rotacija oko nepokretnog koordinatnog sistema prema sljedećem rasporedu:

- rotacija oko z ose za vrijednost ugla θ ,
- rotacija oko y ose za vrijednost ugla φ i
- rotacija oko x ose za vrijednost ugla ψ .

Ovaj niz rotacija se u matičnom obliku može predstaviti kao

$$\mathbf{R}_{0,n} = \mathbf{R}_z(\theta) \mathbf{R}_y(\varphi) \mathbf{R}_x(\psi), \quad (3.30)$$

odnosno

$$\mathbf{R}_{0,n} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix}. \quad (3.31)$$

Kada se izmnože matrice u jednačini (3.31), dobija se:

$$\mathbf{R}_{0,n} = \begin{bmatrix} \cos \theta \cos \varphi & \cos \theta \sin \varphi \sin \psi - \sin \theta \cos \psi & \cos \theta \sin \varphi \cos \psi + \sin \theta \sin \psi \\ \sin \theta \cos \varphi & \sin \theta \sin \varphi \sin \psi + \cos \theta \cos \psi & \sin \theta \sin \varphi \cos \psi - \cos \theta \sin \psi \\ -\sin \varphi & \cos \varphi \sin \psi & \cos \varphi \cos \psi \end{bmatrix}. \quad (3.32)$$

Članovi matrice u (3.32) u suštini predstavljaju koordinate jediničnih vektora koordinatnog sistema hvataljke u početnom koordinatnom sistemu vezanom za bazu. Jedinični vektori koordinatnog sistema hvataljke su: vektor normale \vec{n} , vektor orijentacije \vec{o} i vektor djelovanja \vec{a} [21]. Dalje, matrica rotacije $\mathbf{R}_{0,n}$ se može zapisati i u obliku:

$$\mathbf{R}_{0,n} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}. \quad (3.33)$$

Ukoliko se izjednače (3.32) i (3.33), dobija se sljedeći sistem jednačina:

$$\begin{cases} n_x = \cos \theta \cos \varphi \\ n_y = \sin \theta \cos \varphi, \\ n_z = -\sin \varphi \end{cases}$$

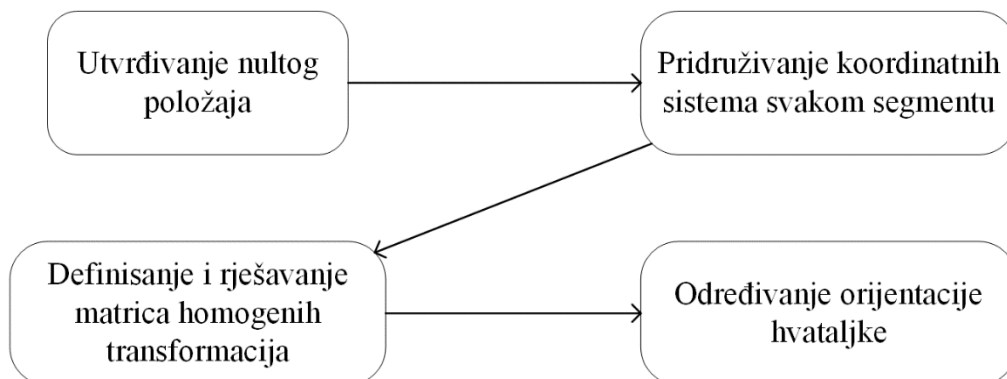
$$\begin{cases} o_x = \cos \theta \sin \varphi \sin \psi - \sin \theta \cos \psi \\ o_y = \sin \theta \sin \varphi \sin \psi + \cos \theta \cos \psi, \\ o_z = \cos \varphi \sin \psi \end{cases} \quad (3.34)$$

$$\begin{cases} a_x = \cos \theta \sin \varphi \cos \psi + \sin \theta \sin \psi \\ a_y = \sin \theta \sin \varphi \cos \psi - \cos \theta \sin \psi. \\ a_z = \cos \varphi \cos \psi \end{cases}$$

3.5. Direktni kinematički problem

3.5.1. Algoritam rješavanja direktnog kinematičkog problema

Postupak određivanja vektora spoljašnjih koordinata na osnovu zadatog vektora unutrašnjih koordinata je poznat kao *direktni kinematički problem*. Rješavanje direktnog kinematičkog problema se može izvršiti prema algoritmu čiji je dijagram dat na slici 3.18.



Slika 3.18. Algoritam rješavanja direktnog kinematičkog problema

Da bi se riješio direktni kinematički problem, najprije je potrebno izvršiti *utvrđivanje nultog položaja undustrijskog robota*, odnosno potrebno je ustanoviti simboličku shemu robota u odnosu na izabrani nepokretni koordinatni sistem. Simbolička shema treba da sadrži sve dimenzije od značaja datog robota. Utvrđeni položaj robota, za koji sve unutrašnje koordinate imaju vrijednosti jednake nuli ($q_i = 0$), predstavlja *nulti položaj robota*.

Sljedeći korak u rješavanju direktnog kinematičkog problema je *pridruživanje koordinatnih sistema pojedinim segmentima manipulatora*. To pridruživanje koordinatnih sistema se vrši na osnovu metode Denavit – Hartenberg.

Treći korak predstavlja *definisanje i rješavanje matrica homogenih transformacija* između pridruženih koordinatnih sistema. Definisanje matrica homogenih transformacija se vrši pomoću D – H parametara, dok rješavanje matrica homogenih transformacija podrazumijeva njihovu kompoziciju (množenje) kako bi se dobila transformacija iz koordinatnog sistema hvataljke u nepokretni, bazni koordinatni sistem (poglavlje 3.4.1).

Poslednji korak podrazumijeva *određivanje orijentacije hvataljke* u odnosu na bazni koordinatni sistem, tj. određivanje uglova između osa koordinatnog sistema hvataljke i fiksnog koordinatnog sistema iz rezultatne matrice homogenih transformacija (poglavlje 3.4.2).

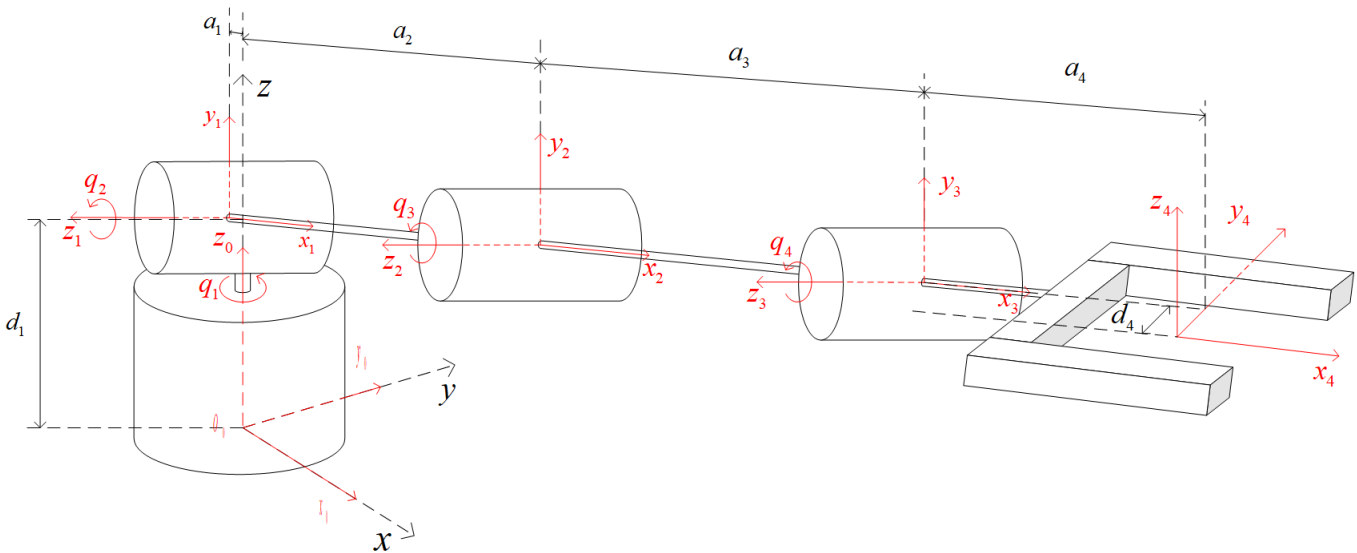
3.5.2. Rješenje direktnog kinematičkog problema za SPT – 3000

SPT – 3000 manipulator ima četiri stepena slobode kretanja. Vodeći se koracima opšteg algoritma rješavanja direktnog kinematičkog problema, nacrtan je nulti položaj robota ($q_i=0$) i pridruženi su koordinatni sistemi svakog segmentu, pri čemu je prvi koordinatni sistem postavljen u podnožju (bazi) robota, dok je poslednji koordinatni sistem postavljen u centru hvataljke (slika 3.19). Ukupan broj koordinatnih sistema je šest – bazni, nepokretni sistem u kojem se traži položaj hvataljke, četiri sistema koja opisuju zglobove manipulatora i krajnji koordinatni sistem koji je smješten u centru hvataljke.

D – H parametri za SPT – 3000 su dati u tabeli 3.1. Na osnovu parametara iz tabele moguće je izračunati matrice homogenih transformacija po D – H metodi.

Zglob(i)	q_i	α_i (°)	a_i (mm)	θ_i	d_i (mm)	$\sin(\alpha_i)$	$\cos(\alpha_i)$
1	q_1	90	-14	0	99,4	1	0
2	q_2	0	134	0	0	0	1
3	q_3	0	132,5	0	0	0	1
4	q_4	-90	118,5	0	22,08	-1	0

Tabela 3.1. D – H parametri za SPT – 3000 manipulator



Slika 3.19. Kinematički model SPT – 3000 manipulatora

Matrice homogenih transformacija date u tabeli 3.1, za kinematički model robota prikazan na slici 3.19, se po Denavit – Hartenbergovoj metodi računaju na osnovu izraza (3.27) i imaju sljedeći oblik:

$$\mathbf{T}_{0,1} = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & a_1 \cos q_1 \\ \sin q_1 & 0 & -\cos q_1 & a_1 \sin q_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \left\{ \begin{array}{l} \sin \alpha_1 = 1 \\ \cos \alpha_1 = 0 \\ a_1 = -14 \text{ mm} \\ d_1 = 99,4 \text{ mm} \end{array} \right. \quad (3.35)$$

$$\mathbf{T}_{1,2} = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & a_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & a_2 \sin q_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{cases} \sin \alpha_2 = 0 \\ \cos \alpha_2 = 1 \\ a_2 = 134 \text{ mm} \\ d_2 = 0 \text{ mm} \end{cases} \quad (3.36)$$

$$\mathbf{T}_{2,3} = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & a_3 \cos q_3 \\ \sin q_3 & \cos q_3 & 0 & a_3 \sin q_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{cases} \sin \alpha_3 = 0 \\ \cos \alpha_3 = 1 \\ a_3 = 132,5 \text{ mm} \\ d_3 = 0 \text{ mm} \end{cases} \quad (3.37)$$

$$\mathbf{T}_{3,4} = \begin{bmatrix} \cos q_4 & 0 & -\sin q_4 & a_4 \cos q_4 \\ \sin q_4 & 0 & \cos q_4 & a_4 \sin q_4 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{cases} \sin \alpha_4 = -1 \\ \cos \alpha_4 = 0 \\ a_4 = 118,5 \text{ mm} \\ d_4 = 22,08 \text{ mm} \end{cases} \quad (3.38)$$

Množenjem matrica (3.35 – 3.38), dobija se matrica:

$$\mathbf{T}_{0,4} = \mathbf{T}_{0,1} \cdot \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,3} \cdot \mathbf{T}_{3,4}, \quad (3.39)$$

koja predstavlja kretanje robota u radnom prostoru, tj. odnos između koordinatnog sistema hvataljke i baznog (početnog) koordinatnog sistema. Nakon uvrštavanja brojnih vrijednosti i množenja matrica u (3.39), matrica $\mathbf{T}_{0,4}$ poprima sljedeći oblik:

$$\mathbf{T}_{0,4} = \begin{bmatrix} 1 & 0 & 0 & 371 \\ 0 & 1 & 0 & -22.8 \\ 0 & 0 & 1 & 99.4 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.40)$$

Kako matricno opisivanje kretanja hvataljke ima sljedeći oblik:

$$\mathbf{T}_{0,4} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} = \left[\begin{array}{c|c} \begin{bmatrix} \textit{matrica} \\ \textit{orijentacije} \\ 3 \times 3 \end{bmatrix} & \begin{bmatrix} \textit{vektor} \\ \textit{pozicije} \\ 3 \times 1 \end{bmatrix} \\ \hline 0 & 0 & 0 & 1 \end{array} \right], \quad (3.41)$$

izjednačavanjem članova matrica (3.40) i (3.41) dobijaju se:

- jedinični vektori orijentacije hvataljke u odnosu na bazni koordinatni sistem:

$$\begin{cases} \vec{n} = [n_x \ n_y \ n_z]^T = [1 \ 0 \ 0]^T \\ \vec{o} = [o_x \ o_y \ o_z]^T = [0 \ 1 \ 0]^T, \\ \vec{a} = [a_x \ a_y \ a_z]^T = [0 \ 0 \ 1]^T \end{cases} \quad (3.42)$$

- vektor pozicije hvataljke u odnosu na bazni koordinatni sistem:

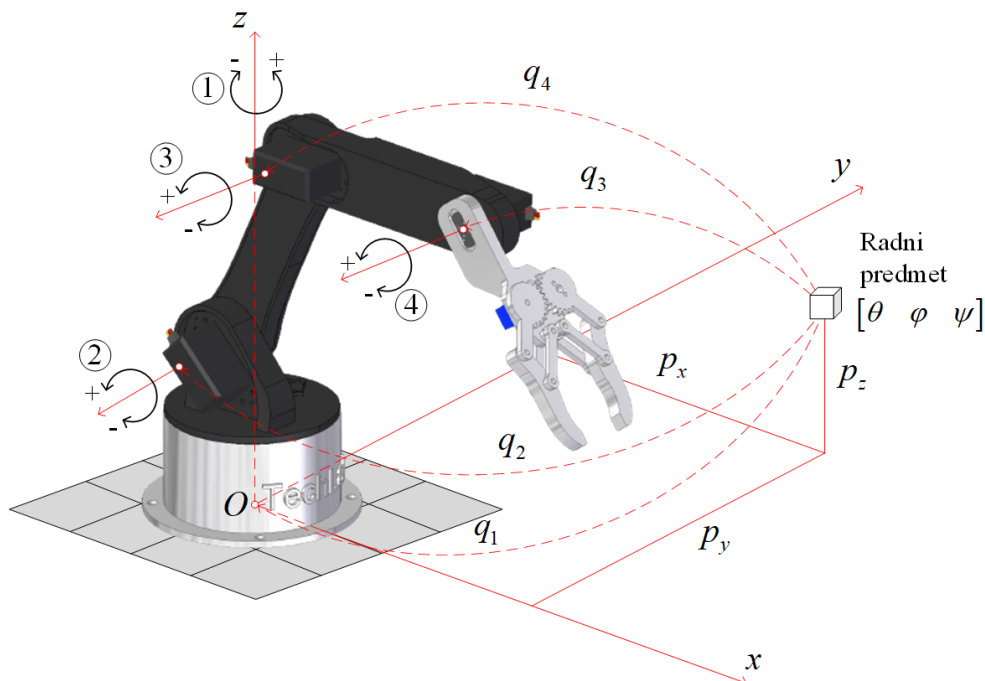
$$\vec{p} = [p_x \ p_y \ p_z]^T = [371 \ -22.8 \ 99.4]^T. \quad (3.43)$$

Izrazi (3.42) i (3.43) predstavljaju rješenje direktnog kinematičkog problema za nulti položaj SPT – 3000 manipulatora. Sa ovim je u potpunosti definisana direktna kinematika manipulatora, što znači da se za date vrijednosti unutrašnjih koordinata, koje odgovaraju mogućnostima i strukturi manipulatora, mogu odrediti pozicija i orijentacija završnog alata u prostoru spolašnjih koordinata.

3.6. Inverzna kinematika robotskog manipulatora

U u prethodnom poglavlju pokazano je kako odrediti položaj završnog organa za date koordinate zglobova. Međutim, problem koji je od većeg praktičnog interesa je inverzni kinematički problem. Dakle, za poznati položaj objekta u radnom prostoru robota, potrebno je odrediti koordinate zglobova koje su robotu potrebne da bi dostigao taj položaj (slika 3.20). Inverzni kinematički problem se u matematičkoj formi može zapisati kao

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{r}). \quad (3.44)$$



Slika 3.20. Inverzni kinematički problem

Funkcija f , koja povezuje spoljašnje i unutrašnje koordinate je nelinearna, vektorska trigonometrijska funkcija. To ima za posledicu da IKP nije jednostavno riješiti kao direktni problem, jer se rješavanje inverznog problema svodi na rješavanje sistema nelinearnih jednačina. Iz tog razloga, pri rješavanju inverznog kinematičkog problema javljaju se sljedeći problemi:

- Kako su jednačine nelinearne, nije uvijek moguće naći *rješenje u zatvorenoj formi*.
- Može se javiti *problem postojanja više rješenja*.
- Dobijena rješenja mogu biti i *neprihvatljiva*, usljed ograničenja koja nameće kinematička struktura manipulatora.

Rješenje u zatvorenoj formi ili analitičko rješenje predstavlja matematički izraz koji se formira od konstanti, promjenljivih i konačnog broja standardnih matematičkih operacija i funkcija. Generalno, rješenja u zatvorenoj formi se mogu naći samo za manipulatore sa šest stepeni slobode, koji zadovoljavaju sljedeće uslove [23]:

- ose tri poslednja zgloba se sijeku u jednoj tački, tj. šaka manipulatora ima *sfernu konfiguraciju*,
- ose tri uzastopne zgloba su paralelne.

Sa druge strane, problem postojanja više rješenja ne zavisi samo od broja stepeni slobode, već i od broja nenultih $D - H$ parametara. Generalno, što je veći broj $D - H$ parametara različitih od nule, veći je i broj prihvatljivih rješenja. Prihvatljiva rješenja su ona rješenja koja zadovoljavaju ograničenja koja nameće kinematička struktura manipulatora, kao i ostala ograničenja (radni prostor, mehanički limiti zglobova, itd.).

U slučajevima kada je manipulator *kinematički redundantan* [23], dobija se beskonačan broj rješenja. Razlog za to je što kinematički redundantan manipulator posjeduje više stepeni slobode nego što je striktno potrebno za izvršenje datog zadatka. U tim slučajevima, da bi riješio inverzni problem, neophodno je nametnuti broj ograničenja u kretanju manipulatora koji je jednak razlici između broja stepeni slobode manipulatora i dimenzije radnog prostora.

U slučajevima kada rješenje inverznog problema postoji, ali se ne može zapisati u zatvorenoj formi, koriste se numeričke metode. U nastavku glave biće izloženi analitički i numerički pristupi rješavanju inverznog kinematičkog problema, uz navođenje njihovih prednosti i mana.

3.6.1. Analitičko rješenje inverznog kinematičkog problema

Analitičko određivanje funkcije f^{-1} je složen matematički problem za koji ne postoji rješenje u opštem obliku za proizvoljnu strukturu robota. Odnosno, analitičko rješavanje inverznog kinematičkog problema je moguće samo za određene konfiguracije robota.

Prednosti analitičkog rješenja u odnosu na numeričko rješenje [21] su:

- računanje traje znatno kraće, jer je potreban manji broj matematičkih operacija da bi se došlo do rješenja,
- svako dobijeno analitičko rješenje je tačno, odnosno ne dolazi do nagomilavanja greške kao u slučaju numeričkog rješavanja,
- singularitete, tj. tačke prostora u kojima završni alat robota gubi jedan ili više stepeni slobode, je moguće predvidjeti i uzeti u obzir prilikom analitičkog rješavanja inverznog problema.

Matrica transformacije koordinata iz n -tog u m -ti koordinatni sistem, pri čemu je $m < n$, definisana je izrazom (3.20). Prema tome, analitičko rješenje inverznog kinematičkog problema se može naći na sljedeći način:

$$\begin{cases} \mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,n} = \mathbf{T}_{1,n} \\ \mathbf{T}_{1,2}^{-1} \cdot \mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,n} = \mathbf{T}_{2,n} \\ \vdots \\ \mathbf{T}_{n-2,n-1}^{-1} \cdot \dots \cdot \mathbf{T}_{2,3}^{-1} \cdot \mathbf{T}_{1,2}^{-1} \cdot \mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,n} = \mathbf{T}_{n-1,n} \end{cases} \quad (3.45)$$

Drugi način analitičkog rješavanja inverznog kinematičkog problema je podjela inverznog problema na dva jednostavnija problema, odnosno određivanje rješenja za minimalnu konfiguraciju robota i određivanje rješenja za završni uređaj robota. Matrični zapis podjele inverznog kinematičkog problema je sljedeći:

$$\mathbf{T}_{0,n} = \mathbf{T}_{0,R} \cdot \mathbf{T}_{R,H}, \quad (3.46)$$

pri čemu se pojedine matrice:

- $\mathbf{T}_{0,n}$ – ukupno kretanje robota,
- $\mathbf{T}_{0,R}$ – kretanje ruke,
- $\mathbf{T}_{R,H}$ – kretanje završnog uređaja.

Pristup podjele inverznog problema je moguć u slučaju manipulatora sa šest stepeni slobode i sfernom konfiguracijom šake, koji se veoma često mogu sresti u industriji. Sferna konfiguracija šake omogućava promjenu orijentacije završnog organa, bez promjene njegove pozicije. U tom slučaju se na osnovu zadate pozicije i orijentacije hvataljke može odrediti položaj vrha minimalne konfiguracije (ruke) manipulatora.

3.6.2. Numeričko (iterativno) rješenje inverznog kinematičkog problema

Rješavanje skupa nelinearnih jednačina se može vršiti primjenom različitih numeričkih metoda: Njutnova metoda, Brojdenova metoda, gradijentna metoda, itd. Ove metode su opšte i samim tim ne zavise od kinematičke strukture robota. Upravo to je i glavna prednost numeričkih metoda, jer pružaju mogućnost primjene jednog programa na različite konfiguracije robota. Međutim, prilikom izvršavanju jednog takvog programa mogu se javiti problemi kao što su neprepoznavanja singularnih položaja robota i divergencije postupka računanja.

Najčešće korišćena metoda rješavanja inverznog problema je Njutnova metoda [33], kod koje je neophodno odrediti Jakobijevu matricu $\mathbf{J}(\mathbf{q})$, kao i njenu inverznu matricu \mathbf{J}^{-1} . Nedostaci Njutnove metode su:

- relativno veliko vrijeme računanja inverzne matrice,
- neophodno definisanje početnog rješenja, koje mora biti blisko stvarnom rješenju,
- pri pojavi singulariteta neophodno je izvršiti modifikaciju trajektorije ili je zadati u prostoru koordinata zglobova (u okolini singularnih tačaka nije moguće izračunati \mathbf{J}^{-1}).

Neka se posmatra funkcija $\mathbf{F}(\mathbf{q})$:

$$\mathbf{F}(\mathbf{q}) = \mathbf{f}(\mathbf{q}) - \mathbf{r},$$

gdje je $\mathbf{r} \in \mathbb{R}^m$ vektor spoljašnjih koordinata, $\mathbf{q} \in \mathbb{R}^n$ vektor unutrašnjih koordinata i $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ neprekidna, vektorska diferencijabilna funkcija koja vrši preslikavanje unutrašnjih koordinata u spoljašnje.

Cilj je odrediti vektor unutrašnjih koordinata \mathbf{q}_k za čije je vrijednosti funkcija $\mathbf{F}(\mathbf{q})$ jednaka nuli, odnosno:

$$\mathbf{F}(\mathbf{q}_k) = \mathbf{f}(\mathbf{q}_k) - \mathbf{r} = 0. \quad (3.47)$$

Bitno je napomenuti da približno, tj. početno rješenje \mathbf{q}_p se mora nalaziti u blizini rješenja \mathbf{q}_k , kako bi algoritam konvergirao. Kada se funkcija $\mathbf{f}(\mathbf{q})$ razvije u Tejlorov red u okolini tačke \mathbf{q}_k , uz zadržavanje samo prva dva člana reda, i uvrsti u jednačinu (3.59), dobija se sljedeće:

$$\mathbf{f}(\mathbf{q}_k) - \mathbf{r} + \mathbf{J}(\mathbf{q}_k)(\mathbf{q} - \mathbf{q}_k) = 0, \quad (3.48)$$

Jakobijeva matrica parcijalnih izvoda funkcije $\mathbf{f}(\mathbf{q})$ u (3.60) ima oblik:

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix}. \quad (3.49)$$

Matrica (3.49) povezuje brzine promjene unutrašnjih koordinata $\dot{\mathbf{q}} = [\dot{q}_1 \quad \dot{q}_2 \quad \cdots \quad \dot{q}_n]^T$ sa brzinama promjena spoljašnjih koordinata $\dot{\mathbf{p}} = [\dot{p}_x \quad \dot{p}_y \quad \dot{p}_z \quad \dot{\theta} \quad \dot{\phi} \quad \dot{\psi}]^T$ i ima veliki značaj u sintezi kretanja robota.

Ukoliko se iz (3.48) izrazi vektor unutrašnjih koordinata dobija se:

$$\mathbf{q} = \mathbf{q}_k - \mathbf{J}^{-1}(\mathbf{q}_k)(\mathbf{f}(\mathbf{q}_k) - \mathbf{r}) = \mathbf{q}_k + \mathbf{J}^{-1}(\mathbf{q}_k) \Delta \mathbf{r}_k, \quad (3.50)$$

gdje je:

- $\mathbf{J}^{-1}(\mathbf{q}_k)$ invertovana matrica Jakobijana \mathbf{J} ,
- $\Delta \mathbf{r}_k = \mathbf{r} - \mathbf{f}(\mathbf{q}_k)$ vektor razlike zadatih spoljašnjih koordinata u odnosu na spoljašnje koordinate koje odgovaraju približnom rješenju \mathbf{q}_k .

Ukoliko se u (3.50) usvoji da je $\mathbf{q}_k = \mathbf{q}_{k+1}$, dobija se *iterativni postupak*. U prvoj iteraciji \mathbf{q}_k uzima vrijednost početnog rješenja \mathbf{q}_p . Uslov za završetak iterativnog postupka je $\|\Delta \mathbf{r}_k\| < \varepsilon$, gdje je ε mala pozitivna konstanta. Njutnov postupak ima kvadratnu konvergenciju, što znači da postupak brzo konvergira, pri čemu ovaj postupak daje samo jedno rješenje i to ono koje se nalazi u blizini početnog rješenja. Postojanje inverzne Jakobijeve matrice zavisi od njene determinante, tj. ukoliko je $\det(\mathbf{J}(\mathbf{q})) \neq 0$, inverzna matrica postoji. U slučajevima kada u nekoj tački $\det(\mathbf{J}(\mathbf{q})) = 0$ – singularitet, ili kada matrica \mathbf{J} nije kvadratna, nije moguće odrediti inverzni Jakobijan. Tada se računa pseudoinverzni Jakobijan [23].

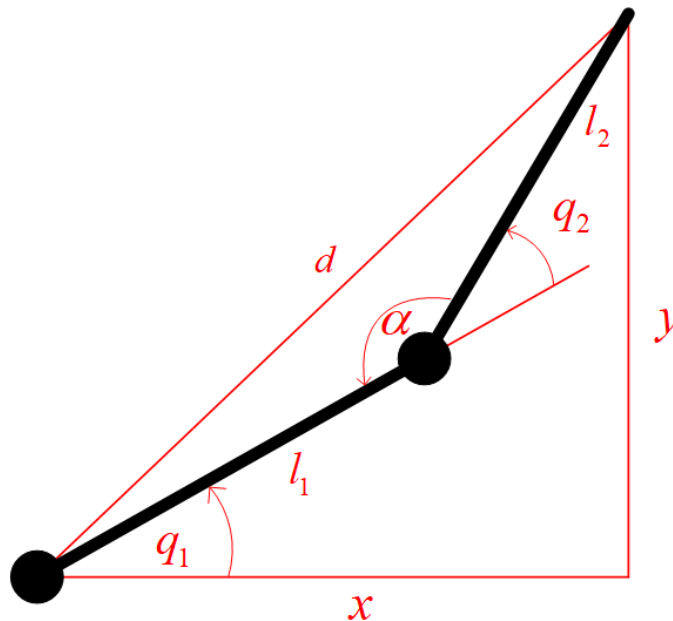
3.7. Rješavanje inverznog kinematičkog problema

Matematička formulacija inverznog kinematičkog problema je sljedeća: Za zadati položaj završnog uređaja $\mathbf{r} \in \mathbb{R}^m$ odrediti skup vrijednosti koordinata zglobova $\mathbf{q} \in \mathbb{R}^n$ koji zadovoljava direktnu kinematiku $\mathbf{r} = \mathbf{f}(\mathbf{q})$. Drugim riječima, potrebno je riješiti sistem $\mathbf{q} = \mathbf{f}^{-1}(\mathbf{r})$. U cilju boljeg razumijevanja problematike rješavanja inverznog problema, u nastavku će najprije biti dat jednostavan primjer rješavanja istog.

3.7.1. Planarni manipulator sa dva stepena slobode

Rješava se inverzni problem planarni manipulator koji sadrži dva rotaciona kinematička para, koji je dat na slici 3.21. Neka su koordinate krajnje tačke manipulatora (x, y) . Rješavanje inverznog zadatka podrazumijeva određivanje koordinata zglobova koje zadovoljavaju zadate koordinate u radnom prostoru, odnosno:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \mathbf{f}^{-1}(\mathbf{r}). \quad (3.51)$$



Slika 3.21. Planarni manipulator sa dva stepena slobode

Posmatra se slika 3.21. Na osnovu slike se mogu zapisati sljedeće jednakosti:

$$\begin{aligned} d^2 &= x^2 + y^2, \\ d^2 &= l_1^2 + l_2^2 - 2l_1l_2 \cos \alpha. \end{aligned} \quad (3.52)$$

Dalje, na osnovu (3.52) može se pisati:

$$\cos \alpha = \frac{l_1^2 + l_2^2 - d^2}{2l_1l_2}. \quad (3.53)$$

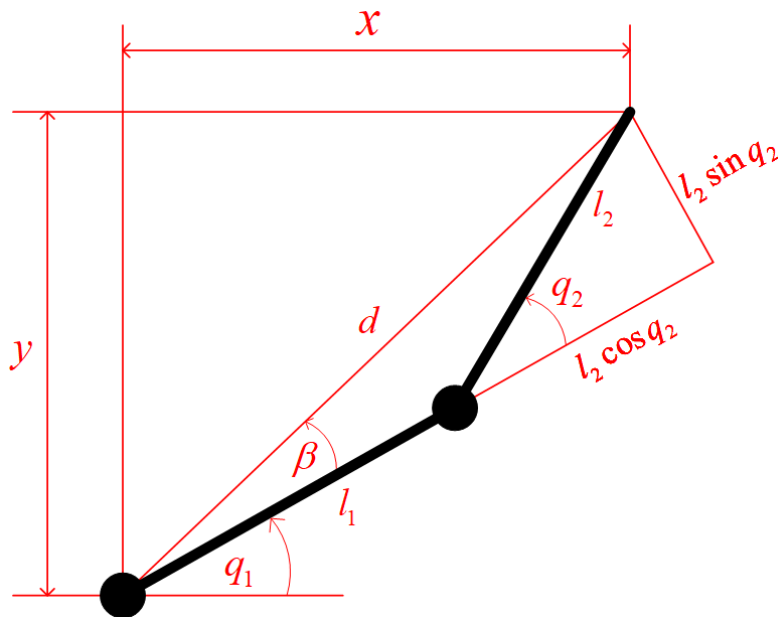
Dodatno, kako je $q_2 = \pi - \alpha$, može se pisati:

$$\begin{aligned} \cos q_2 &= \cos(\pi - \alpha), \\ \cos q_2 &= \cos \pi \cos \alpha + \sin \pi \sin \alpha = -\cos \alpha, \\ \cos q_2 &= \frac{d^2 - l_1^2 - l_2^2}{2l_1l_2} = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}. \end{aligned} \quad (3.54)$$

Konačno, dobija se izraz za koordinatu drugog zgloba:

$$q_2 = \pm \arccos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right). \quad (3.55)$$

Iz izraza (3.55) se može uočiti da postoje dva rješenja za ugao drugog zgloba, tj. da postoje dvije konfiguracije koje zadovoljavaju željeni položaj manipulatora. Pozitivno rješenje iz (3.55) daje „lakat dolje” konfiguraciju – $q_2 \in (0, \pi/2)$, dok negativno rješenje odgovara „lakat gore” konfiguraciji – $q_2 \in (-\pi/2, 0)$.



Slika 3.22. Uz računanje ugaone koordinate prvog zgloba robota

Da bi se izračunala koordinata q_1 , posmatra se slika 3.22. Na osnovu slike može se pisati:

$$\begin{aligned} \tan(q_1 + \beta) &= \frac{y}{x} \rightarrow q_1 = \arctan\left(\frac{y}{x}\right) - \beta, \\ \beta &= \arctan\left(\frac{l_2 \sin q_2}{l_1 + l_2 \cos q_2}\right). \end{aligned} \quad (3.56)$$

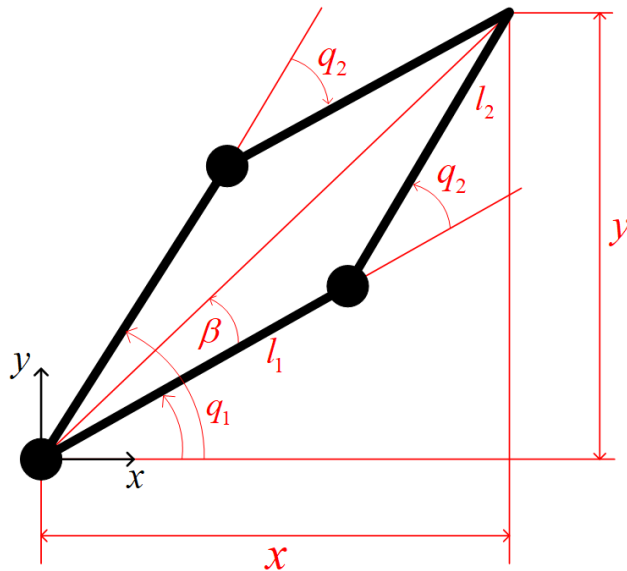
Iz (3.56) se može zapisati da je ugaona koordinata prvog zgloba:

$$q_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{l_2 \sin q_2}{l_1 + l_2 \cos q_2}\right). \quad (3.57)$$

Kako za ugao q_2 postoje dva prihvatljiva rješenja koja odgovaraju spoljašnjim koordinatama, tako se i za q_1 dobijaju dva rješenja koja odgovaraju „lakat dolje” i „lakat gore” konfiguracijama.

Slika 3.23 prikazuje dvije prihvatljive konfiguracije planarnog manipulatora sa dva stepena slobode koje odgovaraju zadatim spoljašnjim koordinatama. Dakle, postoje dvije različite konfiguracije zglobova $\mathbf{q} \in \mathbb{R}^2$ koje odgovaraju zatom položaju završnog uređaja $\mathbf{r} \in \mathbb{R}^2$, odnosno:

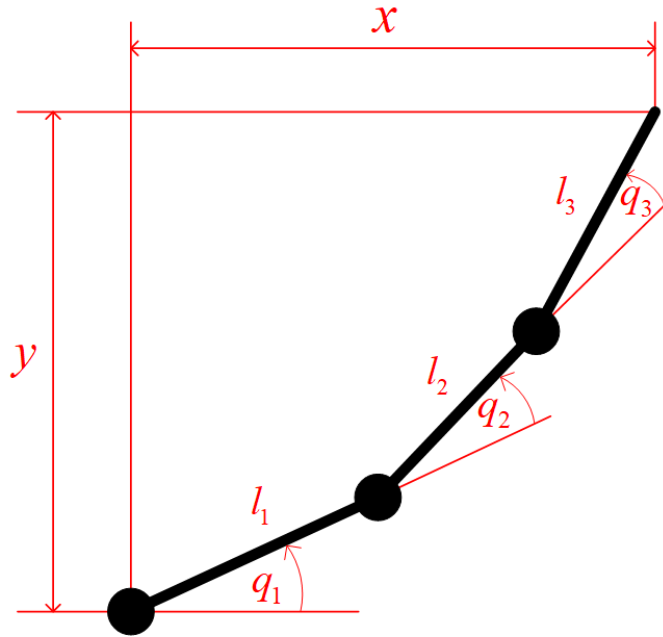
$$\mathbf{q} = \begin{cases} \left[\begin{array}{c} \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{l_2 \sin q_2}{l_1 + l_2 \cos q_2}\right) \\ \arccos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \end{array} \right] \\ \left[\begin{array}{c} \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{l_2 \sin q_2}{l_1 + l_2 \cos q_2}\right) \\ -\arccos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \end{array} \right] \end{cases}. \quad (3.58)$$



Slika 3.23. Rješenja inverznog kinematičkog problema

Ovaj jednostavan primjer pokazuje da, za otvorene kinematičke lance, inverzni kinematički problem može imati više rješenja, što je u suprotnosti sa direktnom kinematikom, gdje je za date vrijednosti koordinata zglobova postoji jedinstveni položaj završnog organa. Dodatno, planarni otvoreni lanac sa tri stepena slobode daje beskonačan broj rješenja za bilo koju tačku (x, y) koja se

nalazi unutar radnog prostora (slika 3.24), jer je broj stepeni slobode ($n = 3$) veći od broja stepeni slobode potrebnog za izvršavanje zadatka (kinematički redundantan manipulator).



Slika 3.24. Planarni manipulator sa tri stepena slobode

3.7.2. Rješenje inverznog kinematičkog problema za SPT – 3000

Analitičko rješenje inverznog kinematičkog problema se nalazi na osnovu opšte jednačine:

$$\mathbf{T}_{n-2,n-1}^{-1} \cdot \dots \cdot \mathbf{T}_{2,3}^{-1} \cdot \mathbf{T}_{1,2}^{-1} \cdot \mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,n} = \mathbf{T}_{n-1,n}. \quad (3.59)$$

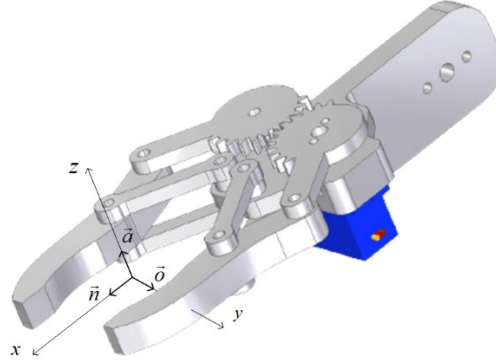
U slučaju manipulacionog robota SPT – 3000, koji posjeduje četiri stepena slobode, rješenje inverznog problema se svodi na rješavanje sljedećeg sistema jednačina:

$$\begin{aligned} \mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,4} &= \mathbf{T}_{1,4}, \\ \mathbf{T}_{1,2}^{-1} \cdot \mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,4} &= \mathbf{T}_{2,4}, \\ \mathbf{T}_{2,3}^{-1} \cdot \mathbf{T}_{1,2}^{-1} \cdot \mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,4} &= \mathbf{T}_{3,4}. \end{aligned} \quad (3.60)$$

Matrica homogene transformacije $\mathbf{T}_{0,n}$ predstavlja položaj i orijentaciju centra prihvata hvataljke u odnosu na nepokretni (bazni) koordinatni sistem, odnosno:

$$\mathbf{T}_{0,n} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{0,n} & \mathbf{p}_{0,n} \\ \hline 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.61)$$

Matrica $\mathbf{R}_{0,n}$ je matrica orijentacije hvataljke u odnosu na bazni koordinatni sistem, koja sadrži projekcije jediničnih vektora koordinatnog sistema hvataljke na bazni koordinatni sistem (slika 3.25). Vektor $\mathbf{p}_{0,n}$ predstavlja poziciju hvataljke u odnosu na bazni koordinatni sistem.



Slika 3.25. Vektori orijentacije hvataljke

Kao i u slučaju direktnog kinematičkog zadatka (poglavlje 3.6.2), SPT – 3000 se nalazi u nultom položaju. Za slučaj nultog položaja, matrica koja opisuje položaj hvataljke u odnosu na nepokretni koordinatni sistem je:

$$\mathbf{T}_{0,4} = \begin{bmatrix} 1 & 0 & 0 & 371 \\ 0 & 1 & 0 & -22.8 \\ 0 & 0 & 1 & 99.4 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Dalje, da bi se riješio sistem jednačina (3.60), neophodno je izračunati inverzne matrice, odnosno:

$$\begin{aligned} \mathbf{T}_{0,1}^{-1} &= \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & -a_1 \\ 0 & 0 & 1 & -d_1 \\ \sin \theta_1 & -\cos \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_{1,2}^{-1} = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 & -a_2 \\ -\sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{T}_{2,3}^{-1} &= \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 & -a_3 \\ -\sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_{3,4}^{-1} = \begin{bmatrix} \cos \theta_4 & \sin \theta_4 & 0 & -a_4 \\ 0 & 0 & -1 & d_4 \\ -\sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3.62)$$

Vrijednosti koordinata zglobova $\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T$ se dobijaju rješavanjem sistema jednačina (3.60). Iz prve jednačine sistema:

$$\mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,4} = \mathbf{T}_{1,4},$$

kada se izmnože matrice dobija se jednakost:

$$n_x \sin q_1 - n_y \cos q_1 = 0,$$

odakle je rješenje za ugaoni pomak prvog zgloba:

$$q_1 = \arctan\left(\frac{n_y}{n_x}\right) = \arctan\left(\frac{0}{1}\right) = 0. \quad (3.63)$$

Dalje, na osnovu druge relacije iz (3.60) može se zapisati:

$$\begin{aligned} \sin(q_2 + q_3 + q_4) &= n_z, \\ \cos(q_2 + q_3 + q_4) &= a_z. \end{aligned} \quad (3.64)$$

Nakon uvrštavanja brojnih vrijednosti u (3.62) dobija se da je zbir nepoznatih uglova jednak nuli.

Nakon množenja preostalih jednačina sistema i izjednačavanja odgovarajućih relacija, pri čemu je poznato da se robot nalazi u početnom položaju, u kojem je ostvaren i njegov maksimalni doseg, može se zaključiti da su brojne vrijednosti preostalih uglova:

$$q_2 = q_3 = q_4 = 0. \quad (3.65)$$

Rješavanje sistema nelinearnih, tj. trigonometrijskih jednačina je vremenski veoma zahtjevno i zahtijeva algebarsku i trigonometrijsku intuiciju od strane inženjera ili programera. Iako je broj operacija manji nego u slučaju numeričkog rješavanja, operacija inverzije matrice, kao i rješavanje dobijenog sistema nelinearnih jednačina nakon toga, su znatno kompleksnije u odnosu na operacije koje se javljaju kod numeričkog rješavanja. Iz tog razloga, kao i činjenice da su numerički algoritmi nezavisni od strukture manipulatora, se numerički postupak rješavanja inverznog problema najčešće primjenjuje u praksi.

4. Planiranje kretanja i koncepti upravljanja

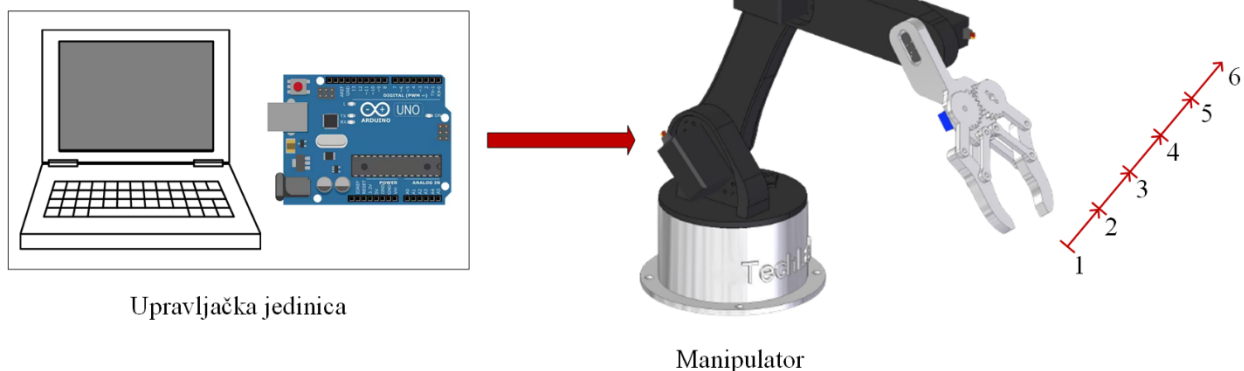
U trećoj glavi je detaljno opisana problematika izrade kinematičkog modela robotskog manipulatora, nakon čega su dati primjeri rješavanje direktnog i inverznog kinematičkog problema. Na izučavanje kinematike manipulatora, pri čemu je njen najbitniji aspekt rješavanje inverznog kinematičkog problema, se nadovezuje planiranje kretanja i upravljanje manipulatorom. Planiranje kretanja, odnosno planiranje trajektorije u suštini predstavlja generisanje niza tačaka u vremenu koje se dobijaju interpolacijom željenog puta u radnom prostoru manipulatora. Upravljanje manipulatorom podrazumijeva generisanje upravljačkih signala, primjenom upravljačkih algoritama, koji djeluju na aktuatora radi obavljanja željenog kretanja. Prvi dio ove glave se bavi strategijama upravljanja – upravljanje „od tačke do tačke” i upravljanje sa kontinualnom putanjom, kao i algoritmima upravljanja, pomoću kojih se generišu upravljački signali. Drugi dio ove glave će biti posvećen tehnikama generisanja trajektorije za različite strategije upravljanja, gdje će se obrađivati problem generisanja trajektorije u prostoru zglobova, a zatim će se izložiti osnovni koncepti planiranja trajektorije u radnom prostoru manipulatora.

4.1. Upravljanje robotom

Da bi robotski sistem, opremljen sensorima i aktuatorskim jedinicama, ostvario željeno kretanje i izvršio postavljeni radni zadatak, neophodno je adekvatno upravljanje, tj. upravljačko djelovanje. Jedno od svojstava zbog kojeg su roboti sve više zastupljeni u savremenoj industriji jeste fleksibilnost, odnosno mogućnost brzog prilagođavanja novim zadacima. Ovo svojstvo robota je moguće iz razloga što se upravljanje obavlja pomoću upravljačke jedinice, čiji se program mijenja relativno brzo i lako.

Ulogu upravljačke jedinice kod robotskog sistema vrši računar. Računar predstavlja *mozak robotskog sistema*. U slučaju SPT – 3000, upravljački sistem posjeduje dvije odvojene jedinice – računar i Arduino mikrokontroler, koje dijele uloge koje inače izvršava jedna samostalna jedinica. Zadatke rješavanja inverzne kinematike, odnosno izračunavanja trajektorija zglobova vrši računar, dok Arduino mikrokontroler vrši generisanje upravljačkih signala koji se šalju servo motorima radi izvođenja željenog kretanja. Ukoliko je trajektorija manipulatora definisana u prostoru unutrašnjih koordinata, odnosno ukoliko su trajektorije zglobova unaprijed zadate, tada problem upravljanja postaje znatno jednostavniji.

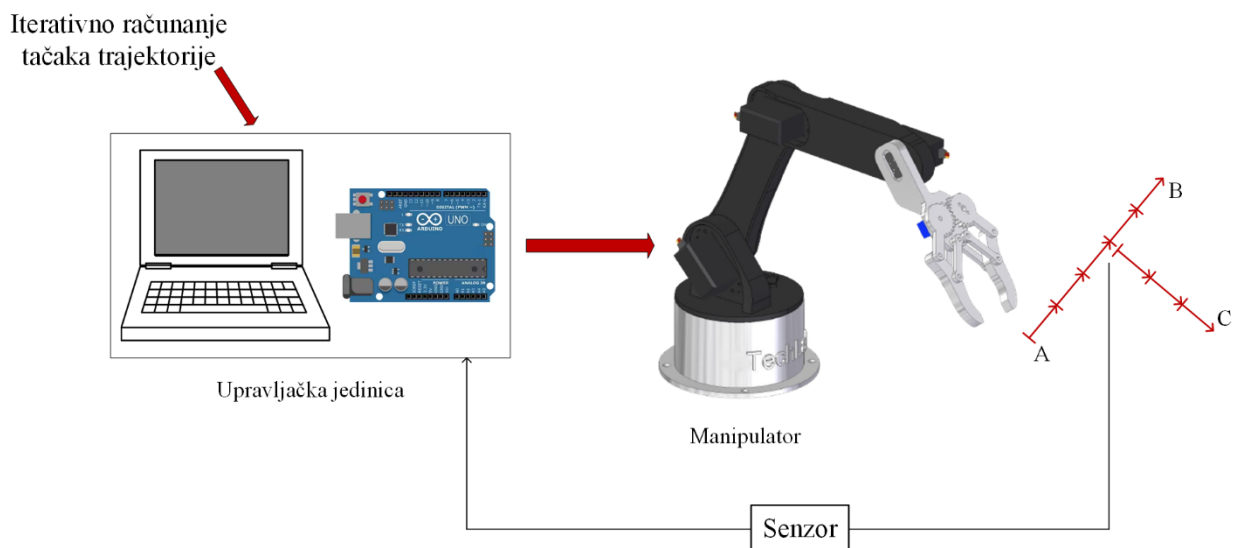
Ukoliko se sva neophodna izračunavanja obave prije nego što manipulator počne da se kreće, tada se kaže da on radi u *offline* režimu [34]. Upravljačka jedinica robotskog sistema u svojoj memoriji skladišti podatke o položajima zglobova za određene tačke trajektorije hvataljke, koji se koriste za generisanje upravljačkih signala kad robot počne sa izvršavanjem zadatka. Roboti koji rade u *offline* režimu nisu sposobni da reaguju na promjene u okolini i ne zahtijevaju složenu sensoriku. Na slici 4.1 prikazan je *offline* način rada robota, kod kojeg se završni uređaj kreće po unaprijed definisanoj trajektoriji, koja se nakon početka rada ne može mijenjati.



Slika 4.1. *Offline* režim rada

Sa druge strane, ukoliko se računanje trajektorija obavlja u toku obavljanja radnog zadatka, tada se kaže da robot radi *u realnom vremenu* [21]. Ovaj režim rada podrazumijeva da upravljačka jedinica, za svaku sljedeću tačku trajektorije završnog uređaja, u toku kretanja računa nove koordinate zglobova. Prilikom računanja se takođe moraju uzeti u obzir informacije koje se dobijaju od senzora o promjenama nastalim u okruženju robota. Iz navedenih razloga rad robota u realnom vremenu je značajno zahtjevniji sa tehničke strane, jer zahtijeva veoma brzu upravljačku jedinicu koja može vršiti potrebna izračunavanja prilagođavanja trajektorije manipulatora u realnom vremenu.

Slika 4.2 prikazuje rad robota u realnom vremenu, uz primjer promjene trajektorije u toku samog kretanja. Upravljačka jedinica vrši interpolaciju između tačaka trajektorije, tj. računa određeni broj međutačaka između zadatih tačaka A i B. Međutim, u toku kretanja između tačaka A i B, računar dobija informaciju sa senzora i zaključuje da je neophodno ciljnu tačku B zamijeniti tačkom C. U datom trenutku je neophodno izračunati koordinate novih međutačaka, kako bi završni organ nastavio kretanje i došao u novu ciljnu tačku C.



Slika 4.2. Upravljanje u realnom vremenu

Iako rad u realnom vremenu zahtijeva bržu upravljačku jedinicu i složenije upravljačke algoritme, ovaj režim rada pruža prednosti koje podrazumijevaju značajno veću fleksibilnost u radu, odnosno sposobnost da robot mijenja trajektoriju u svakom trenutku. Ta fleksibilnost naročito dolazi do izražaja pri radu u nepredvidljivim okruženjima, kada robot na osnovu informacija prikupljenih sa senzora može donositi logičke odluke.

U slučaju SPT – 3000 manipulatora, primjenjuje se *offline* režim rada, zbog ograničenja u brzini rada koje nameću MATLAB sa RTB paketom, Arduino mikrokontroler, kao i ograničenja koja nameće sistem senzora (nedostatak senzora za mjerenje parametara radnog prostora manipulatora).

4.2. Principi upravljanja robotom

Postoje razni kriterijumi za podjelu sistema upravljanja. Da bi se primijenili zakoni upravljanja na neki sistem, isti je potrebno najprije matematički modelovati. Matematičko modelovanje sistema predstavlja njegovo opisivanje sistemom jednačina koje opisuju odnos između promjenljivih sistema. Na osnovu veza između promjenljivih, sistemi upravljanja se mogu podijeliti na *linearne* i *nelinearne*.

Linearne upravljačke tehnike se mogu primijeniti samo kada je sistem matematički modelovan linearnim diferencijalnim jednačinama. U slučaju upravljanja robotom, linearne metode se mogu koristiti jedino ako se robot linearizuje, jer je robot opisan isključivo nelinearnim diferencijalnim jednačinama. Osnovni razlog zbog kojeg se vrši linearizacija nelinearnih modela je što su linearne upravljačke tehnike znatno jednostavnije za implementaciju od nelinearnih. Tehnike podešavanja parametara linearnih upravljačkih zakona su detaljno razrađene i veoma jednostavne za praktičnu primjenu. Štaviše, primjenom linearnih metoda upravljanja se može riješiti oko 90% upravljačkih zadataka u praksi.

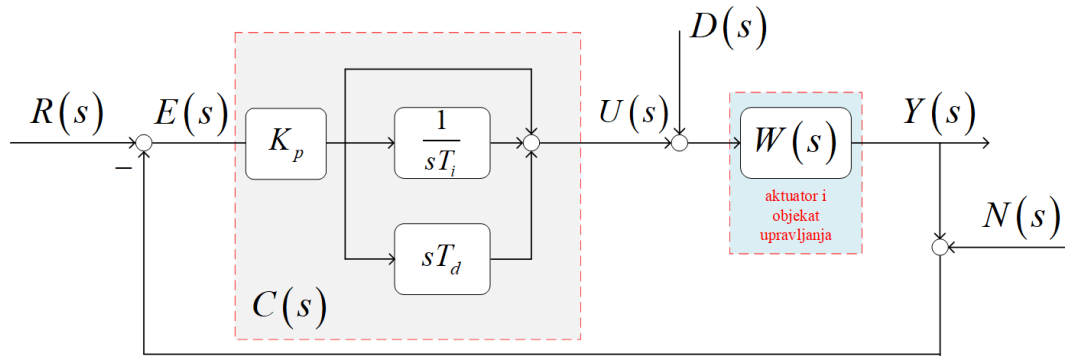
4.2.1. PID regulator

Regulator u linearnom upravljačkom sistemu može imati tri načina djelovanja: proporcionalno (P), integralno (I) i diferencijalno (D), pa se samim tim zakon upravljanja kojim se pomenuta dejstva implementiraju naziva PID regulator [35]. Upravljački zakon PID regulatora ima sljedeći oblik:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt}, \quad (4.1)$$

gdje $e(t)$ predstavlja grešku u praćenju referentnog signala.

Slika 4.4 prikazuje primjer PID regulatora u sistemu sa jediničnom povratnom spregom. U zavisnosti od toga koja od tri pomenuta dejstva su uključena u zakon upravljanja, osim PID, regulatori mogu biti i P, PD ili PI.



Slika 4.4. PID regulator

Proporcionalni regulator P, kako mu I samo ime kaže, ostvaruje isključivo proporcionalno djelovanje. Proporcionalni zakon upravljanja ima oblik:

$$u(t) = K_p e(t). \quad (4.2)$$

Proporcionalno djelovanje zavisi od veličine pojačanja K_p . Povećavanjem proporcionalnog dejstva sistem postaje brži i robusniji na poremećaje, ali i istovremeno postaje manje stabilan i osjetljiviji na šumove. Takođe, iako se povećavanjem pojačanja regulatora smanjuje statička greška sistema, ista se na ovaj način ne može u potpunosti eliminisati, tj. pasti na nulu.

Integralno dejstvo je direktno proporcionalno konstanti integraljenja T_i , koja određuje težinu njegovog djelovanja. Integrator, tj. integralni regulator vrši integraljenje signala greške, odnosno:

$$u(t) = \frac{1}{T_i} \int_0^t e(\tau) d\tau, \quad (4.3)$$

Sa povećavanjem integralnog vremena T_i sistem se usporava i smanjuje se preskok. Razlog za to je činjenica da, pri nagloj promjeni ulaznog signala, izlaz integratora se mijenja znatno sporije u odnosu na izlaz proporcionalnog regulatora. Međutim, uvođenje integratora u sistem se u potpunosti eliminiše grešku u ustaljenom stanju! Naime, sve dok postoji signal greške, ma koliko mala bila njegova vrijednost, izlaz integratora će se mijenjati, a samim će se i upravljana veličina približavati referentnom signalu.

Da bi brzina odziva sistema pri promjeni vrijednosti signala greške bila zadovoljavajuća, integralno dejstvo se uvijek koristi isključivo u kombinaciji sa proporcionalnim dejstvom, čime se dobija PI regulator. Upravljački zakon PI regulatora je:

$$u(t) = K_p e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau. \quad (4.4)$$

Podešavanjem K_p i T_i parametara mogu se postići bolje performanse sistema, pri čemu se pravi kompromis između brzine i stabilnosti sistema.

Pri željenoj brzini odziva sistema, preskok se može smanjiti jedino uvođenjem diferencijalnog dejstva. Diferencijalni regulator vrši diferenciranje signala greške:

$$u(t) = T_d e(t). \quad (4.5)$$

Sa povećavanjem diferencijalnog vremena T_d smanjuje se preskok i povećava stabilnost sistema, sistem postaje brži, ali se i značajno povećava osjetljivost sistema na mjerne šumove. Dakle, uvođenjem diferencijalnog dejstva, postiže se bolje praćenje promjene referentnog signala, upravo iz raloga što diferenciranje signala greške daje informaciju o brzini i smjeru njene promjene.

Spajanjem proporcionalnog i diferencijalnog djelovanja dobija se PD regulator. Upravljački zakon PD regulatora je:

$$u(t) = K_p e(t) + T_d \frac{de(t)}{dt}. \quad (4.6)$$

Pojačanje K_p smanjuje grešku u ustaljenom stanju i obezbjeđuje dovoljno veliku vrijednost upravljačkog signala, dok diferencijalno djelovanje unosi prigušenje u sistem i time povećava stabilnost.

Na osnovu izraza (4.5), prenosna funkcija PID regulatora je:

$$G(s) = \text{PID}(s) = K_p \left(1 + \frac{1}{sT_i} + sT_d \right). \quad (4.7)$$

Dakle, PID regulator objedinjuje dobra svojstva tri osnovna dejstva:

- proporcionalno dejstvo obezbjeđuje konstantno pojačanje,
- integralno djelovanje obezbjeđuje tačnost u stacionarnom stanju,
- diferencijalno djelovanje obezbjeđuje brzo reagovanje na promjene.

Diferencijalno dejstvo u praksi može izazvati velike problem u slučaju pojave šuma. Kako diferencijator po svojoj prirodi predstavlja visokopropusni filter, a šum se po pravilu javlja na visokim frekvencijama, dolazi do značajnog povećavanja amplitude signala šuma. Samim tim, PID će obrađivati signal šuma, što će dovesti do degradiranja performansi sistema. Efekat pojačanja šuma se može izbjeći tako što se bi se eliminisalo diferenciranje signala greške na visokim učestanostima. To znači da se diferencijalno dejstvo mora modifikovati i da se u praksi realizuje u sljedećem obliku:

$$sT_d \rightarrow \frac{sT_d}{s \frac{T_d}{N} + 1},$$

pa će prenosna funkcija realnog PID regulatora biti:

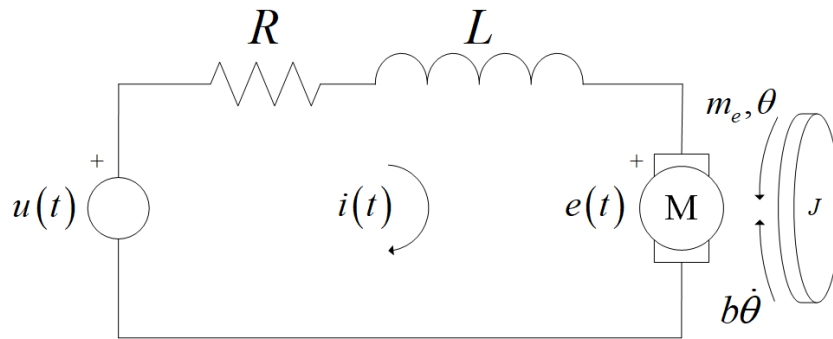
$$G_r(s) = \text{PID}_r(s) = K_p \left(1 + \frac{1}{sT_i} + \frac{sT_d}{s \frac{T_d}{N} + 1} \right). \quad (4.8)$$

Modifikovanjem diferencijalnog dejstva se uvodi realni pol, koji na visokim frekvencijama poništava uticaj nule i smanjuje uticaj šuma. Pritom, na nižim frekvencijama uticaj pola je zanemarljiv, pa se time zadržavaju prednosti diferencijalnog dejstva.

4.2.2. Servo upravljanje jednosmjernim motorom

Jednosmjerni ili DC motor je mašina jednosmjerne struje koja pretvara električnu energiju u mehaničku. DC motor se sastoji od dva nezavisna, odvojena namotaja. Jedan namotaj se nalazi na fiksnom dijelu motora, statoru, i naziva se statorski ili pobudni namotaj, dok se drugi namotaj nalazi na dijelu motora koji rotira – rotoru i stoga se naziva rotorski namotaj. Bitno je naglasiti da je uloga pobudnog namotaja stvaranje magnetskog polja koje dovodi do rotacije rotora pri proticanju struje kroz namotaj na rotoru. Pritom, pobudni namotaj se može zamijeniti stalnim magnetima koju su fiksirani na statoru.

Kako je cilj upravljati pozicijom DC motora, tj. realizovati pozicioni servomehanizam, neophodno je najprije odrediti matematički model DC motora. Iz tog razloga posmatra se šema jednosmjernog motora data na slici 4.5. Smatra se da pobuda, tj. magnetsko polje potiče od stalnih magneta.



Slika 4.5. Šema jednosmjernog motora

Dovođenje napona na namotaj rotora dovodi do proticanja struje kroz njega. Kako se rotor nalazi u prostoru u kom postoji magnetsko polje čiji je fluks Φ , dolazi do stvaranja elektromagnetskog momenta koji teži da pokrene rotor. Vrijednost ovog momenta je proporcionalna proizvodu magnetskog fluksa i jačini struje rotora:

$$m_e(t) \sim \Phi i(t) = k_m i(t), \quad (4.9)$$

gdje je $m_e(t)$ – obrtni moment koji stvara motor, k_m – konstanta motora, a $i(t)$ – struja koja protiče kroz namotaj rotora.

Usljed obrtanja rotora, a samim tim i namotaja na njemu, ugaonom brzinom ω unutar magnetskog polja, dolazi do indukovanja elektromotorne sile u rotorskom namotaju koja je proporcionalna proizvodu ugaone brzine rotora i fluksa magnetskog polja:

$$e(t) \sim \Phi \omega(t) = k_m \omega(t), \quad (4.10)$$

gdje je $\omega(t)$ – ugaona brzina motora, a $e(t)$ – indukovana elektromotorna sila u namotaju rotora. Indukovana elektromotorna sila se još naziva i kontra elektromotorna sila, jer je uvijek takvog smjera da se suprotstavlja naponu koji se dovodi na krajeve namotaja rotora.

Na osnovu izraza (4.9) i (4.10) i šeme motora sa slike 4.5, mogu se napisati sljedeće jednačine:

$$J \frac{d\omega(t)}{dt} + b\omega(t) = k_m i(t) \rightarrow (Js + b)\Omega(s) = k_m I(s),$$

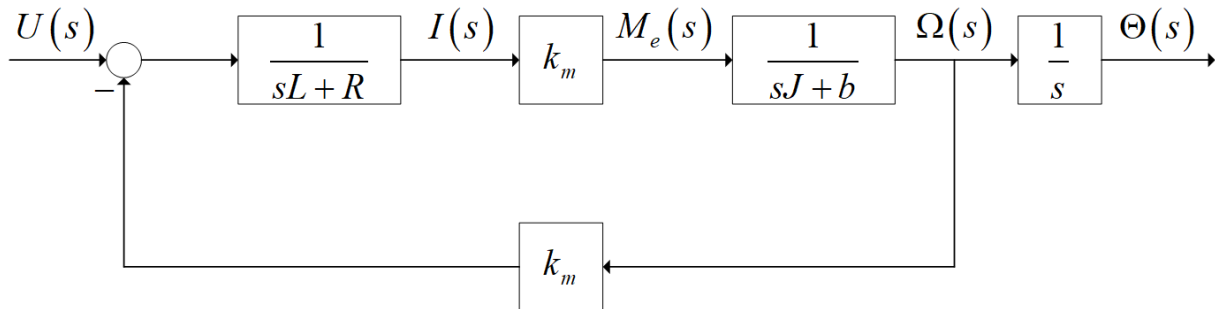
$$L \frac{di(t)}{dt} + Ri(t) = u(t) - k_m \omega(t) \rightarrow (Ls + R)I(s) = U(s) - k_m \Omega(s),$$
(4.11)

gdje je L – induktivnost namotaja rotora, R – otpornost namotaja rotora, J – ekvivalentni moment inercije motora i opterećenja sveden na osovinu motora, b – koeficijent viskozno-friktionog trenja, $u(t)$ – napon na namotaju rotora (upravljački signal).

Na osnovu jednačina (4.11), funkcija prenosa DC motora je:

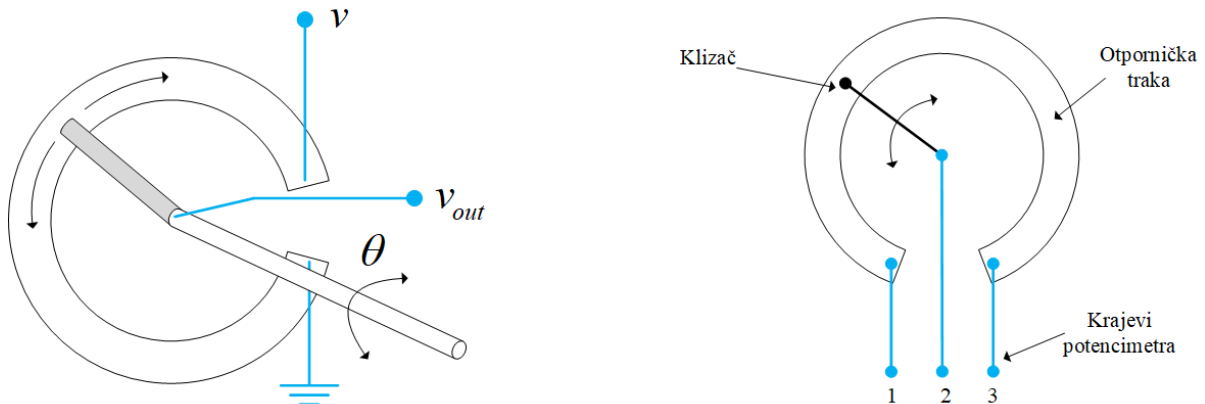
$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{k_m}{JLs^2 + (bL + JR)s + bR + k_m^2}.$$
(4.12)

Kako se nastoji upravljati ugaonom pozicijom jednosmjernog motora, korisno je da se ista posmatra kao izlaz sistema. Ugaona pozicija se dobija kao integral ugaone brzine, odnosno u uvodi se integrator u strukturni blok dijagram sistema, koji je dat na slici 4.6.



Slika 4.6. Strukturni blok dijagram DC motora

Ukoliko se želi upravljati ugaonom pozicijom jednosmjernog motora pomoću povratne sprege, neophodan je senzor za mjerenje ugaone pozicije na osovini motora. Jedan od često korišćenih senzora pozicije, a koji se i koristi u servo motorima za SPT – 3000, jeste potencijometar, koji u suštini predstavlja pretvarač ugaone pozicije u naponski signal.



Slika 4.7. Princip rada potencijometra

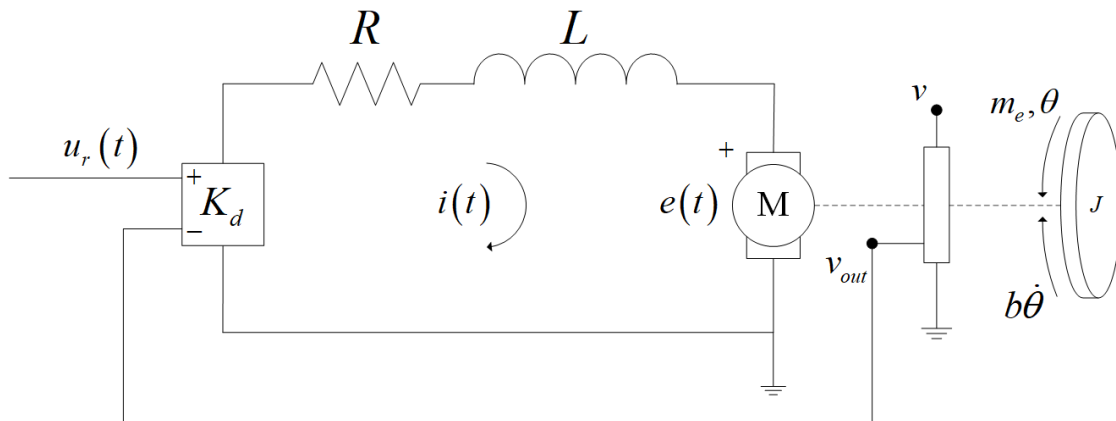
Na slici 4.7 je prikazan princip rada potencijometra. Dugačak provodnik torusnog oblika je postavljen u unutrašnjostu potencijometra, pri čemu se jedan kraj provodnika vezuje za jednosmjerni

napon, dok je drugi njegov kraj uzemljen. U unutrašnjosti potencijometra se nalazi i klizač, vezan za osovinu oko koje rotira. Rotacijom klizača mijenja se njegova pozicija, a samim tim i izlazni napon potencijometra. Kako se ugao povećava, klizač se približava kraju na koji je prikačen izvor napona, pa je samim tim izlazni napon veći. Odnos između izlaznog napona potencijometra i ugaone pozicije klizača je sljedeći:

$$v_{out} = \frac{R_{\theta}}{R} v = \frac{\frac{\theta}{\theta_{pot}} R}{R} v = \frac{v}{\theta_{pot}} \theta = k_p \theta. \quad (4.13)$$

Dakle, veza između napona na izlazu potencijometra i ugaone pozicije klizača je linearna i može se opisati nekom konstantom proporcionalnosti k_p .

Sada će biti opisan pozicioni servomehanizam po ugaonoj poziciji, na bazi jednosmjernog motora sa konstantnom pobudom, uz potencijometar kao senzor pozicije. Šematski dijagram pozicionog servo sistema dat je na slici 4.8. Na osovini DC motora je postavljen potencijometar koji daje informaciju, tj. napon koji je proporcionalan uglu rotacije. Referentni naponski signal, koji predstavlja željenu ugaonu poziciju, i signal sa potencijometra se vode na ulaz diferencijalnog pojačavača, koji razliku dva napona pojačava K_d puta i generiše upravljački signal.

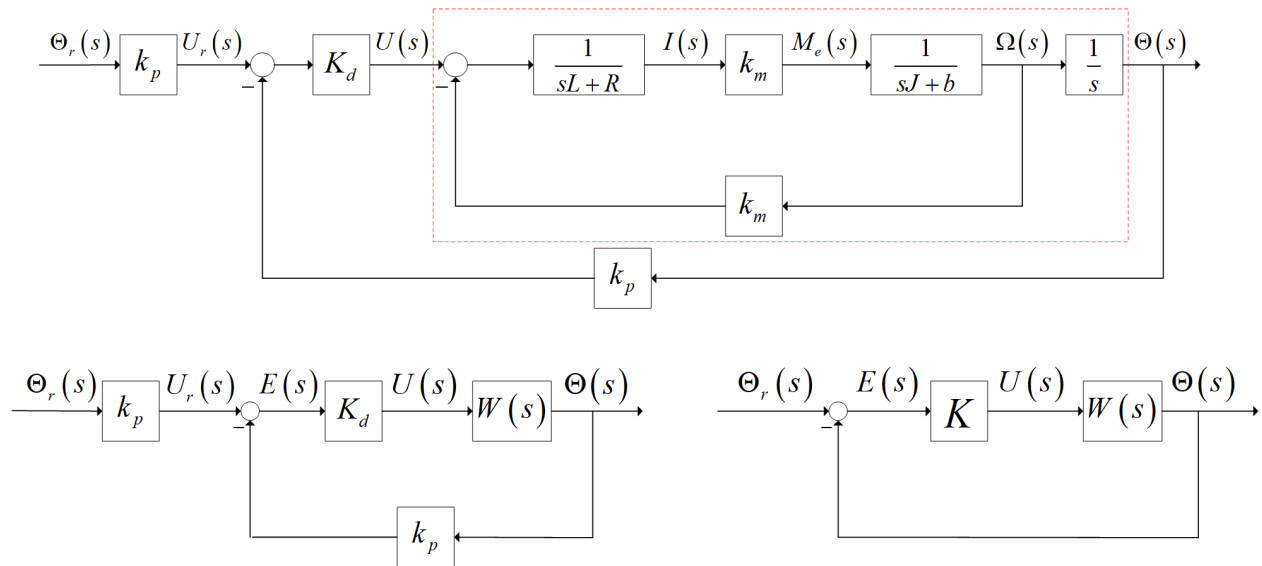


Slika 4.8. Pozicioni servomehanizam

Na osnovu dijagrama na slici 4.8, može se nacrtati strukturni blok dijagram pozicionog servo sistema. Jednosmjerni motor i opterećenje se mogu predstaviti jednom funkcijom prenosa, pa se cjelokupni blok dijagram može svesti na šemu sa jednom jediničnom povratnom spregom (slika 4.9).

Servo motori SPT – 3000 posjeduju integrisani pozicioni servomehanizam, koji uključuje primjenu povratne sprege i PID regulatora. Prilikom procesa sklapanja manipulatora razmatrana je mogućnost implementacije dodatne povratne sprege na Arduino mikrokontroleru, uz dobijanje informacije o pozicijama zglobova sa potencijometara unutar servo motora. Na ovaj način bi se moglo postići preciznije praćenje zadatih pozicija zglobova. Međutim, problem kod ovakvog pristupa je što je povratna informacija koja se dobija od senzora nelinearna i ne mijenja se u cjelokupnom opsegu AD konvertora mikrokontrolera. Ovo za posljedicu ima da se, prilikom skaliranja informacije sa senzora, pikovi i nelinearnosti u povratnoj informaciji znatno pojačavaju i da se, samim tim, gubi na preciznosti upravljanja servo motorom. Iz tog razloga se u eksperimentu

koristi isključivo pretvaranje trenutnih pozicija servo motora u širine upravljačkih impulsa, čime se postiže rezolucija upravljačkog signala od $0,135^\circ/\mu s$ (tabela 2.1).



Slika 4.9. Strukturni blok dijagram pozicionog servosistema

4.2.3. Nivoi i strategije upravljanja robotom

U drugoj glavi su definisane upravljačke promjenljive za pogone SPT – 3000 manipulatora. U slučaju jednosmjernih servo motora upravljačke promjenljive su PWM signali koji sadrže informaciju u željenim ugaonim pozicijama zglobova. Nakon definisanja upravljačkih promjenljivih, zadatak upravljanja se može definisati kao: *Neophodno je obezbijediti takvu promjenu zadatih veličina koja će proizvesti zadato kretanje u zglobovima. Dakle, upravljački zadatak se svodi na pokretanje zglobova na zadati način.*

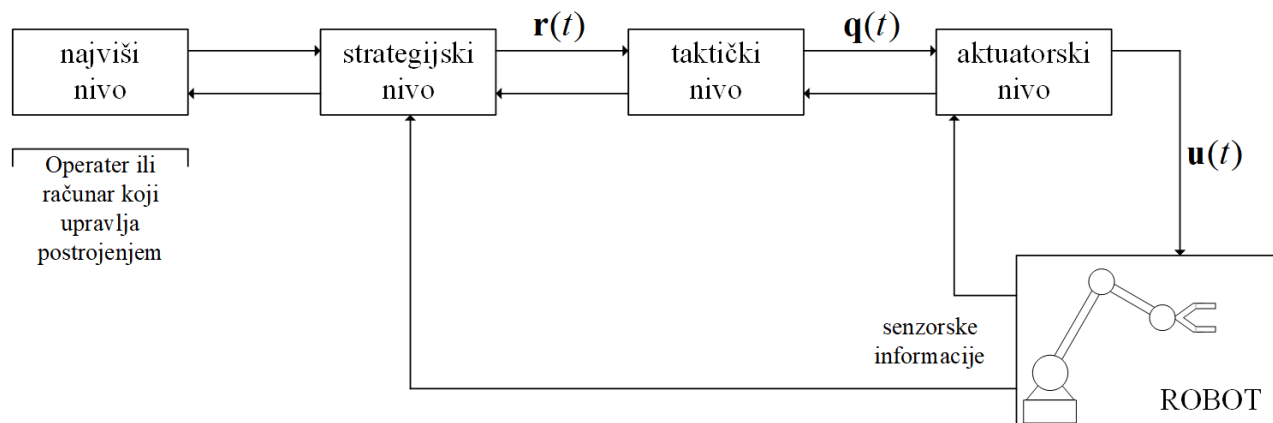
Prethodna definicija upravljačkog zadatka se svodi na operativni ili aktuatorski nivo. Međutim, manipulator treba da vrši zadato kretanje kako bi završni uređaj, zavisno od njegove konfiguracije, izvršio traženu operaciju (zavarivanje, farbanje, sklapanje, itd.). S tim na umu, zadatak upravljanja se može preformulisati na sljedeći način: *Potrebno je obezbijediti promjenu upravljačkih promjenljivih koja će rezultovati traženim kretanjem završnog uređaja u radnom prostoru.* Ova definicija uključuje i prethodnu, jer je najprije potrebno izračunati trajektorije zglobova, a tek nakon toga i vršiti upravljanje zglobovima.

Prethodne dvije definicije problema upravljanja su se oslanjale na kinematiku manipulatora. Međutim, zadatak upravljanja može biti i opštiji, tj. može se formulirati u vidu zahtjeva za izvršenje neke složene, praktične operacije. Na primjer, zadatak može biti: zavrnuti šraf u određeni otvor. Ovako definisan zadatak sadrži niz operacija. Prva je hvatanje šrafa, zatim postavljanje ispred otvora, nakon toga je potrebno zašrafiti šraf i na kraju vraćanje u početni položaj. Ovaj zadatak može biti još složeniji ako je, na primjer, potrebno sakupiti sve predmete na nekoj podlozi. U tom slučaju se broj elementarnih operacija značajno povećava.

Ovako definisan zadatak nije, kao kod prethodnih slučajeva, kinematički orijentisan (zadato kretanje), već je sada problemski orijentisan (izvršavanje određene radnje). Postavljeni problem manipulator najprije mora raščlaniti na elementarne radnje, tj. pokrete, a zatim iste i izvršiti. Takođe se mora uzeti u obzir i činjenica da radni prostor manipulatora i radne operacije nijesu uvijek precizno definisani. Najčešći primjer toga je pojava prepreka u radnom prostoru. Dakle, osim raščlanjivanje problema i utvrđivanje redosljeda pokreta, neophodno je i vršiti stalno ispitivanje radnog prostora kako bi se neophodne korektivne akcije. Sada se zadatak upravljanja može formulisati kao: *analizirati, a zatim izvršiti traženu operaciju*. Očigledno, ovakva formulacija zahtijeva određenu inteligenciju i sensoriku od robotskog sistema.

Na kraju, može se uvesti i dodatno uopštenje – ne mora se zadavati čak ni operacija koju treba izvršiti. Može se, na primjer, postaviti zadatak: sklopiti dati uređaj. U ovom slučaju robotski sistem najprije vrši analizu, nakon koje određuje koje operacije treba izvršiti (npr. postavljenje dijelova, povezivanje dijelova, itd.). Ovo dodatno uopštenje orijentiše zadatak upravljanja ka cilju, pa je njegova konačna definicija: *Neophodno je izvršiti operacije potrebne za postizanje traženog cilja*.

Uopštavanjem zadatka koji se postavlja robotu dolazi se do *nivoa upravljanja* [36], koji su prikazani na slici 4.10, pri čemu svaki viši nivo razlaže postavljeni zadatak i upravljanja nivoom ispod njega. U zavisnosti od potrebe, svaki nivo raspolaže određenim informacijama dobijenim od senzora.



Slika 4.10. Nivoi upravljanja

Aktuatorski ili *izvršni nivo* predstavlja najniži nivo upravljanja i na njemu se neposredno izvršava kretanje. Zadatak za ovaj nivo mora biti u obliku željenih trajektorija zglobova $\mathbf{q}(t)$. Pojedinačni aktuatori (u slučaju SPT – 3000 servo motori) u svakom zglobu izvršavaju traženo kretanje. Ovaj nivo prima zadatak od višeg nivoa ili direktno od operatera, a od senzorskih informacija može koristiti informacije o pozicijama, brzinama i ubrzanjima zglobova.

Taktički nivo je prvi od viših nivoa upravljanja i on vrši raspodjelu kretanja pojedinačnih zglobova. Drugim riječima, na ovom nivou se vrši rješavanje inverznog kinematičkog zadatka. Ovaj nivo ne zahtijeva senzorske informacije.

Strategijski nivo analizira problemski orijentisan zadatak i vrši njegovu podjelu na pojedinačne pokrete završnog alata $\mathbf{r}(t)$. Pri tome je potrebno izvršiti i planiranje kretanja koje omogućava da se podjela izvrši na jednoznačan način (npr. utvrditi kriterijum redosljeda

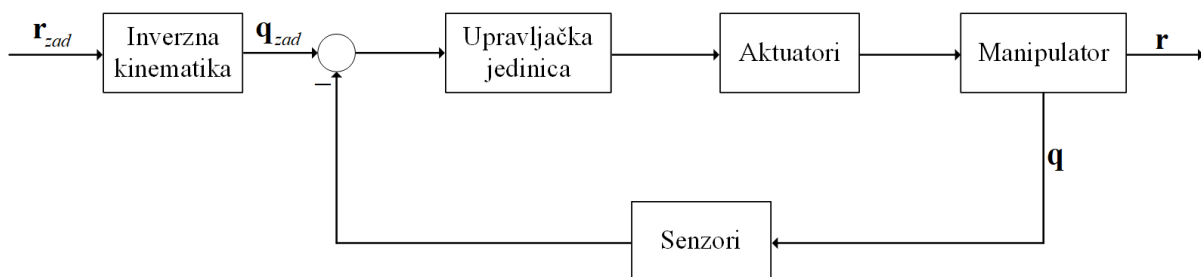
operacija). Sensorske informacije na ovom nivou često potiču od sistema vizije, senzora distance i slično.

Najviši nivo analizira cilj koji se želi postići, analizira ga i generiše skup radnih operacija koje su potrebne za njegovo postizanje.

Ukoliko se žele imati nivou upravljanja viši od taktičkog, neophodno je da robotski sistem posjeduje određeni nivo vještačke inteligencije, uz adekvatnu sensoriku kojom može ispitivati svoje radno okruženje. Kako SPT – 3000 manipulator, ne posjeduje elemente vještačke inteligencije i sensoriku kojom bi ispitivao svoje radno okruženje, njegov najviši nivo upravljanja je taktički nivo, koji rješava inverzni kinematički zadatak koji zadaje programer. Ukoliko se uzme u obzir da programer može definisati radni zadatak kako u prostoru unutrašnjih, tako i u prostoru spoljašnjih koordinata, jasno se izdvajaju dva tipa upravljačkih strategija:

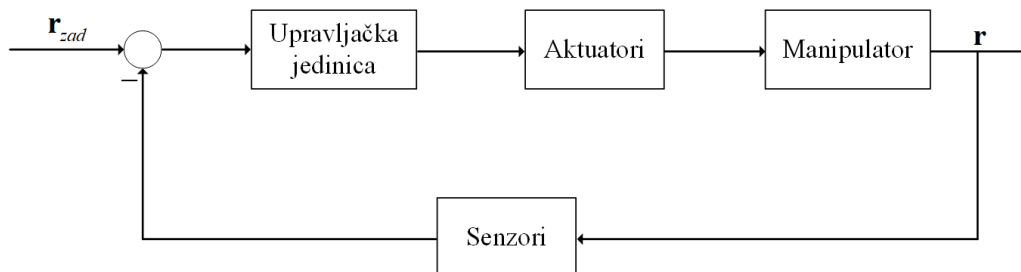
- upravljanje u prostoru zglobova (eng. *joint space control*) i
- upravljanje u radnom prostoru (eng. *operational space control*).

U slučaju upravljanja u prostoru zglobova, čija je šema data na slici 4.11, zadato kretanje završnog uređaja se prevodi u prostor unutrašnjih koordinata, tj. dobijaju se trajektorije zglobova. Povratna petlja je zatvorena preko vektora unutrašnjih koordinata, što predstavlja manu ove strategije upravljanja. Razlog tome je što se, kod ovog pristupa, sve nepreciznosti u strukturi i mehanici manipulatora direktno prenose na promjenljive u radnom prostoru.



Slika 4.11. Upravljanje u prostoru zglobova

Upravljanje u radnom prostoru (slika 4.12) zahtijeva znatno kompleksnije računске operacije i algoritme upravljanja, u odnosu na prethodni pristup. Razlog tome je činjenica da je, kod ove strategije upravljanja, inverzna kinematika dio povratne sprege. Međutim, prednost ovog pristupa upravljanju u odnosu na upravljanje u prostoru zglobova je praćenje promjenljivih radnog prostora i zatvaranje povratne sprege preko istih, čime se mogu eliminisati nepreciznosti u kinematičkom modelu.



Slika 4.12. Upravljanje u radnom prostoru

4.2.4. Tipovi upravljanja

Ukoliko se posmatra izvršni, tj. aktuatorski nivo upravljanja, mogu se izdvojiti dva osnovna tipa upravljanja, koja su tijesno vezana i za načine postavljanja zadataka. Dva osnovna tipa upravljanja su:

- upravljanje od tačke do tačke (eng. *point-to-point control*),
- upravljanje sa kontinualnom putanjom (eng. *continuous path control*) [21].

Upravljanje od tačke do tačke predstavlja zadavanje niza različitih položaja manipulatoru, pri čemu on mora redom da dođe u svaki od njih, pri čemu je nebitan način kretanja između dva položaja. Dakle, zadatak upravljanja se svodi na to da svaki pojedinačni zglob robota, polazeći iz proizvoljnog položaja, postigne svoju zadatu poziciju. Ovim načinom upravljanja se može riješiti određena grupa zadataka u industriji, kao na primjer prenošenje predmeta, bušenje rupa, sklapanje uređaja i slično.

Upravljanje sa kontinualnom putanjom podrazumijeva praćenje zadate trajektorije od strane manipulatora, odnosno da svaki zglob i sa određenom tačnošću prati zadatu trajektoriju $q_{izad}(t)$. Tipična primjena ovog načina upravljanja je kod farbanja, lijepljenja, lučnog zavarivanja i slično. Upravljanje sa kontinualnom putanjom predstavlja komplikovaniji i opštiji problem od upravljanja od tačke do tačke. U suštini, upravljanje od tačke do tačke se može posmatrati kao upravljanje kontinualnom putanjom, ukoliko se usvoji da je zadata putanja jednaka konstanti ($q_{izad}(t) = q_i = const.$).

Iako se upravljanjem od tačke do tačke može riješiti jednostavan zadatak kao što je prenošenje predmeta, što je i osnovna namjena manipulatora kao što je SPT – 3000, prilikom eksperimenta se primjenjuje isključivo upravljanje sa kontinualnom putanjom. Razlog za to je upravo činjenica da se kod upravljanja od tačke do tačke ne obraća pažnja na način kretanja između dva položaja. Ukoliko bi se servo motorima SPT – 3000 prosljedila isključivo informacija o zadatoj poziciji u vidu upravljačkog impulsa konstantne širine, servo motori bi se kretali maksimalnim mogućim brzinama, što bi dovelo do pojave visokih vrijednosti ubrzanja, a samim tim i sila koje djeluju na zglobove, čime se javlja opasnost od havarije i oštećenja na manipulatoru. Kako bi se izbjegle potencijalne havarije i oštećenja, primjenjuje se isključivo upravljanje sa kontinualnom putanjom. Kako bi se vršilo upravljanje sa kontinualnom putanjom, neophodno je generisati trajektorije zglobova.

4.3. Planiranje trajektorije

Cilj planiranja trajektorije je generisanje referentnih ulaza za sistem upravljanja čime se garantuje da će manipulator ispratiti planirane trajektorije. Korisnik ili programer definiše niz parametara koji opisuju željenu trajektoriju. Planiranje u suštini predstavlja generisanje vremenske sekvence vrijednosti koje se dobijaju interpolacijom tačaka koja predstavljaju željenu putanju. U ovom poglavlju će biti opisane pojedine tehnike za generisanje trajektorije, za osnovne tipove upravljanja: kada su zadate samo početna i krajnja puta (upravljanje od tačke do tačke) i u slučaju kada je je zadat konačan niz tačaka u prostoru duž željene putanje (upravljanje kontinualnom putanjom). Najprije se razmatra problem planiranja trajektorije u prostoru zglobova, a zatim se razmatraju osnovni koncepti planiranja trajektorije u radnom prostoru robota.

4.3.1. Put i trajektorija

Minimalni zahtjev za svaki manipulator je sposobnost kretanja od početnog do krajnjeg zadatog položaja. Takođe, tranzicija između početnog i krajnjeg položaja treba da podliježe zakonima kretanja koji drže aktuatore unutar njihovih sopstvenih limita, kako ne bi došlo do nepoželjnih pokreta. U suštini, neophodno je osmisliti algoritme planiranja koji generišu prihvatljive, a ujedno i „glatke” trajektorije.

Kako se izrazi put i trajektorija često poistovjećuju, potrebno je objasniti razliku između njih. *Put* predstavlja niz tačaka u prostoru zglobova, ili u prostoru spoljašnjih koordinata, koji manipulator mora da prati u toku izvršavanja zadatog kretanja, odnosno put je čisto geometrijski opis kretanja. Sa druge strane, *trajektorija* predstavlja put za koji je određen zakon vremena, u smislu brzina i/ili ubrzanja u svakoj njegovoj tački [22].

U principu, može se smatrati da su ulazi planiranja trajektorije opis puta, ograničenja puta i ograničenja koja nameće sama dinamika manipulatora, dok su izlazi trajektorije završnog organa, odnosno vremenske sekvence koje definišu promjene položaja, brzine i ubrzanja. Dakle, algoritam za planiranje trajektorije generiše vremensku sekvencu promjenljivih koje opisuju poziciju i orijentaciju završnog uređaja u vremenu, uzimajući u obzir nametnuta ograničenja. Kako se upravljanje manipulatorom izvodi u prostoru zglobova, nakon planiranja trajektorije završnog uređaja neophodno je, korišćenjem odgovarajućeg algoritma inverzne kinematike, dobiti i trajektorije zglobova koje odgovaraju trajektoriji u radnom prostoru.

Ukoliko prilikom dobijanja trajektorija zglobova iz trajektorije završnog uređaja dođe do poteškoća usljed pojave singulariteta ili redundantnih stepena slobode, tada se preporučuje zadavanje puta u prostoru unutrašnjih koordinata. Iz tog razloga, i radi jasnoće izlaganja, najprije će se obraditi planiranje trajektorije u prostoru zglobova, a dobijeni rezultati će se zatim proširiti na slučaj trajektorija u prostoru spoljašnjih koordinata.

4.3.2. Trajektorije u prostoru zglobova

Kretanje manipulatora se najčešće definiše u prostoru spoljašnjih koordinata u smislu parametara trajektoriji kao što su početni i krajni položaj završnog uređaja, mogući položaji između početne i krajnje tačke i vrijeme putovanja na pojedinim geometrijskim putevima. Ukoliko se trajektorija želi planirati u prostoru zglobova, najprije se moraju odrediti vrijednosti koordinata zglobova za svaki zadati položaj završnog organa duž puta. Algoritam za planiranje generiše funkciju $q(t)$ koja vrši interpolaciju nizova koordinata svakog pojedinačnog zgloba, uzimajući u obzir nametnuta ograničenja. U opštem slučaju, algoritam planiranja trajektorije u prostoru zglobova mora ispunjavati sljedeće kriterijume:

- generisane trajektorije nisu zahtjevne u smislu potrebne računarske (procesorske) moći,
- pozicije i brzine zglobova su kontinualne funkcije vremena,
- neželjeni efekti, npr. pojava pikova u trajektorijama prilikom interpolacije puta, su svedeni na minimum.

Najprije će se posmatrati slučaj kada su zadate samo početna tačka puta, krajnja tačka puta i vrijeme putovanja između njih (kretanje od tačke do tačke), zatim će rezultati biti generalizovani za slučaj kada se zadaju i tačke između početka i kraja puta (kretanje kroz niz tačaka).

4.3.2.1. Kretanje od tačke do tačke

U slučaju kretanja od tačke do tačke, manipulator mora doći iz početne u krajnju konfiguraciju zglobova za neko vrijeme t_k . U ovom slučaju putanja koju pređe završni organ nije od značaja. Algoritam treba da generiše trajektoriju koja, osim što zadovoljava prethodno navedene kriterijume, je pritom i glatka, tj. da je njenih prvih par izvoda po vremenu kontinualno. Razlog za to je smanjenje dinamičkog opterećenja koja trpe aktuatori u zglobovima i izbjegavanje pojave trzaja i preskoka prilikom kretanja manipulatora. Najčešći zahtjevi koji se postavljaju je da su prvi i drugi izvod trajektorije zgloba, brzina i ubrzanje, kontinualne funkcije vremena, a nekada i izvod ubrzanja (treći izvod pozicije) ili trzaj.

Očigledan predlog za funkciju trajektorije koja zadovoljava uslov glatkoće je *polinomijalna funkcija vremena* [24]. Polinomijalne funkcije su jednostavne za izračunavanje i pomoću njih se lako može dobiti tražena glatkoća trajektorije zgloba. Na primjer, posmatra se polinomijalna funkcija trećeg reda koja predstavlja trajektoriju zgloba:

$$q(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0, \quad (4.14)$$

za koju se dobija parabolni profil brzine zgloba

$$\dot{q}(t) = 3a_3 t^2 + 2a_2 t + a_1, \quad (4.15)$$

i linearni profil ubrzanja

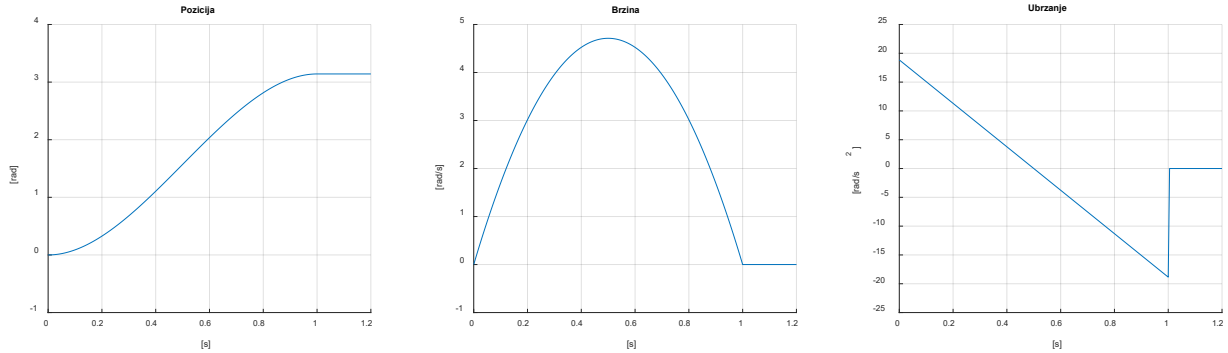
$$\ddot{q}(t) = 6a_3 t + 2a_2. \quad (4.16)$$

Kako je potrebno odrediti četiri koeficijenta, mogu se, osim početne i krajnje vrijednosti položaja zgloba q_p i q_k , definisati i početna i krajnja vrijednost brzine \dot{q}_p i \dot{q}_k , koje su obično jednake nuli. Rješavanjem sljedećeg sistema jednačina:

$$\begin{cases} a_0 = q_p \\ a_1 = \dot{q}_p \\ a_3 t_k^3 + a_2 t_k^2 + a_1 t_k + a_0 = q_k \\ 3a_3 t_k^2 + 2a_2 t_k + a_1 = \dot{q}_k \end{cases}, \quad (4.17)$$

gdje je t_k vrijeme trajanja kretanja, mogu se odrediti koeficijenti polinoma (4.18) i samim tim dobiti određena trajektorija zgloba.

Slika 4.13 prikazuje promjene pozicije, brzine i ubrzanja u vremenu za sljedeće parametre: $q_p = 0$, $q_k = \pi$, $\dot{q}_p = \dot{q}_k = 0$ i $t_k = 0$. Sa slike se može zaključiti da brzina ima parabolni profil, dok ubrzanje ima linearni profil i nepoželjne pikove u početnom i krajnjem trenutku.



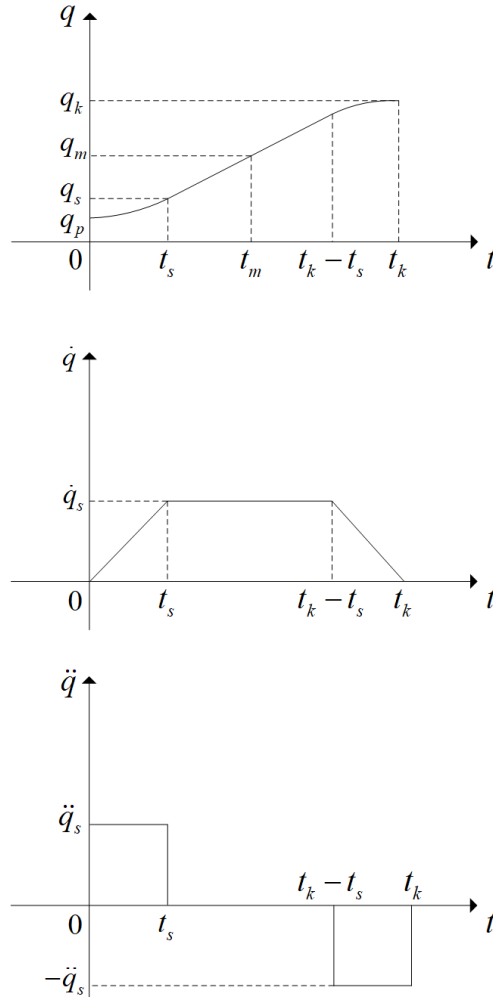
Slika 4.13. Promjena pozicije, brzine i ubrzanja u vremenu

Ukoliko se postave ograničenja na početnu i krajnju vrijednost ubrzanja, uz već pomenuta ograničenja pozicije i brzine, dobija se da je potrebno zadovoljiti šest ograničenja odnosno uslova, pa je samim tim potrebno izabrati polinom makar petog reda. U opštem slučaju, zakon promjene pozicije zgloba u vremenu se može zapisati kao:

$$q(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0, \quad (4.18)$$

čiji se koeficijenti mogu odrediti, kao i slučaju (4.18), određivanjem vrijednosti za $t=0$ i $t=t_k$ za poziciju zgloba i njena prva dva izvoda.

Alternativni pristup, koji se veoma često primjenjuje u industriji, je pristup sa kombinovanjem polinoma različitih stepena, koji omogućava testiranje brzina i ubrzanja dobijenih proračunom direktno na mehaničkom manipulatoru. Kod ovog pristupa se usvaja *trapezoidni profil brzine* [37], koji karakteriše konstantno ubrzanje u početnoj fazi, konstantna brzina u sredini i konstantno usporavanje prilikom dolaska u krajnju tačku. Rezultantna trajektorija pozicije se dobija kombinovanjem linearnog segmenta u sredini i parabolnih segmenata u početnoj i krajnjoj fazi. Takođe, u industrijskoj praksi korisniku se omogućava da postavi procenat maksimalne brzine koju manipulator može dostići u toku kretanja, čime se izbjegavaju slučajevi kada je zadato vrijeme kretanja premalo, što bi dovelo do prevelikih vrijednosti brzina i ubrzanja, koje manipulator fizički ne može postići.



Slika 4.14. Trajektorija zgloba – trapezoidni profil brzine

Slika 4.14 prikazuje profile pozicije i ubrzanja, zajedno sa odabranim trapezoidnim profilom brzine. Za slučaj sa slike je usvojeno da su početna i krajnja brzina jednake nuli i da segmenti sa konstantnim ubrzanjima imaju jednako vrijeme trajanja, čime se dobija jednake magnitude ubrzanja kod oba segmenta. Takođe je bitno napomenuti da odabrani profil brzine daje trajektoriju zgloba koja je simetrična u odnosu na srednju tačku, tj.

$$q_m = \frac{q_k + q_p}{2}, t = t_k/2.$$

Trajektorija zgloba mora zadovoljiti određene uslove kako bi se osiguralo pomjeranje od q_p do q_k za vrijeme t_k . Na primjer, brzina na kraju parabolnog segmenta mora biti jednaka konstantnoj brzini linearnog segmenta, odnosno

$$\ddot{q}_s t_s = \frac{q_m - q_s}{t_m - t_s}, \quad (4.19)$$

gdje je q_s vrijednost pozicije zgloba na kraju parabolnog segmenta, za vrijeme t_s i konstantno ubrzanje \ddot{q}_s . Takođe se može zapisati

$$q_s = q_p + \frac{1}{2} \ddot{q}_s t_s^2. \quad (4.20)$$

Kombinovanjem (4.19) i (4.20) dobija se

$$\ddot{q}_s t_s^2 - \ddot{q}_s t_k t_s + q_k - q_p = 0. \quad (4.21)$$

Često se za \ddot{q}_s uvodi uslov $\text{sgn } \ddot{q}_s = \text{sgn}(q_p - q_k)$, pa se za date t_k , q_p i q_k vrijeme parabole t_s , na osnovu (4.21), izračunava po formuli ($t_s \leq t_k/2$):

$$t_s = \frac{t_k}{2} - \frac{1}{2} \sqrt{\frac{t_k^2 \ddot{q}_s - 4(q_k - q_p)}{\ddot{q}_s}}. \quad (4.22)$$

Pritom, ubrzanje mora zadovoljavati uslov:

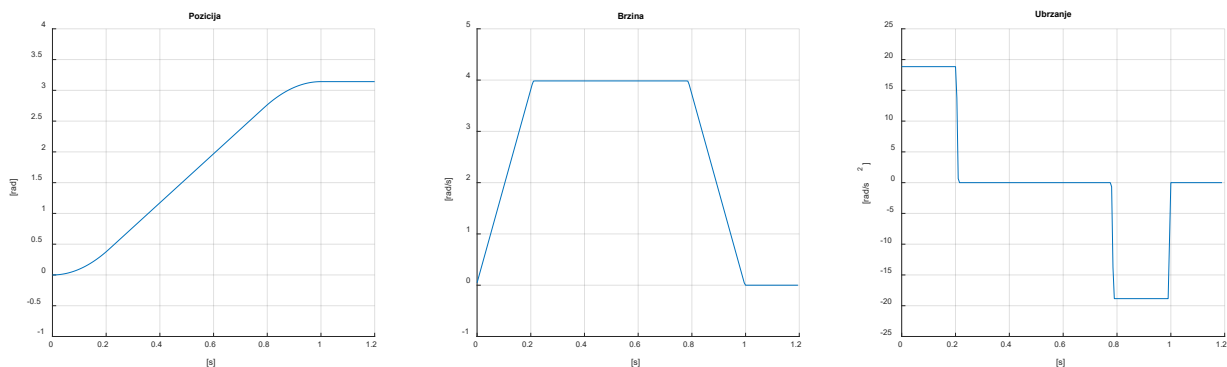
$$|\ddot{q}_s| \geq \frac{4|q_k - q_p|}{t_k^2}. \quad (4.23)$$

U slučaju kada ubrzanje \ddot{q}_s zadovoljava uslov (4.23) sa znakom jednakosti, rezultatna trajektorija neće imati segment sa konstantnom brzinom, već samo segmente sa ubrzanjem i usporenjem (*trougaoni* profil brzine).

Za poznate q_p , q_k i t_k , iz uslova (4.23) se može odrediti odgovarajuće ubrzanje. Zatim se iz (4.22) može odrediti vrijeme t_s i na kraju se dobiti sljedeći niz polinoma, tj. trajektorija:

$$q(t) = \begin{cases} q_p + \frac{1}{2} \ddot{q}_s t^2 & 0 \leq t < t_s \\ q_p + \ddot{q}_s t_s (t - t_s/2) & t_s < t \leq t_k - t_s \\ q_p - \frac{1}{2} \ddot{q}_s (t_k - t)^2 & t_k - t_s < t \leq t_k \end{cases} \quad (4.24)$$

Slika 4.15 prikazuje zakon promjene kretanja (4.28), za sljedeće parametre: $q_p = 0$, $q_k = \pi$, $t_k = 1$ i $|\ddot{q}_s| = 6\pi$.



Slika 4.15. Promjena pozicije, brzine i ubrzanja u vremenu – trapezoidni profil brzine

4.3.2.2. Kretanje kroz niz tačaka

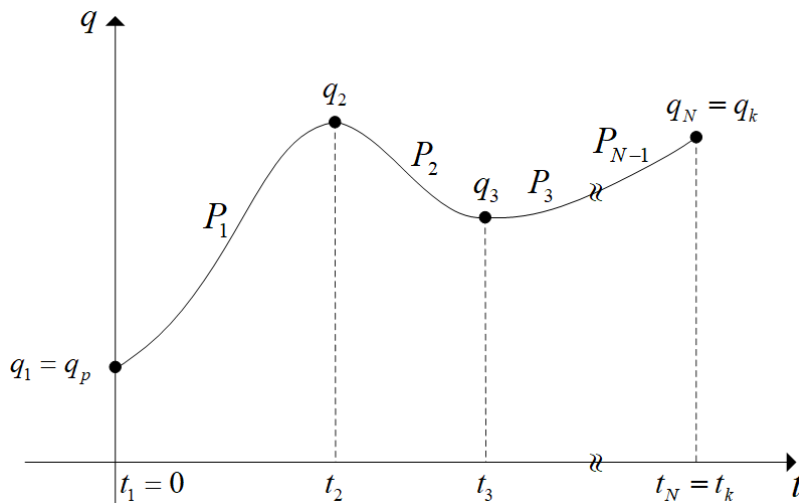
U mnogim slučajevima put je opisan sa više od dvije tačke. Na primjer, čak i u slučaju kretanja od tačke od tačke za zadatak premještanja radnog objekta može biti korisno definisanje dvije tačke između početne i krajnje tačke – mogu se odrediti prikladne pozicije za podizanje i spuštanje objekta, čime bi se smanjile brzine kretanja u odnosu na direktno kretanje ka objektu. Za komplikovanije zadatke poželjno je definisati niz tačaka, pri čemu je gustina tačaka veća na djelovima puta gdje je potrebno izbjeći prepreke ili se očekuje visoka zakrivljenost puta. Takođe se mora uzeti u obzir činjenica da se odgovarajuće promjenljive zglobova moraju izračunati na osnovu položaja u radnom prostoru.

Dakle, problem koji treba riješiti je generisanje trajektorije na osnovu N zadatah tačaka, koje se nazivaju *tačke puta*, koje manipulator treba da dostigne u određenim vremenskim trenucima. Za svaki zglob je definisano N ograničenja, pa je samim tim poželjno odabrati polinom $(N - 1)$ -og reda. Međutim, ovaj pristup ima sljedeće mane:

- nije moguće definisati početne i krajnje brzine,
- sa povećanjem reda polinoma, raste i njegovo oscilatorno ponašanje, što može dovesti do trajektorija koji nisu prirodne za manipulator,
- numerička preciznost računanja koeficijenata polinoma opada sa povećanjem reda polinoma,
- dobijeni sistem jednačina može biti komplikovan za rješavanje,
- koeficijenti polinoma zavise od svih tačaka puta, ukoliko se želi promijeniti neka od tačaka, koeficijenti svih polinoma se moraju ponovo računati.

Ove mane se mogu prevazići ukoliko se jedan polinom visokog reda zamijeni sa prikladnim brojem polinoma nižeg reda, koji su kontinualni u tačkama puta.

U poglavlju 4.3.2.1 pokazano je da je interpolaciju moguće vršiti polinomom najmanje trećeg reda, jer se time omogućava kontinualnost brzina u tačkama puta. Za jednu promjenljivu zgloba, traži se funkcija $q(t)$, koja se formira od niza koji sadrži $N - 1$ polinoma trećeg reda $P_k(t)$, za $k = 1, \dots, N - 1$, koji su kontinualni zajedno sa svojim prvim izvodima. Funkcija $q(t)$ uzima vrijednosti q_k za $t = t_k$ ($k = 1, \dots, N$) i $q_1 = q_p$, $t_1 = 0$, $q_N = q_k$, $t_N = t_k$, gdje q_k predstavlja tačku željene trajektorije u trenutku $t = t_k$ (slika 4.16).



Slika 4.16. Trajektorija dobijena polinomijalnom interpolacijom niza tačaka

Razmatraće se sljedeće situacije:

- zadavanje proizvoljnih vrijednosti brzina $\dot{q}(t)$ u tačkama puta,
- zadavanje brzina u tačkama puta po nekom kriterijumu,
- pojednostavljenje generisanja trajektorije.

Da bi se problem generisanja trajektorije pojednostavio, moguće je vršiti interpolaciju i polinomima manjeg reda od tri, čime se dobijaju trajektorije koje prolaze kroz okolinu tačaka puta u zadatim vremenskim trenucima.

1) Interpolacija polinomima uz zadate vrijednosti brzina u tačkama puta

Ovaj pristup omogućava operateru da zadaje željenu vrijednost brzine u svakoj tački puta.

Sistem jednačina čijim rješavanjem se dobijaju koeficijenti $N - 1$ polinoma trećeg reda koji vrše interpolaciju N tačaka puta se dobija uvođenjem sljedećih uslova za polinom $P_k(t)$ koji interpolira tačke q_k i q_{k+1} , za $k = 1, \dots, N - 1$:

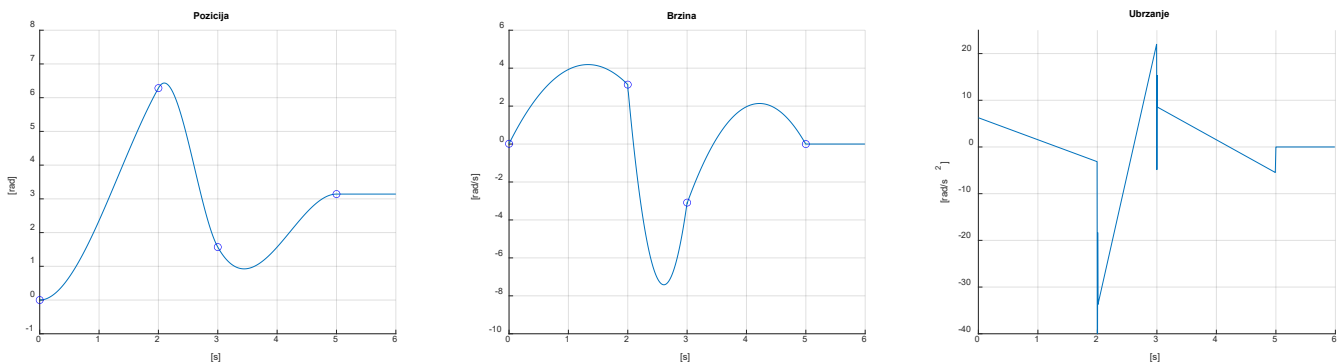
$$\begin{aligned} P_k(t_k) &= q_k, \\ P_k(t_{k+1}) &= q_{k+1}, \\ \dot{P}_k(t_k) &= \dot{q}_k, \\ \dot{P}_k(t_{k+1}) &= \dot{q}_{k+1}. \end{aligned} \quad (4.25)$$

Rezultat uvođenja uslova (4.25) je $N - 1$ sistema od četiri jednačine sa četiri nepoznata koeficijenta k -tog polinoma, koji se mogu riješiti nezavisno jedan od drugog. Početna i krajnje brzina za trajektorije se obično postavljaju na nulu ($\dot{q}_1 = \dot{q}_N$), a kontinuitet brzine u tačkama puta se dobija postavljanjem uslova

$$\dot{P}_k(t_{k+1}) = \dot{P}_{k+1}(t_{k+1}),$$

za $k = 1, \dots, N - 2$.

Slika 4.17 prikazuje promjenu pozicije, brzine i ubrzanja za sljedeće parametre: $q_1 = 0$, $q_2 = 2\pi$, $q_3 = \pi/2$, $q_4 = \pi$, $t_1 = 0$, $t_2 = 2$, $t_3 = 3$, $t_4 = 5$, $\dot{q}_1 = 0$, $\dot{q}_2 = \pi$, $\dot{q}_3 = -\pi$, $\dot{q}_4 = 0$.



Slika 4.17. Promjena pozicije, brzine i ubrzanja za slučaj zadatih vrijednosti brzina u tačkama puta

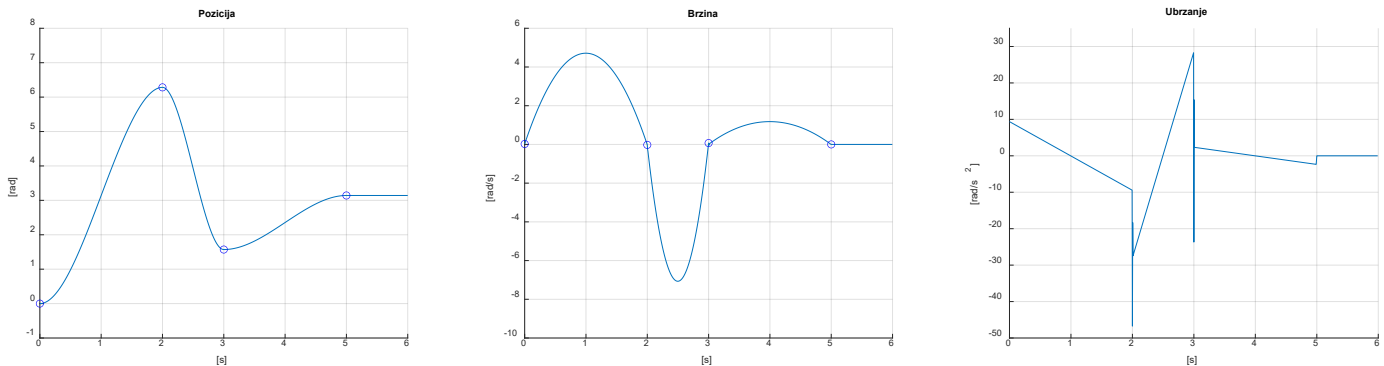
2) Interpolacija polinomima za vrijednosti brzina dobijene po određenom kriterijumu

U ovom slučaju brzina zgloba u tački puta se računa po nekom određenom kriterijumu. Kada se interpolacija između tačaka puta vrši linearnim segmentima, relativne brzine u tačkama se mogu računati po sljedećem pravilu:

$$\begin{aligned} \dot{q}_1 &= 0, \\ \dot{q}_k &= \begin{cases} 0 & \text{sgn}(v_k) \neq \text{sgn}(v_{k+1}) \\ 1/2(v_k + v_{k+1}) & \text{sgn}(v_k) = \text{sgn}(v_{k+1}) \end{cases}, \\ \dot{q}_N &= 0, \end{aligned} \quad (4.26)$$

gdje $v_k = (q_k - q_{k-1}) / (t_k - t_{k-1})$ predstavlja nagib segmenta u vremenskom intervalu $[t_{k-1}, t_k]$. Uz uvođenje uslova (4.30), određivanje koeficijenata polinoma se svodi na prethodni slučaj.

Slika 4.18 prikazuje promjenu pozicije, brzine i ubrzanja za sljedeće parametre: $q_1 = 0$, $q_2 = 2\pi$, $q_3 = \pi/2$, $q_4 = \pi$, $t_1 = 0$, $t_2 = 2$, $t_3 = 3$, $t_4 = 5$, $\dot{q}_1 = 0$, $\dot{q}_4 = \pi$. Sa slike se može uočiti da se za zadate parametre i kriterijum računanja brzina (4.30) dobijaju brzine jednake nuli u svim zadatim tačkama puta.



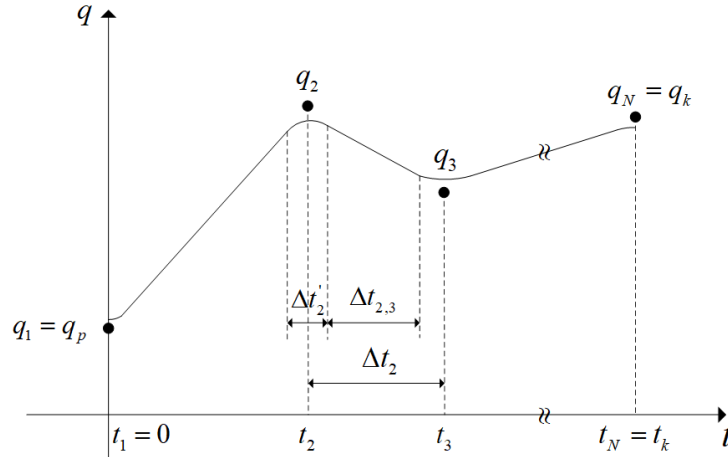
Slika 4.18. Promjena pozicije, brzine i ubrzanja za slučaj računanja vrijednosti brzina u tačkama puta

Bitno je uočiti da ni u prvom ni u drugom slučaju ubrzanje nije bilo kontinualno. Međutim, može se pokazati da se uvođenjem uslova za kontinualnost ubrzanja u tačkama puta dobija sistem jednačina koji ima više nepoznatih nego jednačina i koji se samim tim ne može riješiti. Ukoliko se želi osigurati kontinualno ubrzanje, mogu se uvesti dodatne, virtuelne međutačke ili se povećati red polinoma. U svakom slučaju, dobija se sistem jednačine koji nije nimalo jednostavan za rješavanje.

3) Interpolacija linearnim polinomima u kombinaciji sa parabolama

Pojednostavljenje u planiranju trajektorije se može postići na sljedeći način. Razmatra se slučaj kada se želi izvršiti interpolacija N tačaka puta q_1, \dots, q_N koje odgovaraju vremenskim trenucima t_1, \dots, t_N korišćenjem linearnih segmenata. Da bi se izbjegao problem prekida koji se javljaju kod prvog izvoda u vremenskim trenucima t_k , funkcija $q(t)$ mora imati parabolni profil (krivinu) u okolini t_k , što znači da je cijelokupna trajektorija sastavljena od linearnih i kvadratnih polinoma – tolerišu se prekidi koji se javljaju kod funkcije ubrzanja $\ddot{q}(t)$.

Neka je $\Delta t_k = t_{k+1} - t_k$ vremensko rastojanje između q_k i q_{k+1} i neka je $\Delta t_{k,k+1}$ vremenski interval tokom kojeg je trajektorija koja vrši interpolaciju između q_k i q_{k+1} linearna funkcija vremena. Takođe, neka je $\dot{q}_{k,k+1}$ konstantna brzina i \ddot{q}_k ubrzanje na parabolnom dijelu čije je trajanje $\Delta t'_k$. Rezultantna trajektorija je prikazana na slici 4.19. Smatra se da su vrijednosti q_k , Δt_k i $\Delta t'_k$ unaprijed poznate.



Slika 4.19. Trajektorija zgloba dobijena kombinacijom linearnih i parabolnih segmenata
Brzine i ubrzanja za međutačke trajektorije se računaju na sljedeći način:

$$\begin{cases} \dot{q}_{k-1,k} = \frac{q_k - q_{k-1}}{\Delta t_{k-1}} \\ \ddot{q}_k = \frac{\dot{q}_{k,k+1} - \dot{q}_{k-1,k}}{\Delta t'_k} \end{cases} \quad (4.27)$$

Prvom i poslednjem segmentu je potrebno posvetiti posebnu pažnju. Ustvari, ukoliko se želi zadržati podudarnost trajektorije sa prvim i poslednjim segmentom, makar dijelom vremena, rezultantna trajektorija će imati duže vrijeme trajanja – $t_N - t_1 + (\Delta t'_k + \Delta t'_N)$, a brzine $\dot{q}_{0,1} = \dot{q}_{N,N+1} = 0$ su unaprijed zadate kako bi se izračunalo početno i krajnje ubrzanje.

Sa slike 4.19 se može primjetiti da trajektorija $q(t)$ ne prolazi ni kroz jednu od zadatih tačaka q_k , već im se samo približava. U ovom slučaju, tačke puta se još nazivaju i *tačke za navođenje*, pritom, što je veće ubrzanje na parabolnim djelovima, to trajektorija prolazi bliže pomoćnim tačkama. Dakle, za date q_k , Δt_k i $\Delta t'_k$, vrijednosti $\dot{q}_{k-1,k}$ i \ddot{q}_k se računaju na osnovu (4.27), zatim se generiše niz linearnih polinoma u kombinaciji sa parabolama.

Može se primjetiti da ova tehnika zapravo predstavlja primjenu trapezoidnog profila brzine na problem interpolacije.

4.3.3. Trajektorija u radnom prostoru

Algoritam za planiranje trajektorije u radnom prostoru generiše vremensku sekvencu vrijednosti koje odgovaraju zglobnim promjenljivim $\mathbf{q}(t)$ tako da manipulator izvrši prelazak iz početne u krajnju konfiguraciju, eventualno kretanjem kroz niz međukonfiguracija. Međutim, rezultatno kretanje završnog uređaja nije lako predvidljivo, zbog nelinearne veze između spoljašnjih i unutrašnjih koordinata. Kada se želi postići da manipulator prati geometrijski zadati put u radnom prostoru, neophodno je vršiti planiranje trajektorije u istom tom prostoru. Planiranje se može vršiti ili interpolacijom niza zadatih tačaka ili generisanjem relativne trajektorije analitičkim putem.

Kod oba slučaja, vremenski sekvencu vrijednosti koordinata u radnom prostoru se, putem algoritma inverzne kinematike, prevodi u trajektorije zglobova. Zadavanje puta u radnom prostoru ili u prostoru zadatka se može izvršiti zadavanjem N tačaka koje definišu vrijednosti promjenljivih \mathbf{r} koje opisuju položaj završnog organa u prostoru spoljašnjih koordinata u vremenskim trenucima t_k , za $k = 1, \dots, N$. Slično kao kod slučaja generisanja trajektorija za pojedinačne zglobove, trajektorija u radnom prostoru se generiše određivanjem glatke vektorske funkcije koja vrši interpolaciju između zadatih tačaka puta. Takva funkcija se može odrediti primjenom bilo koje od tehnika za interpolaciju iz poglavlja 4.3.2 na svaku pojedinačnu promjenljivu iz \mathbf{r} .

Dakle, neka su date tačke puta $\mathbf{r}(t_k)$, za odgovarajuće komponente $r_i(t_k)$, $i = 1, \dots, m$ (gdje je m dimenzija radnog prostora) se može izvršiti interpolacija pomoću niza polinoma trećeg reda, kombinacije linearnih polinoma i parabola, i tako dalje. Sa druge strane, ukoliko završni uređaj treba da prati unaprijed zadatu trajektoriju, ona se mora definisati analitički. Odnosno, neophodno je osvrnuti se na primitive kretanja koje definišu geometriju puta i vremenske primitive koje definišu vremenski zakon na samom putu.

4.3.3.1. Definicija puta u prostoru

Radi definisanja *puta u prostoru* najprije je potrebno osvrnuti se na parametarsko opisivanje puteva u prostoru. Neka je \mathbf{p} vektor dimenzija 3×1 i $\mathbf{f}(\sigma)$ neprekidna vektorska funkcija definisana na intervalu $[\sigma_p, \sigma_k]$. Posmatra se jednačina

$$\mathbf{p} = \mathbf{f}(\sigma), \quad (4.28)$$

niz vrijednosti \mathbf{p} , sa σ koji uzima vrijednosti iz intervala $[\sigma_p, \sigma_k]$, se naziva *put u prostoru* [22]. Jednačina (4.28) predstavlja *parametrizaciju* puta Γ , a skalar σ predstavlja parametar puta. Kako se σ povećava, tako se tačka \mathbf{p} kreće duž puta u zadatom smjeru. Smjer puta Γ je određen parametrizacijom (4.32). Za put se kaže da je *zatvoren* kada je $\mathbf{p}(\sigma_p) = \mathbf{p}(\sigma_k)$, u suprotnom je *otvoren*.

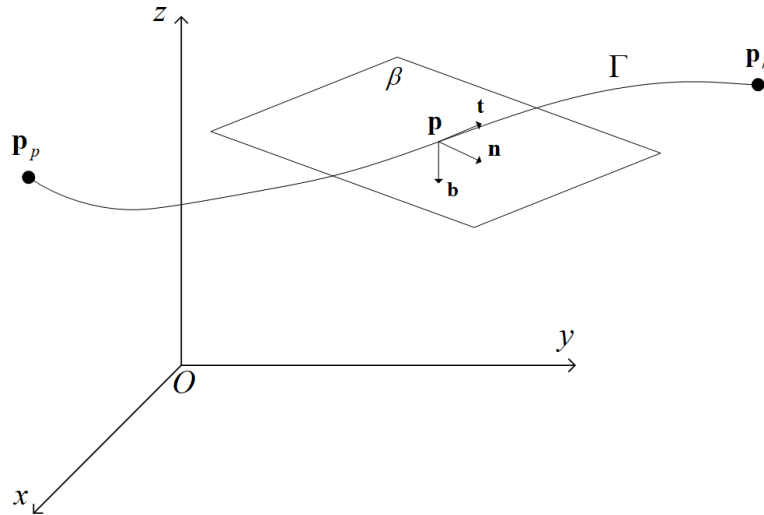
Neka je \mathbf{p}_i tačka na otvorenom putu Γ za koju je definisan smjer. Dužina luka s neke tačke \mathbf{p} je dužina luka puta Γ sa ekstremumima \mathbf{p} i \mathbf{p}_i ako \mathbf{p} dolazi nakon \mathbf{p}_i , suprotno od ovoga je da \mathbf{p} dolazi prije \mathbf{p}_i . Iz prethodne definicije se može zaključiti da svakoj vrijednosti s odgovara određena tačka puta, pa se samim tim dužina luka može koristiti kao parametar prilikom drugačije parametrizacije puta Γ :

$$\mathbf{p} = \mathbf{f}(s), \quad (4.29)$$

pri čemu je varijacija parametra s niz dužina lukova koji se dodjeljuju tačkama puta Γ .

Neka je put Γ definisan sa (4.29). Neka je \mathbf{p} tačka koja odgovara dužini luka s . Osim u pojedinim specijalnim slučajevima, \mathbf{p} dozvoljava definisanje tri jedinična vektora koja opisuju put. Orijentacija ovih vektora zavisi isključivo od geometrije puta, dok njihov smjer zavisi od smjera puta definisanog u (4.29).

Prvi jedinični vektor je *vektor tangente* i obilježava se sa \mathbf{t} . Ovaj vektor je usmjeren u smjeru koji je određen lukom s . Drugi jedinični vektor je *vektor normale* i označava se sa \mathbf{n} . Vektor normale je postavljen duž linije koja siječe \mathbf{p} pod pravim sa \mathbf{t} i leži u takozvanoj oskulirajućoj ravni β (slika 4.20). Oskulirajuća ravan predstavlja krajnju poziciju ravni koja sadrži vektor \mathbf{t} i tačku $\mathbf{p}' \in \Gamma$ kada \mathbf{p}' teži \mathbf{p} duž puta. Smjer vektora normale je takav da put Γ u blizini \mathbf{p} , u odnosu na ravan koja sadrži \mathbf{t} i koja je normalna na \mathbf{n} , leži na istoj strani kao i sami vektor \mathbf{n} . Treći jedinični vektor je *binormalni vektor* i označava se sa \mathbf{b} . Ovaj vektor je takav da je koordinatni sistem koji definišu ortovi $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ desni koordinatni sistem (slika 4.20). Na kraju je bitno napomenuti da nije uvijek moguće jednoznačno definisati ovakav sistem.



Slika 4.20. Parametrizacija puta u prostoru

Može se pokazati da su tri jedinična vektora koja opisuju put povezana putem dužine luka s parametrizacije puta Γ . Konkretno, može se zapisati:

$$\begin{aligned} \mathbf{t} &= \frac{d\mathbf{p}}{ds}, \\ \mathbf{n} &= \frac{1}{\left\| \frac{d^2\mathbf{p}}{ds^2} \right\|} \frac{d^2\mathbf{p}}{ds^2}, \\ \mathbf{b} &= \mathbf{t} \times \mathbf{n}. \end{aligned} \tag{4.30}$$

U nastavku su date najčešće parametrizacije puta koje su korisne za generisanje trajektorije u radnom prostoru.

1) Pravolinijski put

Posmatra se linearni segment koji spaja tačku \mathbf{p}_p sa tačkom \mathbf{p}_k . Parametrizacija ovog linearnog segmenta je

$$\mathbf{p}(s) = \mathbf{p}_p + \frac{s}{\|\mathbf{p}_k - \mathbf{p}_p\|} (\mathbf{p}_k - \mathbf{p}_p). \quad (4.31)$$

Iz (4.35) se može primjetiti da je $\mathbf{p}(0) = \mathbf{p}_p$ i $\mathbf{p}(\|\mathbf{p}_k - \mathbf{p}_p\|) = \mathbf{p}_k$. Stoga je smjer puta Γ čija je parametrizacija data izrazom (4.35) od tačke \mathbf{p}_p do tačke \mathbf{p}_k . Diferenciranjem (4.35) po parametru s dobija se:

$$\begin{aligned} \frac{d\mathbf{p}}{ds} &= \frac{1}{\|\mathbf{p}_k - \mathbf{p}_p\|} (\mathbf{p}_k - \mathbf{p}_p), \\ \frac{d^2\mathbf{p}}{ds^2} &= \mathbf{0}. \end{aligned} \quad (4.32)$$

U ovom slučaju nije moguće jednoznačno definisati koordinatni sistem $(\mathbf{t}, \mathbf{n}, \mathbf{b})$.

2) Kružnica

Posmatra se kružnica Γ u trodimenzionalnom prostoru. Prije parametrizacije kružnice, neophodno je uvesti njegove značajne parameter. Neka se krug definiše pomoću sljedećih parametara (slika 4.21):

- jedinični vektor ose kružnice \mathbf{r} ,
- vektor pozicije tačke na osi kružnice \mathbf{d} ,
- vektor pozicije tačke na kružnici \mathbf{p}_p .

Pomoću ovih parametara se može naći vektor pozicije centra kruga \mathbf{c} . Ako je $\boldsymbol{\delta} = \mathbf{p}_p - \mathbf{d}$, tada, kako tačka \mathbf{p}_p ne bi bila osi kružnice, tj. kako se kružnica ne bi pretvorila u tačku, mora važiti jednakost:

$$|\boldsymbol{\delta}^T \mathbf{r}| < \|\boldsymbol{\delta}\|,$$

i u tom slučaju je

$$\mathbf{c} = \mathbf{d} + (\boldsymbol{\delta}^T \mathbf{r}) \mathbf{r}. \quad (4.33)$$

Sada se želi naći parametrizacija kružnice kao funkcije dužine luka. Parametrizacija se znatno pojednostavljuje adekvatnim izborom referentnog koordinatnog sistema. Posmatra se sistem $O'x'y'z'$, pri čemu se O' nalazi u centru kružnice, osa x' je usmjerena u smjeru $\mathbf{p}_p - \mathbf{c}$, smjer ose z' je određen vektorom \mathbf{r} , a osa y' se određuje na osnovu prethodne dvije ose. Kada se izrazi u referentnom koordinatnom sistemu, parametrizacija kružnice je:

$$\mathbf{p}'(s) = \begin{bmatrix} \rho \cos(s/\rho) \\ \rho \sin(s/\rho) \\ 0 \end{bmatrix}, \quad (4.34)$$

gdje je $\rho = \|\mathbf{p}_p - \mathbf{c}\|$ poluprečnik kružnice, a tačka \mathbf{p}_p je uzeta za početak dužine luka. Za neki drugi referentni koordinatni sistem, parametrizacija puta će biti

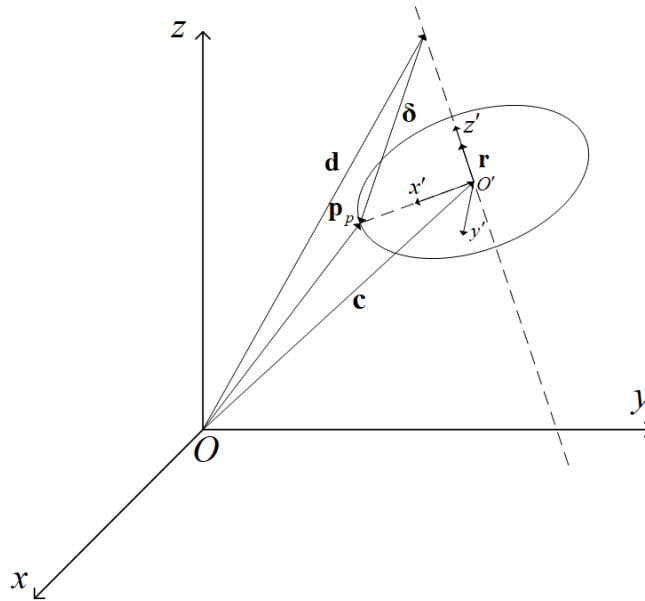
$$\mathbf{p}(s) = \mathbf{c} + \mathbf{R}\mathbf{p}'(s), \quad (4.35)$$

gdje je \mathbf{c} izražen u koordinatnom sistemu $Oxyz$ i \mathbf{R} matrica rotacije koordinatnog sistema $O'x'y'z'$ u odnosu na $Oxyz$.

Diferenciranjem (4.35) po parametru s dobija se:

$$\frac{d\mathbf{p}}{ds} = \mathbf{R} \begin{bmatrix} -\sin(s/\rho) \\ \cos(s/\rho) \\ 0 \end{bmatrix}, \quad (4.36)$$

$$\frac{d^2\mathbf{p}}{ds^2} = \mathbf{R} \begin{bmatrix} -\cos(s/\rho)/\rho \\ -\sin(s/\rho)/\rho \\ 0 \end{bmatrix}.$$



Slika 4.21. Parametrizacija kružnice u prostoru

4.3.3.2. Pozicija

Neka je \mathbf{r} vektor spoljašnjih koordinata koji opisuje položaj završnog uređaja manipulatora. Generisanje trajektorije u radnom prostoru znači određivanje funkcije $\mathbf{r}(t)$ koja vodi završni uređaj od početnog do krajnjeg položaja za vrijeme t_k . Najprije se razmatra pozicija završnog organa.

Neka je $\mathbf{p} = \mathbf{f}(s)$ vektor parametrizacije puta Γ , dimenzija 3×1 , funkcija dužine luka s – koordinatni početak sistema hvataljke vrši pomjeranje od \mathbf{p}_p do \mathbf{p}_k za vrijeme t_k . Radi jednostavnosti, neka je početak dužine luka u tački \mathbf{p}_p i neka je smjer Γ od \mathbf{p}_p do \mathbf{p}_k . Dužina luka se mijenja od vrijednosti $s = 0$ u $t = 0$ do vrijednosti $s = s_k$ (dužina cijelog puta) za $t = t_k$. Zakon promjene u vremenu dužine luka je dat funkcijom $s(t)$. Za određivanja analitičkog izraza za $s(t)$ se

može primijeniti bilo koja od tehnika izloženih u poglavlju 4.3.2, kao na primjer tehnika interpolacije korišćenjem polinoma trećeg reda.

Korisno je dati neke napomene o promjeni \mathbf{p} u vremenu na putu Γ , za definisani zakon promjene $s(t)$. Brzina u tački \mathbf{p} je data izviodom po vremenu

$$\dot{\mathbf{p}} = \frac{d\mathbf{p}}{ds} \frac{ds}{dt} = \dot{s}\mathbf{t},$$

gdje je \mathbf{t} vektor tangente puta u tački \mathbf{p} iz (4.34), a \dot{s} je magnituda vektora brzine u odnosu na tačku \mathbf{p} , čiji znak zavisi od smjera $\dot{\mathbf{p}}$ duž vektora \mathbf{t} . Magnituda $\dot{\mathbf{p}}$ u trenutku $t = 0$ je jednaka nuli, a dalje varira sa parabolnim ili trapezodnim profilom, i na kraju se vraća na nulu u trenutku $t = t_k$.

Kao prvi primjer posmatra se segment koji povezuje tačku \mathbf{p}_p sa tačkom \mathbf{p}_k . Parametrizacija ovog puta je data jednačinom (4.31). Brzina i ubrzanje \mathbf{p} iz (4.31) su:

$$\begin{aligned} \dot{\mathbf{p}} &= \frac{\dot{s}}{\|\mathbf{p}_k - \mathbf{p}_p\|} (\mathbf{p}_k - \mathbf{p}_p) = \dot{s}\mathbf{t}, \\ \ddot{\mathbf{p}} &= \frac{\ddot{s}}{\|\mathbf{p}_k - \mathbf{p}_p\|} (\mathbf{p}_k - \mathbf{p}_p) = \ddot{s}\mathbf{t}. \end{aligned} \quad (4.37)$$

Kao drugi primjer posmatra se kružnica Γ u prostoru. Iz parametrizacije kružnice date izrazom (4.35), brzina i ubrzanje u tački kružnice \mathbf{p} će biti:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{R} \begin{bmatrix} -\dot{s} \sin(s/\rho) \\ \dot{s} \cos(s/\rho) \\ 0 \end{bmatrix}, \\ \ddot{\mathbf{p}} &= \mathbf{R} \begin{bmatrix} -\dot{s}^2 \cos(s/\rho)/\rho - \ddot{s} \sin(s/\rho) \\ -\dot{s}^2 \sin(s/\rho)/\rho + \ddot{s} \cos(s/\rho) \\ 0 \end{bmatrix}. \end{aligned} \quad (4.38)$$

Na osnovu (4.38) se može zaključiti da vektor brzine ima isti smjer kao i vektor tangente \mathbf{t} , dok je kod vektora ubrzanja prva komponenta u smjeru vektora normale \mathbf{n} i predstavlja normalno ubrzanje, a druga komponenta je u smjeru vektora \mathbf{t} i predstavlja tangencijalno ubrzanje.

Na kraju, posmatra se put koji se sastoji od $N + 1$ tačaka, $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_N$, koje povezuje N segmenata. Parametrizacija cjelokupnog puta se može zapisati na sljedeći način:

$$\mathbf{p} = \mathbf{p}_0 + \sum_{j=1}^N \frac{s_j}{\|\mathbf{p}_j - \mathbf{p}_{j-1}\|} (\mathbf{p}_j - \mathbf{p}_{j-1}). \quad (4.39)$$

U (4.39) s_j je dužina luka j -tog segemnta puta, koji povezuje tačku \mathbf{p}_{j-1} sa tačkom \mathbf{p}_j , i definiše se kao:

$$s_j(t) = \begin{cases} 0 & 0 \leq t \leq t_{j-1} \\ s'_j(t) & t_{j-1} < t \leq t_j, \\ \|\mathbf{p}_j - \mathbf{p}_{j-1}\| & t_j < t \leq t_k \end{cases} \quad (4.40)$$

gdje su $t_0 = 0$ i $t_0 = t_N$ početni i krajnji trenutak trajektorije, respektivno, t_j je vremenski trenutak koji odgovara tački \mathbf{p}_j i $s'_j(t)$ je neka analitička funkcija (npr. polinom trećeg reda) čija se vrijednost mijenja od $s'_j = 0$ u trenutku $t = t_{j-1}$ do $s'_j = \|\mathbf{p}_j - \mathbf{p}_{j-1}\|$ u trenutku $t = t_j$. Brzina i ubrzanje se dobijaju diferenciranjem jednačina iz (4.42) po vremenu, odnosno:

$$\begin{aligned} \dot{\mathbf{p}} &= \sum_{j=1}^N \frac{\dot{s}_j}{\|\mathbf{p}_j - \mathbf{p}_{j-1}\|} (\mathbf{p}_j - \mathbf{p}_{j-1}) = \sum_{j=1}^N \dot{s}_j \mathbf{t}_j, \\ \ddot{\mathbf{p}} &= \sum_{j=1}^N \frac{\ddot{s}_j}{\|\mathbf{p}_j - \mathbf{p}_{j-1}\|} (\mathbf{p}_j - \mathbf{p}_{j-1}) = \sum_{j=1}^N \ddot{s}_j \mathbf{t}_j, \end{aligned} \quad (4.41)$$

gdje je \mathbf{t}_j vektor tangente j -tog segmenta.

Zbog prekida koji se javljaju kod prvog izvoda trajektorije u tačkama puta između dva neporavnata segmenta, manipulator se mora zaustaviti u tim tačkama, a zatim nastaviti kretanje duž narednog segmenta. Ukoliko se zanemari uslov da manipulator mora striktno proći kroz svaku tačku puta, moguće je izbjeći zaustavljanje manipulatora povezivanjem susjednih neporavnatih segmenata u okolini tačaka puta, koje u tom slučaju postaju *tačke za navođenje u radnom prostoru*. Ovim pristupom se u najmanju ruku obezbjeđuje neprekindost prvog izvoda.

4.3.3.3. Orijehtacija

Sada će se ukratko razmotriti orijentacija završnog uređaja. Orijehtacija se najčešće zadaje u vidu vremenski promjenljive matrice rotacije koordinatnog sistema završnog organa u odnosu na bazni koordinatni sistem. Kao što je već poznato, tri kolone matrice rotacije predstavljaju tri jedinična vektora sistema hvataljke u odnosu na bazni sistem. Međutim, generisanje trajektorije linearnom interpolacijom između početne i krajnje orijentacije jediničnih vektora \mathbf{n} , \mathbf{s} i \mathbf{a} ne garantuje njihovu međusobnu ortonormalnost u svakom trenutku kretanja.

Neka su $\mathbf{R}_{0,p}$ i $\mathbf{R}_{0,k}$, respektivno, matrice rotacije početnog sistema trajektorije $O_{pxpy_pz_p}$ i krajnjeg sistema trajektorije O_{kxky_kzk} u odnosu na bazni sistem. Matrica rotacije između početnog i krajnjeg sistema se može dobiti iz jednakosti $\mathbf{R}_{0,k} = \mathbf{R}_{0,p}\mathbf{R}_{p,k}$, odnosno:

$$\mathbf{R}_{p,k} = \mathbf{R}_{0,p}^{-1} \mathbf{R}_{0,k} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (4.42)$$

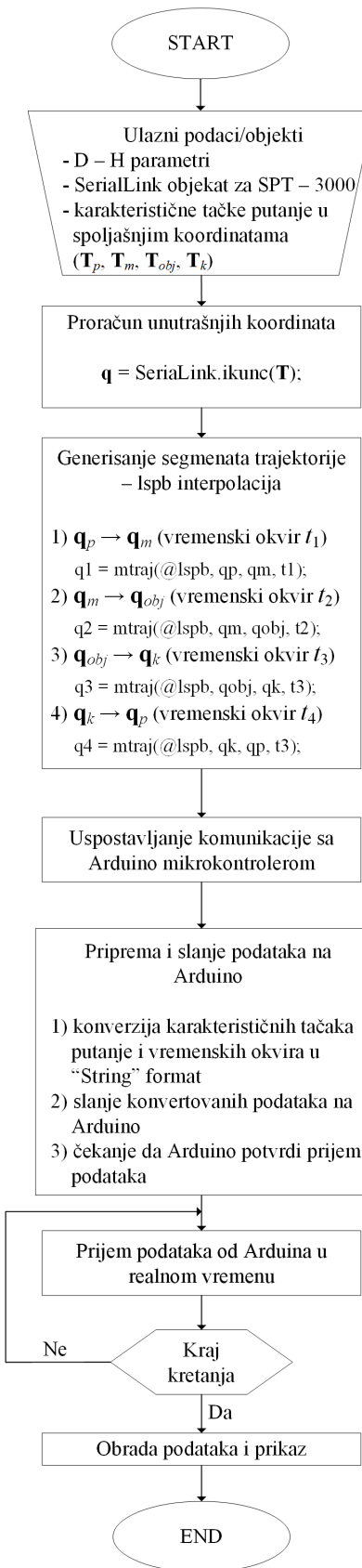
Ukoliko se definiše matrica $\mathbf{R}_l(t)$, koja opisuje prelaz od $\mathbf{R}_{0,p}$ do $\mathbf{R}_{0,k}$, tada mora važiti $\mathbf{R}_l(0) = \mathbf{I}$ i $\mathbf{R}_l(t_k) = \mathbf{R}_{p,k}$.

Konačno, kada se odrede put i trajektorija u prostoru spoljašnjih koordinata u vidu $\mathbf{p}(t)$ i $\mathbf{R}_A(t)$, može se primijeniti algoritam inverzne kinematike kako bi se našle odgovarajuće trajektorije u prostoru zglobova $\mathbf{q}(t)$.

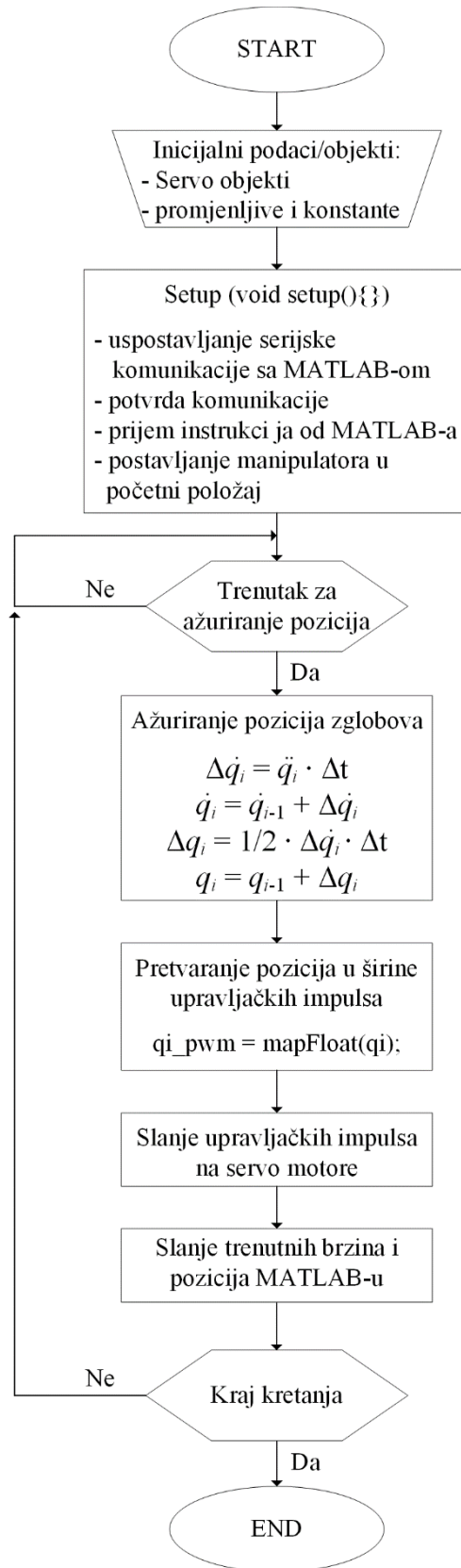
5. Eksperiment

U drugom poglavlju rada su uvedeni osnovni pojmovi i činjenice vezani za industrijske manipulacione robote, sa osvrtom sa SPT – 3000. U trećem poglavlju je izložena problematika formiranja kinematičkog modela robota, uz objašnjenja postupaka rješavanja direktnog i inverznog kinematičkog problema. U četvrtom poglavlju su izloženi principi i algoritmi upravljanja, kao i osnovne metodologije planiranja trajektorije manipulatora.

Shodno tome, u ovom poglavlju će se vršiti primjena principa koji su do sada izloženi u radu. Nakon obrazloženja radnog zadatka, izvršiće se generisanje odgovarajuće trajektorije manipulatora u MATLAB-u. Zatim će se neophodne informacije o trajektoriji manipulatora proslijediti Arduino mikrokontroleru. Nakon obrade podataka dobijenih od MATLAB-a, Arduino će započeti izvršavanje radnog zadatka tako što će, na osnovu dobijenih informacija, vršiti generisanje upravljačkih impulsa koji će prosljeđuju servo motorima, koji vrše odgovarajuće pomjeranje. U toku izvršenja radnog zadatka, Arduino u realnom vremenu šalje realne trajektorije zglobova MATLAB-u. Na kraju, nakon izvršenja radnog zadatka, vrši se obrada i analiza dobijenih rezultata. Blok dijagram MATLAB programa je dat na slici 5.1, dok je blok dijagram Arduino programa dat na slici 5.2.



Slika 5.1. Blok dijagram MATLAB programa



Slika 5.2. Blok dijagram Arduino programa

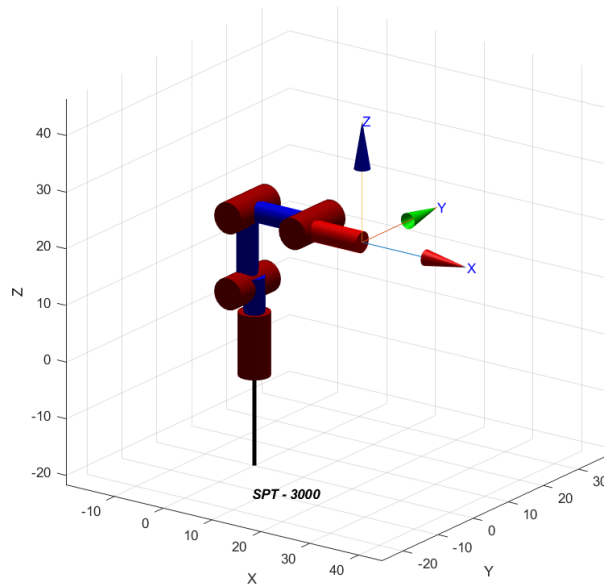
5.1. Definicija i rješavanje radnog zadatka

Ukoliko se uzme u obzir konfiguracija i arhitektura SPT – 3000 manipulatora, jasno se nameće činjenica je njegova osnovna namjena transport radnih predmeta sa jednog mjesta na drugo. Imajući to na umu, definicija radnog zadatka je sljedeća: *Manipulator SPT – 3000 preuzima radni predmet koji se nalazi na određenoj poziciji u njegovom radnom prostoru i prenosi ga na novu poziciju, koja se takođe nalazi unutar njegovog radnog prostora.*

Na osnovu definisanog radnog zadatka se mogu definisati neophodni koraci kako bi se isti izvršio. Najprije je potrebno definisati trajektoriju u radnom prostoru, tj. u prostoru spoljašnjih koordinata. Trajektorija manipulatora sadrži nekoliko „karakterističnih položaja”: *početni položaj*, *međupoložaj* ili položaj koji omogućava adekvatan prilazak radnom predmetu, *položaj radnog predmeta* i *položaj za odlaganje predmeta*. Na osnovu ova četiri položaja se interpolacijom generiše trajektorija manipulatora.

Međutim, ukoliko se trajektorija generiše u prostoru spoljašnjih koordinata, istu je neophodno prevesti u prostor zglobova, kako bi se moglo vršiti upravljanje aktuatorima u zglobovima. Prevođenje trajektorije iz prostora spoljašnjih u prostor unutrašnjih koordinata podrazumijeva rješavanje inverznog kinematičkog problema. Problem kod prevođenja trajektorije u prostor zglobova predstavlja činjenica da je neophodno riješiti inverzni kinematički zadatak za svaku pojedinačnu tačku trajektorije, što je računarski i vremenski veoma zahtjevno. Kako su kod definisanog radnog zadatka bitni samo početni i krajnji položaj predmeta, generisanje trajektorije se može izvršiti na sljedeći način. Rješavanje inverznog kinematičkog problema se vrši samo za ključne položaje trajektorije, čime se isti prevode u prostor unutrašnjih koordinata. Zatim se trajektorija generiše interpolacijom između ključnih položaja manipulatora u prostoru unutrašnjih koordinata. Na ovaj način se značajno skraćuje vrijeme potrebno za rješavanje radnog zadatka, a pritom se ne zanemaruju njegove ključne stavke.

Kako bi se riješio inverzni kinematički zadatak za ključne položaje manipulatora, najprije je potrebno formirati kinematički model SPT – 3000 manipulatora u MATLAB-u. Model se formira na osnovu D – H parametara datih u tabeli 3.1, uz pomoć MATLAB Robotic Toolbox-a. Robotic Toolbox (RTB) predstavlja skup funkcija i klasa u MATLAB-u koje su koriste za proučavanje kinematike, dinamike i generisanje trajektorije za antropomorfni i druge tipove manipulatora [23]. Jedna od najbitnih funkcionalnosti RTB-a je provjera ispravnosti zadatog položaja, odnosno prilikom zadavanja svake položaja u radnom prostoru manipulatora, provjerava se da li postoji rješenje IKP, odnosno da li manipulator može dostići zadati položaj. Ova funkcionalnost je veoma značajna za SPT – 3000, koji posjeduje samo jedan stepen slobode koji vrši orijentaciju hvataljke, što značajno ograničava broj mogućih položaja koje manipulator može zauzeti. SPT – 3000 je predstavljen u MATLAB-u kao serijski kinematički lanac, odnosno kao objekat klase SerialLink. Slika 5.3 prikazuje kinematički model SPT – 3000 u MATLAB-u, pri čemu se SPT – 3000 nalazi u početnom položaju.



Slika 5.3. Kinematički model SPT – 3000 u MATLAB-u

Svaki „karakteristični položaj” se može zapisati u spoljašnjim i unutrašnjim koordinatama. Zadavanje položaja se vrši u MATLAB-u u vidu matrica homogenih transformacija, za koje se zatim rješava IKP korišćenjem funkcija iz RTB-a. Početni položaj robota, tj. položaj iz kojeg robot započinje kretanje se u spoljašnjim i unutrašnjim koordinatama može zapisati kao:

$$\mathbf{T}_p = \begin{bmatrix} 1 & 0 & 0 & 23,7 \\ 0 & 1 & 0 & -2,208 \\ 0 & 0 & 1 & 23,24 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{IKP}} \mathbf{q}_p = [0^\circ \quad 90^\circ \quad -90^\circ \quad 0^\circ]^T. \quad (5.1)$$

Međupoložaj, tj. položaj manipulatora između početnog položaja i položaja radnog predmeta je predstavljen koordinatama:

$$\mathbf{T}_m = \begin{bmatrix} 0 & 0 & 1 & 20 \\ 0 & 1 & 0 & -2,208 \\ -1 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{IKP}} \mathbf{q}_m = [0^\circ \quad 52,18^\circ \quad -46,44^\circ \quad -95,737^\circ]^T. \quad (5.2)$$

Radni predmet koji se koristi za potrebe eksperimenta je kocka, dimenzija 3cm×3cm×3cm, izrađena od PLA materijala. Da bi robot izvršio prihvat radnog predmeta, koordinatni sistem hvataljke se mora poklopiti sa koordinatnim sistemom predmeta (slika 5.4). Koordinatni početak koordinatnog sistema radnog predmeta je vezan za tačku koja se u centru jedne od stranica kocke. Odabrani položaj radnog predmeta u odnosu na bazni koordinatni sistem je: T_p ,

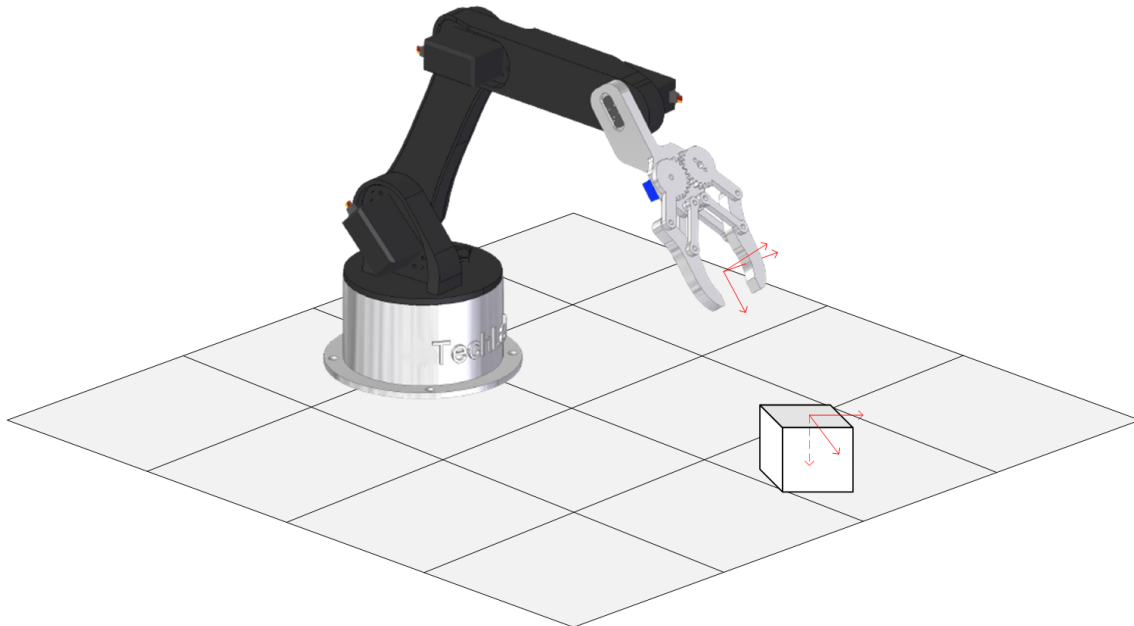
$$\mathbf{T}_{obj} = \begin{bmatrix} 0 & 0 & 1 & 20 \\ 0 & 1 & 0 & -2,208 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

odakle se, nakon rješavanja inverznog kinematičkog problema, dobijaju koordinate zglobova koje odgovaraju početnom položaju radnog predmeta:

$$\mathbf{q}_{obj} = [0^\circ \quad 47,227^\circ \quad -69,05^\circ \quad -68,17^\circ]^T. \quad (5.3)$$

Položaj za odlaganje/ispuštanje radnog predmeta je:

$$\mathbf{T}_k = \begin{bmatrix} 0.7071 & -0.7071 & 0 & 25 \\ 0.7071 & 0.7071 & 0 & 25 \\ 0 & 0 & 1 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{IKP}} \mathbf{q}_k = [47,42^\circ \quad 21,99^\circ \quad 0^\circ \quad -21,91^\circ]^T. \quad (5.4)$$



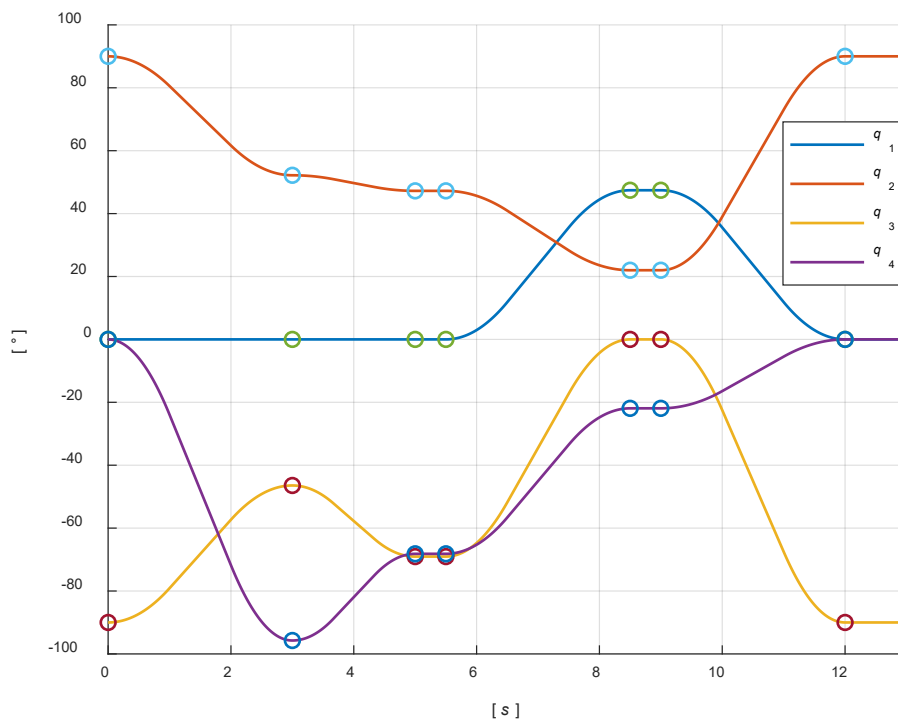
Slika 5.4. Prihvat radnog predmeta

Hronologija kretanja manipulatora je sljedeća: manipulator započinje kretanje iz početnog položaja i kreće se do međupoložaja, zatim od međupoložaja do položaja predmeta, vrši se prihvat predmeta, zatim se predmet prenosi do položaja za odlaganje, vrši se odlaganje/ispuštanje predmeta, i konačno se manipulator vraća u početni položaj. Na osnovu izložene hronologije kretanja može se izvršiti interpolacija između položaja robota u prostoru unutrašnjih koordinata, odnosno može se generisati trajektorija.

U poglavlju 4.3.2. su izložene metode generisanja trajektorije manipulatora u prostoru unutrašnjih koordinata – interpolacija polinomima trećeg reda, interpolacija polinomima petog reda i interpolacija kombinacijom linearnih polinoma sa parabolnim segmentima. Interpolacija polinomima trećeg i petog reda omogućava zadavanje početnih uslova za brzine u ubrzanja u

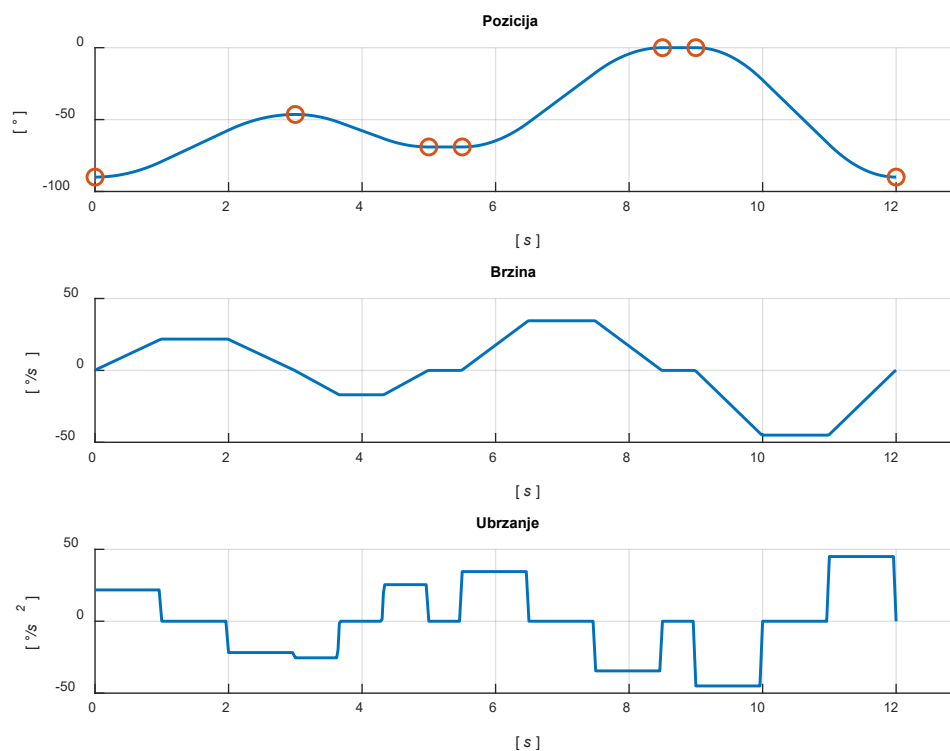
zadatih tačkama trajektorije, ali i dovodi do pojave preskoka u situacijama ukoliko neka od zadatih brzina ima vrijednost različitu od nule. Još jedan problem koji se javlja kod interpolacije polinomima trećeg i višeg reda je činjenica da se zglob većinu vremena kreće brzinom koja je znatno manja od njegove maksimalne brzine. Iz navedenih razloga se bira *hibridna trajektorija*, odnosno interpolacija kombinacijom linearnih i parabolnih segmenata.

Slika 5.5 prikazuje zadate trajektorije zglobova SPT – 3000 manipulatora, generisane u MATLAB-u korišćenjem Robotic Toolbox-a. Ukupno vrijeme trajanja kretanja je 12 sekundi, tokom kojeg zglobovi aktivno vrše kretanje 11 sekundi, uz dvije pauze u kretanju od po 0,5 sekundi, tokom kojih se vrši hvatanje i otpuštanje radnog predmeta. Promjena, odnosno ažuriranje pozicija se vrši sa frekvencijom od 50Hz, odnosno na svakih 20 milisekundi. Frekvencija ažuriranja pozicije je odabrana tako da se poklapa sa frekvencijom upravljačkih impulsa kojima Arduino mikrokontroler upravlja servo motorima u zglobovima manipulatora.



Slika 5.5. MATLAB trajektorije zglobova SPT – 3000 manipulatora

Slika 5.6 prikazuje generisanu trajektoriju zgloba lakta, uz odgovarajuću brzinu i poziciju. Sa slike se može uočiti da se, za primijenjenu metodu interpolacije, dobija trapezoidni profil brzine, za koji odgovarajuće ubrzanje nije kontinualno, što je za dati radni zadatak prihvatljivo, ali ne i slučajevima kada se zahtijeva izvođenje kompleksnijih operacija. Bitno je napomenuti da se, nakon generisanja trajektorija zglobova, mora izvršiti provjera maksimalnih generisanih brzina za svaki pojedinačni zglob. Ukoliko je neka od generisanih brzina veća od maksimalnih brzina servo motora u zglobova, neophodno je ponovo generisati trajektorije, uz definisanje maksimalnih brzina za svaki zglob.



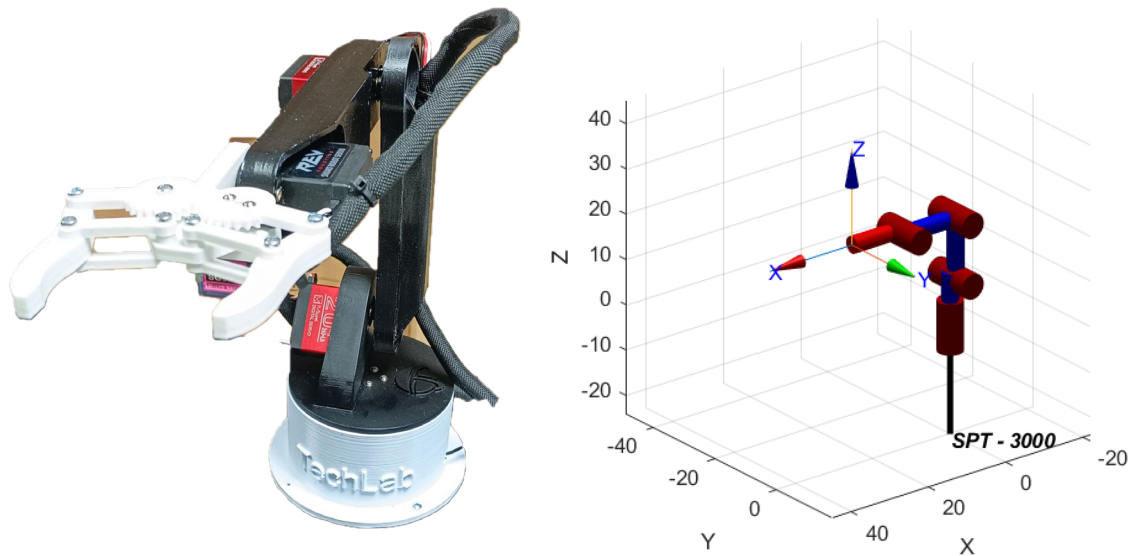
Slika 5.6. Zadana promjena pozicije, brzine i ubrzanja zgloba lakta

5.2. Tok i rezultati eksperimenta

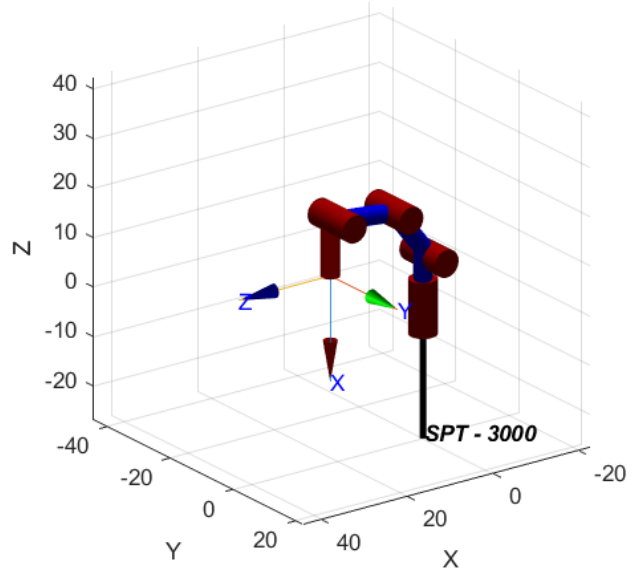
Nakon generisanja trajektorija ostaje da se osnovni podaci o trajektorijama zglobova pošalju na Arduino mikrokontroler i da se na osnovu njih izvrši upravljanje manipulatorom po predviđenoj trajektoriji. Razlog slanja samo osnovnih informacija o trajektoriji je ograničenje programske memorije Arduino mikrokontrolera (32 kilobajta) i ograničenje koje nameće sama komunikacija, koja se obavlja putem serijskog porta koji ima ograničen protok podataka i privremeno skladište podataka čiji je kapacitet ograničen na samo 64 bajta. Podaci koji se putem serijske komunikacije šalju Arduino mikrokontroleru su: „karakteristični položaji” manipulatora (koordinate u prostoru zglobova) i vremena za koja je potrebno izvršiti pojedinačna kretanja između zadatih položaja.

Nakon prijema podataka sa računara, Arduino mikrokontroler u realnom vremenu vrši računanje pozicija zglobova, odnosno servo motora, konvertuje trenutne pozicije u širine upravljačkih impulsa, koje zatim šalje na servo motore koji vrše odgovarajuće kretanje.

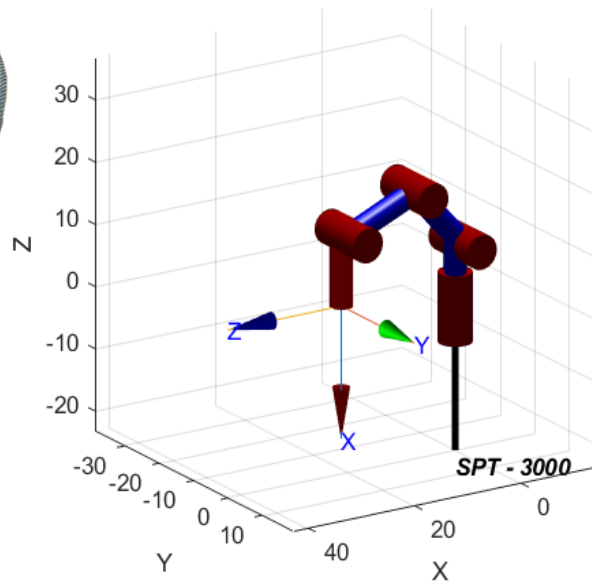
Tok eksperimenta je sljedeći. SPT – 3000 se najprije postavlja u početni položaj, kojem odgovaraju koordinate date relacijom (5.1). Slika 5.7 prikazuje početni položaj na dva načina: pomoću modela formiranog u MATLAB-u korišćenjem RTB-a, kao i pozicioniranje od strane realnog SPT – 3000 manipulatora. Manipulator se zatim, u vremenskom rasponu od tri sekunde kreće od početnog položaja do međupoložaja, datog izrazom (5.2), odnosno pozicionira se direktno iznad predmeta kako bi izvršio prihvata (slika 5.8). Dalje, u naredne dvije sekunde, manipulator prilazi objektu, ili manipulator se kreće od međupoložaja do položaja radnog predmeta, koji je dat izrazom (5.3). Manipulator ostaje u položaju koji se poklapa sa položajem radnog predmeta pola sekunde, kako bi se izvršio prihvata predmeta (slika 5.9). Nakon prihvata radnog predmeta, SPT – 3000 se u vremenskom rasponu od tri sekunde kreće od položaja radnog predmeta do krajnjeg položaja, čije su koordinate date relacijom (5.4). Manipulator u ovom položaju takođe ostaje pola sekunde, kako bi izvršilo odlaganje/ispuštanje radnog predmeta (slika 5.10). Po odlaganju radnog predmeta, SPT – 3000 se vraća u početni položaj, čime je radni zadatak završen.



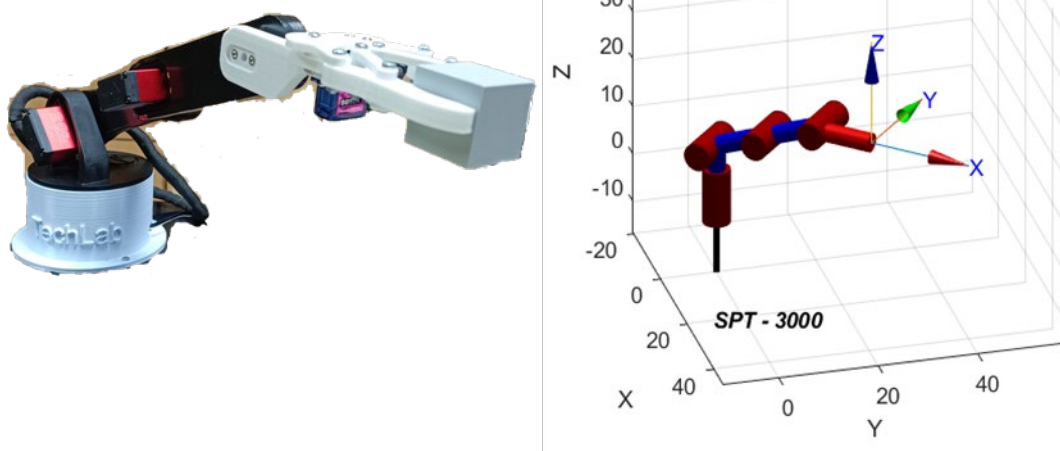
Slika 5.7. Početni položaj SPT – 3000 manipulatora



Slika 5.8. Međupoložaj SPT – 3000 manipulatora

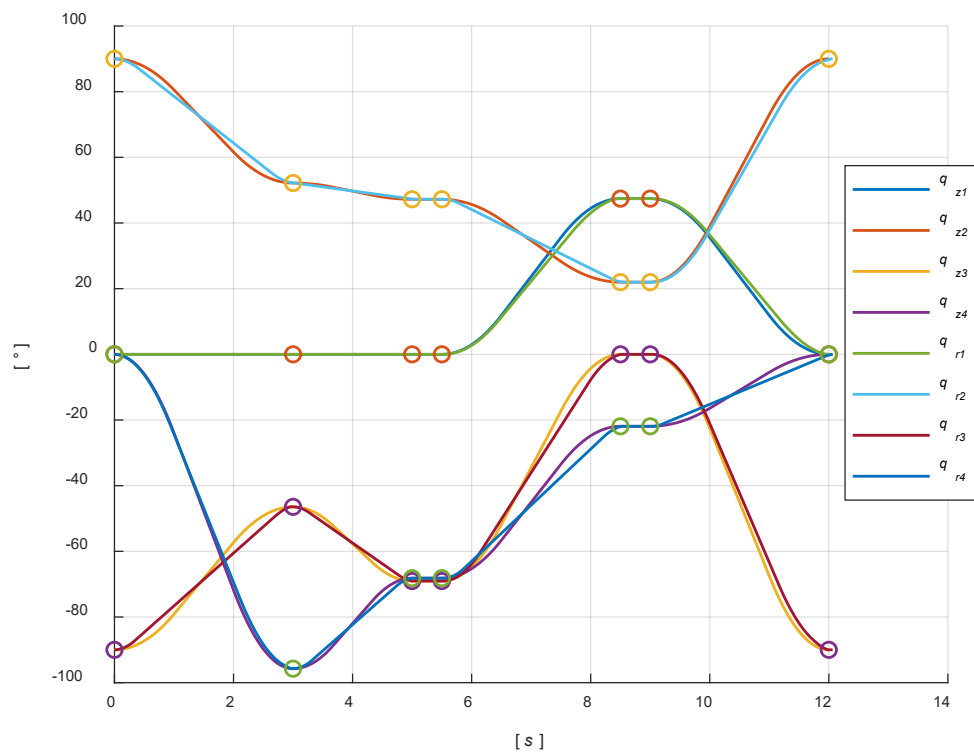


Slika 5.9. Položaj manipulatora koji odgovara položaju objekta



Slika 5.10. Položaj za ispuštanje radnog predmeta

Sada će se posmatrati i analizirati rezultati eksperimenta. Slika 5.11 prikazuje zadate trajektorije zglobova i realizovane, odnosno trajektorije zglobova generisane od strane Arduino mikrokontrolera. Sa slike se može uočiti da je realne trajektorije prate zadate trajektorije, pri čemu je najbitnija činjenica da realne trajektorije prolaze kroz karakteristične položaje, odnosno da se manipulator pozicionira u svim karakterističnim tačkama u toku svog kretanja.



Slika 5.11. Poređenje zadatih i realnih trajektorija zglobova

Kako bi se vršilo poređenje zadatih i realnih trajektorija i sa aspekta brzina i ubrzanja zglobova, posmatra se slika 5.12, na su prikazane pozicija, brzina i ubrzanje zgloba lakta. Poređenjem zadate i realne pozicije uočava se da realna trajektorija znatno „linearnija” u odnosu na zadatu trajektoriju, što znači da se zglob lakta veći dio puta kreće konstantnom brzinom. Ovaj zaključak se može izvesti i poređenjem zadate i realizovane brzine. Razlog za to je činjenica da se manipulator prilikom kretanja između dva karakteristična položaja, na dijelu gdje je konstantna brzina, kreće brzinom manjom od zadate.

Razlog za kretanje brzinom koja je manja od zadate je posljedica toga što se maksimalna brzina zgloba manipulatora na mikrokontroleru zadaje samo jednom, na početku programa, pa se na osnovu zadate maksimalne brzine, distance koju je potrebno preći i vremena koje je dato za prelaženje vrši proračun brzine linearnog segmenta. Arduino mikrokontroler, usljed nedostatka memorije za skladištenje podataka, vrši računanje brzina i pozicija zglobova u realnom vremenu. Razlog za zadavanje maksimalne brzine isključivo na početku programa je činjenica da bi se, prilikom svakog zadavanja nove maksimalne brzine, moralo vršiti preračunavanje koeficijenata, čime se trošilo znatno više procesorskog vremena i usporavao sistem. Takođe je bitno napomenuti da Arduino mikrokontroler, usljed nedostatka memorije za skladištenje podataka, vrši računanje brzina i pozicija zglobova u realnom vremenu.

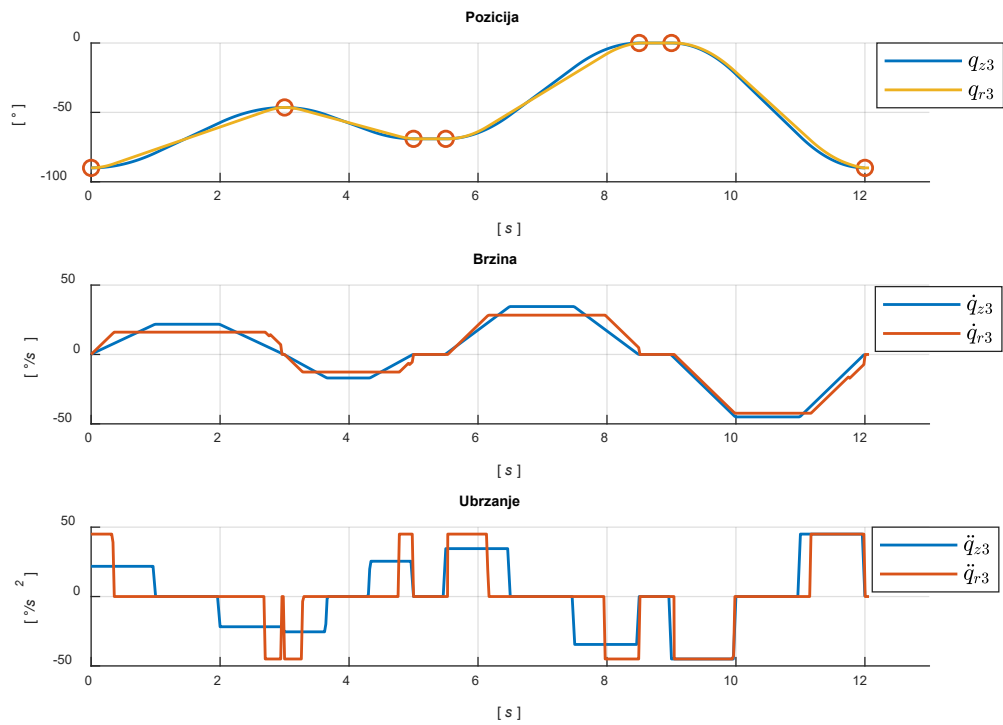
Navedene razlike između zadate i realne brzine se preslikavaju i na ubrzanja. Posmatranjem zadatag i realnog ubrzanja zgloba lakta zaključuje se da je, u toku kretanja između dvije zadate koordinate, realno ubrzanje u kraćim intervalima uzima nenulte vrijednosti u odnosu na zadatag ubrzanje, ali su pritom vrijednosti koje uzima realno ubrzanje veće ili jednake u odnosu na vrijednosti zadatag ubrzanja.

Slika 5.13 prikazuje apsolutne vrijednosti grešaka u praćenju zadatih trajektorija zglobova. Sa slike se može uočiti da se, u toku kretanja manipulatora, najmanja odstupanja od zadate trajektorije javljaju kod prvog (baznog) zgloba, dok se najveća odstupanja od zadate trajektorije javljaju kod trećeg, tj. zgloba lakta. Odnosno, najbolje praćenje referentne trajektorije se postiže za bazni zglob, dok se najgore performanse postižu za zglob lakta. Isti zaključak se dobija i posmatranjem srednje vrijednosti apsolutne greške (eng. *MAE – Mean Absolute Error*), za svaki pojedinačni zglob:

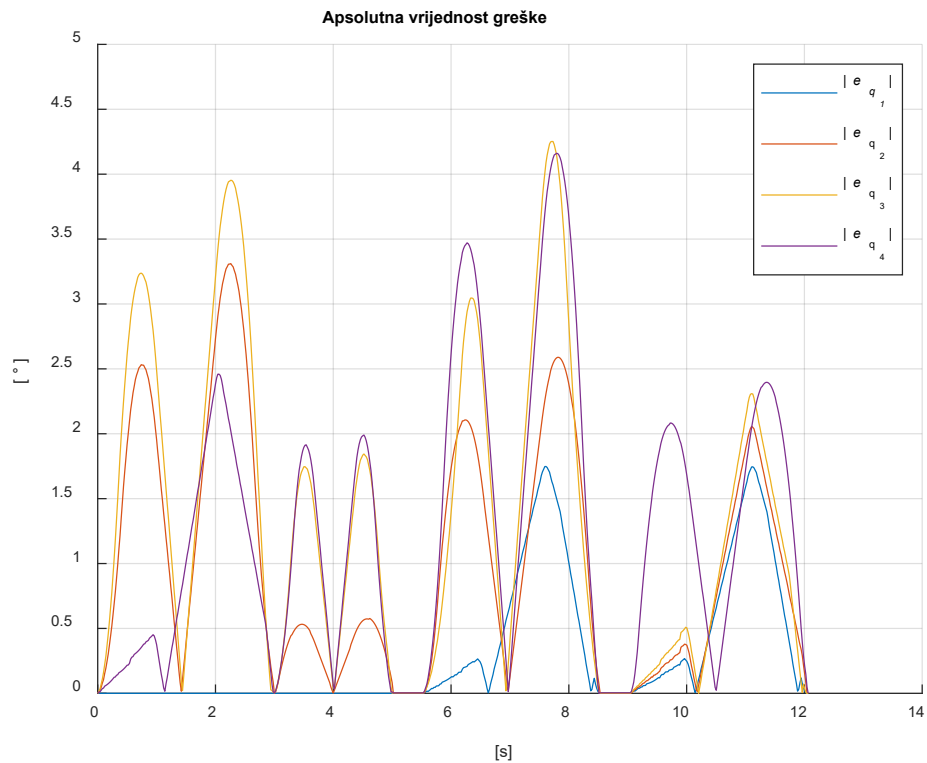
$$\mathbf{MAE} = [0.3201 \quad 1.0690 \quad 1.4504 \quad 1.3694]^T. \quad (5.5)$$

Ukoliko se uzme u obzir da je radni zadatak operacija prenošenja predmeta, kod koje je najbitnije postizanje zadatih položaja, može se zaključiti da nije neophodno redefinisane maksimalnih brzina zglobova u toku rada manipulatora.

Ukoliko se želi postići bolje praćenje zadatih trajektorija, u budućim istraživanjima se može uvesti povratna sprega po brzini ili poziciji servo motora. U ovom radu to nije bilo moguće izvesti iz razloga što servo motori koji se koriste nisu predviđeni za povratnu spregu sa mikrokontrolerom, pa se samim tim povratna informacija koja se dobija od njih ne može adekvatno iskoristiti za poboljšanje upravljanja (poglavlje 4.2.2).



Slika 5.12. Poređenje zadatih i realnih pozicija, brzina i ubrzanja zgloba laktava



Slika 5.13. Apsolutne vrijednosti grešaka u praćenju zadatih trajektorija zglobova

Jedna od mana realizacije SPT – 3000 manipulatora je što MATLAB programski paket nije softver otvorenog koda, čime se ograničava realizacija sličnih manipulatora na osnovu ovog rada. Takođe, komunikacija na relaciji MATLAB – Arduino je asinhrona serijska komunikacija, koja je jednostavna ali i se ujedno i pokazala kao ograničenje prilikom eksperimentalnog dijela rada. Kako je jedan od ciljeva ovog rada bio da se izradi niskobudžetni, reprogramabilni sistem otvorenog koda, sličan industrijskim manipulatorima, u daljim istraživanjima neophodno je posvetiti posebnu pažnju softveru. Iako MATLAB posjeduje Robotic Toolbox, koji je odličan za modelovanje manipulatora i planiranje trajektorije, isti ne pokriva mnogobrojne aspekte robotike, kao što je dinamička analiza, senzorika i algoritmi upravljanja. Iz navedenih razloga, fokus budućih istraživanja biće na implementaciji ROS (*Robot Operating System*) platforme [26]. ROS zapravo nije operativni sistem, već skup softverskih paketa i alata otvorenog koda za razvoj softvera u robotici, apstrakciju hardvera, kontrolu uređaja i komunikaciju između procesa. ROS je, od njegovih početaka 2007. godine u jednoj od laboratorija na Stenfordu, postao *defacto* standard za programiranje i razvoj u robotici, koji se koristi širom svijeta u nauci i industriji. Dakle, s obzirom da SPT – 3000 manipulator ima značajan prostor za nadogradnju, budući rad biće posvećen implementaciji ROS platforme, čime će se otvoriti mogućnosti za nova istraživanja na polju industrijske robotike.

Što se tiče mehaničkog sistema i planiranja kretanja, SPT – 3000 manipulator predstavlja bazični robotski sistem, što ujedno znači da postoji dosta prostora za njegovu nadogradnju. Nadogradnja se može vršiti na svim nivoima: zamjena/poboljšanje aktuatora, povećanje broja stepeni slobode [25], zadavanje radnog zadatka sa viših nivoa upravljanja – primjena vještačke inteligencije, primjena senzorike u radnom prostoru – korišćenje mašinske vizije za prepoznavanje objekata [24] i prepreka i slično. U radu su korišćeni servo motori sa integrisanom povratnom spregom, čime se ograničava modifikacija algoritma upravljanja i kontrola praćenja zadate putanje. Umajući to na umu, prilikom vršenja budućih nadogradnji sistema, prvenstveno je potrebno posveti pažnju odabiru ili izradi servo sistema sa povratnom informacijom, čime bi se značajno proširile mogućnosti sistema u pogledu modifikacije algoritma upravljanja i praćenja referentne trajektorije.

6. Zaključak

U radu je razmatrana problematika dizajna, izrade, programiranja i upravljanja prototipom industrijskog robotskog manipulatora. SPT – 3000 manipulator predstavlja prototip jednostavnog industrijskog manipulatora sa četiri stepena slobode, koji se može koristiti za edukaciju, testiranje i primjenu novih algoritama i koncepta upravljanja u robotici. U radu su, sa osvrtom na SPT – 3000 kao konkretan praktični primjer, izložene tehnike kinematičkog modelovanja, rješavanja direktnog i inverznog kinematičkog problema, planiranja kretanja i upravljanja. Pored teorijske analize, izvršen je i jednostavni eksperiment transporta radnog predmeta sa jednog mjesta na drugo, što predstavlja značajnu prednost ovog rada, budući da se većina radova u literaturi zasniva isključivo na razvijanju novih teorijskih koncepta, bez razvijanja načina na koji bi se ti sami koncepti mogli primijeniti u praksi.

Literatura

- [1] Y. Liu, H. Xu, C. Geng and G. Chen, „A modular manipulator for industrial applications: Design and implement“, *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, Šangaj, Kina, pp. 331-335, 2017.
- [2] Emmanuel Francalanza, Alec Fenech, Paul Cutajar, „Generative design in the development of a robotic manipulator“, *Procedia CIRP*, Vol. 67, pp. 244-249, 2018.
- [3] G. -S. Huang and P. -H. Tseng, „Development and analysis of 5-DOF manipulator kinematics“, *2016 International Conference on System Science and Engineering (ICSSE)*, Puli, Tajvan, pp. 1-4, 2016.
- [4] Huang, H., Dong, E., Xu, M., Yang, J. and Low, K.H., „Mechanism design and kinematic analysis of a robotic manipulator driven by joints with two degrees of freedom (DOF)“, *Industrial Robot*, vol. 45 no. 1, pp. 34-43, 2018.
- [5] F. Liu, G. Gao, L. Shi and Y. Lv, „Kinematic analysis and simulation of a 3-DOF robotic manipulator“, *3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, Gazijabad, Indija, pp. 1-5, 2017.
- [6] Amin A. Mohammed and M. Sunar, „Kinematics Modeling of a 4-DOF Robotic Arm“, *2015 International Conference on Control, Automation and Robotics*, pp. 87-91, 2015.
- [7] Y. Luan, W. Xu, J. Li, D. Zhou, H. Wang and H. Ji, „Kinematics Modeling and Simulation of a 4-DOF Manipulator“, *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, Daljen, Kina, pp. 302-305, 2017.
- [8] I. Chavdarov, V. Nikolov, B. Naydenov and G. Boiadjev, „Design and Control of an Educational Redundant 3D Printed Robot“, *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Hrvatska, pp. 1-6, 2019.
- [9] Swart, Arthur & Hertzog, Pierre, „Promoting Continuous Professional Development Among Academics from A Vocational College by Using A Practical Workshop Based on Arduino Technology“, *Advances in Science Technology and Engineering Systems Journal*, Vol. 5. No. 2, pp. 452-459, 2020.
- [10] P. E. Hertzog and A. J. Swart, „Arduino – Enabling engineering students to obtain academic success in a design-based module“, *2016 IEEE Global Engineering Education Conference (EDUCON)*, Abu Dabi, Ujedinjeni Arapski Emirati, pp. 66-73, 2016.
- [11] K. Jahnvi and P. Sivraj, „Teaching and learning robotic arm model“, *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, Kerala, Indija, pp. 1570-1575, 2017.
- [12] Al Tahtawi, Adnan & Agni, Muhammad & Hendrawati, Trisiani, „Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics“, *Journal of Robotics and Control (JRC)*, Vol. 2, 2021.

- [13] Istiaque Zahur and Pronob Deb Nath, Helal-An-Nahiyan and Uddin, „Design and Construction of an Articulated Pick and Place Manipulator“, *International Conference on Mechanical, Industrial and Energy Engineering 2020*, Kulna, Bangladeš, 2020.
- [14] I. Q. Kalimullah, A. Desikan and V. Kalaichelvi, „Analysis of a proposed algorithm for point to point control of a 3 DOF robot manipulator“, *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, Nagoja, Japan, pp. 289-292, 2017.
- [15] Adrian-Vasile Duka, „Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm“, *Procedia Technology*, Vol. 12, pp. 20-27, 2014.
- [16] Carlos Lopez-Franco, Jesus Hernandez-Barragan, Alma Y. Alanis and Nancy Arana-Daniel, „A soft computing approach for inverse kinematics of robot manipulators“, *Engineering Applications of Artificial Intelligence*, Vol. 74, pp. 104-120, 2018.
- [17] S. V. Reyes and S. P. Gardini, „Inverse kinematics of Manipulator Robot using a PSO Metaheuristic with Adaptively Exploration“, *2019 IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, Lima, Peru, pp. 1-4, 2019.
- [18] Hamza Khan, Hyun Hee Kim, Saad Jamshed Abbasi and Min Cheol Lee, „Real-Time Inverse Kinematics Using Dual Particle Swarm Optimization DPSO of 6-DOF Robot for Nuclear Plant Dismantling“, *IFAC – PapersOnLine*, Vol. 53, Issue 2, pp. 9885-9890, 2020.
- [19] S. C. Suhaimin, N. L. Azmi, M. M. Rahman and H. M. Yusof, „Analysis of Point-to-Point Robotic Arm Control using PID controller“, *2019 7th International Conference on Mechatronics Engineering (ICOM)*, Putradžaja, Malezija, pp. 1-6, 2019.
- [20] V. V. Kapustin, Y. I. Maltsev and A. K. Movchan, „Controlling a Robotic Manipulator Using Optical Feedback“, *2019 International Siberian Conference on Control and Communications (SIBCON)*, Tomsk, Rusija, pp. 1-4, 2019.
- [21] Vlatko Doleček, Isak Karabegović, „Robotika“, Univerzitet u Bihaću, 2002.
- [22] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, „Robotics Modelling, Planning and Control“, *Springer Advanced Textbooks in Control and Signal Processing*, 2009.
- [23] Bruno Siciliano, Oussama Khatib, „Springer Handbook of Robotics“, *Springer*, 2008.
- [24] Peter Corke, „Robotics, Vision and Control“, *Springer Tracts in Advanced Robotics*, 2013.
- [25] X. Cheng and M. Zhao, „Analysis on the Trajectory Planning and Simulation of Six Degrees of Freedom Manipulator“, *2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Huhot, Kina, pp. 385-387, 2018.
- [26] Wyatt S. Newman, „A Systematic Approach to Learning Robot Programming with ROS“, 2018.
- [27] Daniel E. Kandray, „Programmable Automation Technologies“, *Industrial Press Inc. New York*, 2010.
- [28] Emmanuel Agbaraji, Ifeyinwa Obiora-Dimson, „Joint Torque and Motion Computational Analysis for Robotic Manipulator Arm Design“, *Journal of Engineering and Applied Sciences*, 2018.

- [29] Jorge Angeles, „Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms, Second Edition“, *Springer Mechanical Engineering Series*, Montreal, 2002.
- [30] J. Denavit, R.S. Hartenberg, „A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices“, *ASME Journal of Applied Mechanics*, 1955.
- [31] L. Wang, B. Ravani, „Recursive computations of kinematic and dynamic equations for mechanical manipulators“, *IEEE Journal on Robotics and Automation*, Vol. 1, No. 3, pp. 124-131, September 1985.
- [32] Kevin M. Lynch, Frank C. Park, „Modern Robotics – Mechanics, Planning and Control“, *Cambridge University Press*, 2017.
- [33] Andrew A. Goldenberg, B. Benhabib, Robert Fenton, „A Complete Generalized Solution to the Inverse Kinematics of Robots“, *IEEE Journal of Robotics and Automation*, Vol. 1, No. 1, pp. 14-20, 1985.
- [34] Pragathi Praveena, Daniel Rakita, Bilge Mutlu, Michael Gleicher, „User-Guided Offline Synthesis of Robot Arm Motion from 6-DoF Paths“, *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Kanada, 2019.
- [35] Kiam Heong Ang, G. Chong, Yun Li, „PID control system analysis, design, and technology“, *IEEE Transactions on Control Systems Technology*, Vol. 13, No. 4, pp. 559-576, Jul 2005.
- [36] Veljko Potkonjak, „Robotika“, Univerzitet u Beogradu, 1995.
- [37] X. Li, S. Tan, X. Feng, H. Rong, „LSPB Trajectory Planning: Design for the Modular Robot Arm Applications“, *2009 International Conference on Information Engineering and Computer Science*, Vuhan, Kina, 2009.

Prilog

1. MATLAB kod za generisanje referentnih trajektorija

```
%% kinematički model SPT - 3000
n = 4; % broj stepeni slobode

% D - H parametri SPT - 3000 manipulatora; duzine su date u cm
DH_SPT3000 = [0 9.94 -1.4 pi/2; 0 0 13.4 0; 0 0 13.25 0; 0 2.208 11.85 -pi/2];

for i=1:n
    L(i) = Link(DH_SPT3000(i,:));
end

SPT_3000 = SerialLink(L, 'name', 'SPT - 3000');

%% generisanje trajektorija za radni zadatak

% 1) generisanje tacaka trajektorije

qp = [0 pi/2 -pi/2 0]; % pocetni polozaj - un. koordinate
Tp = SPT_3000.fkine(qp); % pocetni polozaj - spolj. koordinate

Tobj = [0 0 1 20; 0 1 0 -2.208; -1 0 0 3; 0 0 0 1]; % pozicija objekta
qobj = SPT_3000.ikunc(Tobj, [0 1 -1 -1]); % polozaj robota koji odgovara objektu
% SPT_3000.plot(qobj)

% sljedeci korak - medjupolozaj izmedju pocetnog i krajnjeg!!!
% medjupolozaj ce biti na visini z = 10

Tm = [0 0 1 20; 0 1 0 -2.208; -1 0 0 10; 0 0 0 1]; % medjupolozaj
qm = SPT_3000.ikunc(Tm, [0 1 -1 -1]); % untrasnje koordinate koje odgovaraju
medjupolozaju

% trajektorija je tipa lspb - kombinacija linearnih polinoma sa parabolama

% 1) robot krece iz pocetnog polozaja do medjupolozaja
% vrijeme kretanja je 3 sekunde, korak je 20ms (0.02s)
dt = 0.02;

t1 = 0:dt:3;
q1 = mtraj(@lspb, qp, qm, t1);

% 2) robot se krece od medjupolozaja do predmeta
% vrijeme kretanja je 2 sekunde, korak je 20ms (0.02s)

t2 = 3:dt:5;
q2 = mtraj(@lspb, qm, qobj, t2);
```

```

% 3) robot se kreće od položaja Tobj do krajnjeg položaja Tk
Tk = troz(45); Tk(1:3, end) = [25 25 20];
qk = SPT_3000.ikunc(Tk, [1 1 -1 0]);

t3 = 5.5:dt:8.5;
q3 = mtraj(@lspb, qobj, qk, t3);

% 4) povratak u početni položaj
t4 = 9:dt:12;
q4 = mtraj(@lspb, qk, qp, t4);

SPT_3000.plot(qk)

q = [q1; q2(2:end, :); q2(end,:).*ones(0.5/dt-1, n); q3; q3(end,:).*ones(0.5/dt-1, 4);q4];
qv = diff(q)/dt;
qv = [qv; zeros(1, 4)];
qa = diff(qv)/dt;
qa = [qa; zeros(1, 4)];

t = 0:dt:12;

tz = [0 3 5 5.5 8.5 9 12]; % vremena u kojima se prolazi kroz zadate tacke
qz = [qp' qm' qobj' qobj' qk' qk' qp']; % zadate tacke trajektorije

```

2. MATLAB kod za komunikaciju sa Arduino mikrokontrolerom

```

clc
close all
clear arduino s q_ard

flag = 0;
% serialport objekat je potrebno zasebno inicijalizovati

arduino = serialport("COM4", 115200);
pause;

write(arduino, 'r', 'char');
pause(1)
s = read(arduino, 1, 'char');

if (s == 'r')
    disp('Komunikacija je uspostavljena.');
```

```

% slanje, prijem i obrada podataka

% uglovi i vremena pojedinačnih se šalju u formi stringova,
% a zatim se na Arduinou pretvaraju u brojne vrijednosti
q_zad = [qm qobj qk];
q_deg = q_zad * 180 / pi;
q_deg_t = [q_deg [3 2 3 3]];
q_deg_t_str = string(round(q_deg_t, 2));

for i = 1:length(q_deg_t_str)
    writeline(arduino, q_deg_t_str(i));
end

temp = "";

while flag ~= 1
    if arduino.NumBytesAvailable > 0
        temp = readline(arduino);
        disp(temp)
    end
    temp = strip(temp);
    if strcmp(temp, "Prijem podataka završen.")
        flag = 1;
    end
end

flag = 0;
i = 1;
temp = "";

while flag ~= 1
    if arduino.NumBytesAvailable > 0
        temp = readline(arduino);
        disp(temp)
    end
    temp = strip(temp);
    if strcmp(temp, "Kretanje je završeno.")
        flag = 1;
    else
        q_ard(i,:) = temp;
        i = i + 1;
    end
end

% dobijeni podaci se mogu obraditi f-jom str2num ili str2double

q_ard = unique(q_ard, 'stable');
q_ard = split(q_ard(2:end)); % razdvajanje brojnih vrijednosti

```

```
% prva kolona predstavlja vremenski zakon Arduino trajektorije
% svaka naredna kolona predstavlja trajektoriju jednog od zglobova
```

```
t_ard = zeros(1, length(q_ard));
q_ard_z1 = zeros(1, length(q_ard));
q_ard_z2 = zeros(1, length(q_ard));
q_ard_z3 = zeros(1, length(q_ard));
q_ard_z4 = zeros(1, length(q_ard));
qv_ard_z1 = zeros(1, length(q_ard));
qv_ard_z2 = zeros(1, length(q_ard));
qv_ard_z3 = zeros(1, length(q_ard));
qv_ard_z4 = zeros(1, length(q_ard));

for i=1:length(q_ard)
    t_ard(i) = str2double(q_ard(i, 1));
    q_ard_z1(i) = str2double(q_ard(i, 2));
    qv_ard_z1(i) = str2double(q_ard(i, 3));
    q_ard_z2(i) = str2double(q_ard(i, 4));
    qv_ard_z2(i) = str2double(q_ard(i, 5));
    q_ard_z3(i) = str2double(q_ard(i, 6));
    qv_ard_z3(i) = str2double(q_ard(i, 7));
    q_ard_z4(i) = str2double(q_ard(i, 8));
    qv_ard_z4(i) = str2double(q_ard(i, 9));
end

% kraj = (t_ard(end) - t_ard(1))/1000;
t_ard = (t_ard - t_ard(1))/1000;
```

```
qa_ard_z1 = diff(qv_ard_z1)./diff(t_ard);
qa_ard_z1 = medfilt1([qa_ard_z1 0], 3);
qa_ard_z2 = diff(qv_ard_z2)./diff(t_ard);
qa_ard_z2 = medfilt1([qa_ard_z2 0], 3);
qa_ard_z3 = diff(qv_ard_z3)./diff(t_ard);
qa_ard_z3 = medfilt1([qa_ard_z3 0], 3);
qa_ard_z4 = diff(qv_ard_z4)./diff(t_ard);
qa_ard_z4 = medfilt1([qa_ard_z4 0], 3);
```

3. Arduino kod za upravljanje SPT – 3000 manipulatorom i komunikaciju sa MATLAB-om

```
#include <Servo.h>
#include "trajectory.h"

// definisanje Servo objekata
Servo servo1; // servo - baza robota
Servo servo2; // servo - rame
Servo servo3; // servo - lakat
Servo servo4; // servo - zglob sake
Servo servo5; // servo - hvataljka
```

```

// definisanje trajectory objekata za servo motore
Trajectory servo_traj1(23.7113, 23.7113);
Trajectory servo_traj2(34.0049, 34.0049);
Trajectory servo_traj3(45.0001, 45.0001);
Trajectory servo_traj4(47.8687, 47.8687);

#define UPDATE_FREQUENCY 50 // frekvencija azuriranja trajektorije
#define UPDATE_TIME (1000 / UPDATE_FREQUENCY)
unsigned long updateTimer = 0;
int moveNumber = 0;
float zad_koord[12];
float poc_koord[5] = {0, 90, -90, 0, 90};
float vremena_traj[4];
char prijem = 'n';

// definisanje redosljeda pokreta
void nextMove() {
    switch (moveNumber) {
        case 0:
            servo_traj1.setTargetPos(zad_koord[0], vremena_traj[0]);
            servo_traj2.setTargetPos(zad_koord[1], vremena_traj[0]);
            servo_traj3.setTargetPos(zad_koord[2], vremena_traj[0]);
            servo_traj4.setTargetPos(zad_koord[3], vremena_traj[0]);
            break;

        case 1:
            servo_traj1.setTargetPos(zad_koord[4], vremena_traj[1]);
            servo_traj2.setTargetPos(zad_koord[5], vremena_traj[1]);
            servo_traj3.setTargetPos(zad_koord[6], vremena_traj[1]);
            servo_traj4.setTargetPos(zad_koord[7], vremena_traj[1]);
            break;

        // uzimanje radnog predmeta
        case 2:
            sendData();
            servo5.write(20);
            delay(500);
            sendData();
            break;

        case 3:
            servo_traj1.setTargetPos(zad_koord[8], vremena_traj[2]);
            servo_traj2.setTargetPos(zad_koord[9], vremena_traj[2]);
            servo_traj3.setTargetPos(zad_koord[10], vremena_traj[2]);
            servo_traj4.setTargetPos(zad_koord[11], vremena_traj[2]);
            break;

        // otpustanje radnog predmeta
        case 4:
            sendData();
            servo5.write(poc_koord[4]);
            delay(500);
            sendData();
            break;
    }
}

```

```

    case 5:
        servo_traj1.setTargetPos(poc_koord[0], vremena_traj[3]);
        servo_traj2.setTargetPos(poc_koord[1], vremena_traj[3]);
        servo_traj3.setTargetPos(poc_koord[2], vremena_traj[3]);
        servo_traj4.setTargetPos(poc_koord[3], vremena_traj[3]);
        break;

    default:
        // ako su svi pokreti završeni, zaustavi program
        Serial.println("Kretanje je završeno.");
        while(1) {}
    }

    moveNumber++;
}

// prijem podataka od MATLAB-a
void prijem_podataka() {
    int i = 0;
    int j = 0;
    int ind = 0;
    String temp;
    while (ind == 0)
    {
        if (Serial.available() > 0)
        {
            temp = Serial.readStringUntil('\n');
            if (i < 12){
                zad_koord[i] = temp.toFloat();
                i++;
            }
            else if (i == 12 && j < 4){
                vremena_traj[j] = temp.toFloat();
                j++;
            }
        }
        if (i == 12 && j == 4)
        {
            ind = 1;
            for (int k = 0; k < i; k++){
                Serial.println(zad_koord[k]);
            }
            for (int k = 0; k < j; k++){
                Serial.println(vremena_traj[k]);
            }
            Serial.println("Prijem podataka završen.");
        }
    }
}

// skaliranje ugaona pozicija zgloba/sirina impulsa
float mapFloat(float x, float in_min, float in_max, float out_min, float
out_max){
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

```

```

// slanje podataka
void sendData() {
    Serial.print(millis());
    Serial.print(" ");
    Serial.print(servo_traj1.getPos());
    Serial.print(" ");
    Serial.print(servo_traj1.getVel());
    Serial.print(" ");
    Serial.print(servo_traj2.getPos());
    Serial.print(" ");
    Serial.print(servo_traj2.getVel());
    Serial.print(" ");
    Serial.print(servo_traj3.getPos());
    Serial.print(" ");
    Serial.print(servo_traj3.getVel());
    Serial.print(" ");
    Serial.print(servo_traj4.getPos());
    Serial.print(" ");
    Serial.println(servo_traj4.getVel());
}

void setup() {
    // uspostavljanje komunikacije
    Serial.begin(115200);

    // provjera da li je komunikacija uspostavljena
    while(prijem == '\n')
    {
        if (Serial.available() > 0)
        {
            prijem = Serial.read();
            if (prijem == 'r');
                Serial.print(prijem);
        }
    }

    // prijem podataka
    prijem_podataka();

    // podesavanje servo motora
    servo1.attach(3);
    servo2.attach(4);
    servo3.attach(5);
    servo4.attach(6);
    servo5.attach(7);

    servo1.writeMicroseconds(map(poc_koord[0], -135, 135, 500, 2500));
    servo2.writeMicroseconds(map(poc_koord[1], -135, 135, 500, 2500));
    servo3.writeMicroseconds(map(poc_koord[2], -135, 135, 500, 2500));
    servo4.writeMicroseconds(map(poc_koord[3], -135, 135, 500, 2500));
    servo5.write(poc_koord[4]);
    delay(1500);
}

```

```

// postavljanje trajektorija na pocetne vrijednosti
servo_traj1.reset(poc_koord[0]);
servo_traj2.reset(poc_koord[1]);
servo_traj3.reset(poc_koord[2]);
servo_traj4.reset(poc_koord[3]);

}

void loop() {

// azuriranje pozicija servo motora se vrši po definisanom intervalu
if (millis() - updateTimer >= UPDATE_TIME) {
    updateTimer = millis();

    // azuriranje trajektorija zglobova
    float servo_angle1 = servo_traj1.update();
    float servo_angle2 = servo_traj2.update();
    float servo_angle3 = servo_traj3.update();
    float servo_angle4 = servo_traj4.update();
    float angle1_skal = mapFloat(servo_angle1, -135.0, 135.0, 500.0,
2500.0);
    float angle2_skal = mapFloat(servo_angle2, -135.0, 135.0, 530.0,
2530.0);
    float angle3_skal = mapFloat(servo_angle3, -135.0, 135.0, 475.0,
2475.0);
    float angle4_skal = mapFloat(servo_angle4, -135.0, 135.0, 520.0,
2520.0);

    // slanje azuriranih pozicija servo motoru; f-ja kao argument uzima
cijele brojeve
    servo1.writeMicroseconds(round(angle1_skal));
    servo2.writeMicroseconds(round(angle2_skal));
    servo3.writeMicroseconds(round(angle3_skal));
    servo4.writeMicroseconds(round(angle4_skal));

    // slanje podataka MATLAB-u
    sendData();

    // kada svi servo motori dostignu zadatu poziciju, prelazi se na
sljedeci potez
    if (servo_traj1.ready() && servo_traj2.ready() && servo_traj3.ready()
&& servo_traj4.ready()) {
        nextMove();
    }
}
}

```