

UNIVERZITET U PRIŠTINI
FAKULTET TEHNIČKIH NAUKA

**Primena teorije zavrtnjeva i njena
implementacija u Pajtonu za upravljanje Niryo
One manipulatorom**

Seminarski rad iz Primene programskog jezika Python (Pajton) za upravljanje
industrijskim robotima

Radio: Vuk Todorović
Broj indeksa: 158/20
Školska godina: 2023/2024.

Predmetni profesor: dr Marko Pantić
Predmetni asistent: dr Marko Pantić

Sadržaj

Uvod	1
1 Robotski Operativni Sistem-ROS	2
1.1 Princip rada ROS-a	5
2 Niryo One manipulator	8
2.1 Osnovna razmatranja	8
2.2 Tehničke specifikacije	9
3 Mehanika robota	12
3.1 Teorijska razmatranja	13
3.1.1 Konfiguracioni prostor	13
3.1.2 Kretanje krutog tela	14
3.1.3 Direktna kinematika	19
3.1.4 Kinematika brzine i statika	20
3.1.5 Inverzna kinematika	21
3.2 Mehaničke karakteristike Niryo One manipulatora	25
4 Eksperiment	30
Zaključak	35
Literatura	36

Uvod

Robotika u modernom dobu dobija sve veću ulogu u oblikovanju modernog života te su pravac inovacija sve više usmerene prema ovoj interdisciplinarnoj naučnoj oblasti. Topologija robota je složena i kako pojedinačni tako i povezani elementi bitno mogu uticati na celokupnu strukturu i funkciju ovog mehanizma. Ovo ide u prilog tome da je sve teže izučavati jedan aspekt strukture robota nezavisno od ostatka ali je zato lako moguće uvideti kako ta zavisna jedinica može uticati na ostatak sistema i izvršiti analizu njegovog uticaja radi utvrđivanja

Suština rada jeste izlaganje osnove **teorije zavrtnjeva** (engl. *screw theory*) i korišćenja za analizu i sintezu mehanike robota i njena primena u upravljanju Niryo One manipulatorom putem programskog jezika **Pajton** (engl. *Python*) i njegovih, javno dostupnih, biblioteka.

Rad počinje u prvom poglavlju opštim izlaganjem na temi **ROS-a** (engl. *Robot Operating System*) kako bi se ukazao na oblik i karakteristike celokupnog softverskog upravljačkog sistema koji će biti korišćen za slanje i primanje podataka od robota. ROS neće dobiti ni približno detaljan opis, jer to nije poenta rada, već samo letimičan pregled osnove.

U drugom poglavlju se navode osnovni pojmovi **robotских manipulatorа** i prilažu se najbitnije specifikacije **Niryo One** robota. Pored toga, prikazuje se struktura robota, pozicije značajnih elemenata, vrsta komunikacijske veze i protokola i navodi se pyniryo Pajton biblioteka koja nam omogućava da sa visokog nivoa apstrakcije primenjujemo ROS za upravljanje robotom.

U trećem poglavlju dolazimo do srži rada. Nakon kratkog osvrta na osnovne pojmove klasične mehanike i mehanike robota, okrećemo se objašnjenju opisu prostora delovanja robota–**konfiguracioni prostor**, njegovoj dimenziji (**stepeni slobode**) i obliku (**topologiji**) i način reprezentacije na kojoj se upravo i zasniva teorija zavrtnjeva o kojoj će biti reč do skoro do kraja poglavlja. Na samom kraju biće primenjena teorija zavrtnjeva i **Paden-Kahanovi podproblemi** za rešavanje **inverzne kinematike** robota (određivanje aktuacije koja odgovara zadatoj specifikaciji robota).

Eksperiment koji prikazuje rad robota i primene dotadašnje teorije je prikazano u četvrtom poglavlju. Izvorni kôd je baziran na dostupnoj Pajton biblioteci od Linča i Parka [8], a dostupan je na veb adresi [5].

Glava 1

Robotski Operativni Sistem-ROS

Definicija robota po **ISO** standardu glasi „Industrijski robot je automatski, servoupravljeni, reprogramabilni, višefunkcionalni manipulator koji ima više stepeni slobode kretanja, koji je pogodan za rukovanje materijalom, delovima, alatima ili posebnim uređajima za izvršavanje varijabilnih programiranih operacija. Industrijski roboti obično imaju jednu ili više ruku, zatim, upravljački uređaj s mogućnošću pamćenja, a ponekad koriste senzore za prikupljanje informacija iz okoline. Oni su višenamenski uređaji koji vrše ponavljajuće operacije i za izvršavanje tih operacija imaju mogućnost prilagođavanja okolini.“[3].

Već po samoj definiciji je evidentno da se mogu ubrojiti mnogobrojne i raznorodne vrste mašina (od mašine za pranje veša pa sve do SCARA industrijskih manipulatora) i s tim u vezi postavlja se pitanje, kako koordinisati rad više heterogenih robota? Postoji manjak standarda koji mogu dati zadovoljavajuće rešenje ovom problemu. Jedno prosto rešenje jeste da se putem odgovarajućeg interfejsa programira rad svakog robota, a zatim da se konstruiše određen protokol komunikacije među uređajima. Ovakav pristup je prilično nepogodan iz više razloga:

- Manjak pravila izvođenja dovodi do različitih izvedbi na različitim projektima-nedovoljna standardizacija;
- Potreba za ponovnim programiranjem standardnih algoritama i protokola na različitim uređajima iste vrste;
- Integracija novog robota u već oformljen sistem zahteva enorman rad.

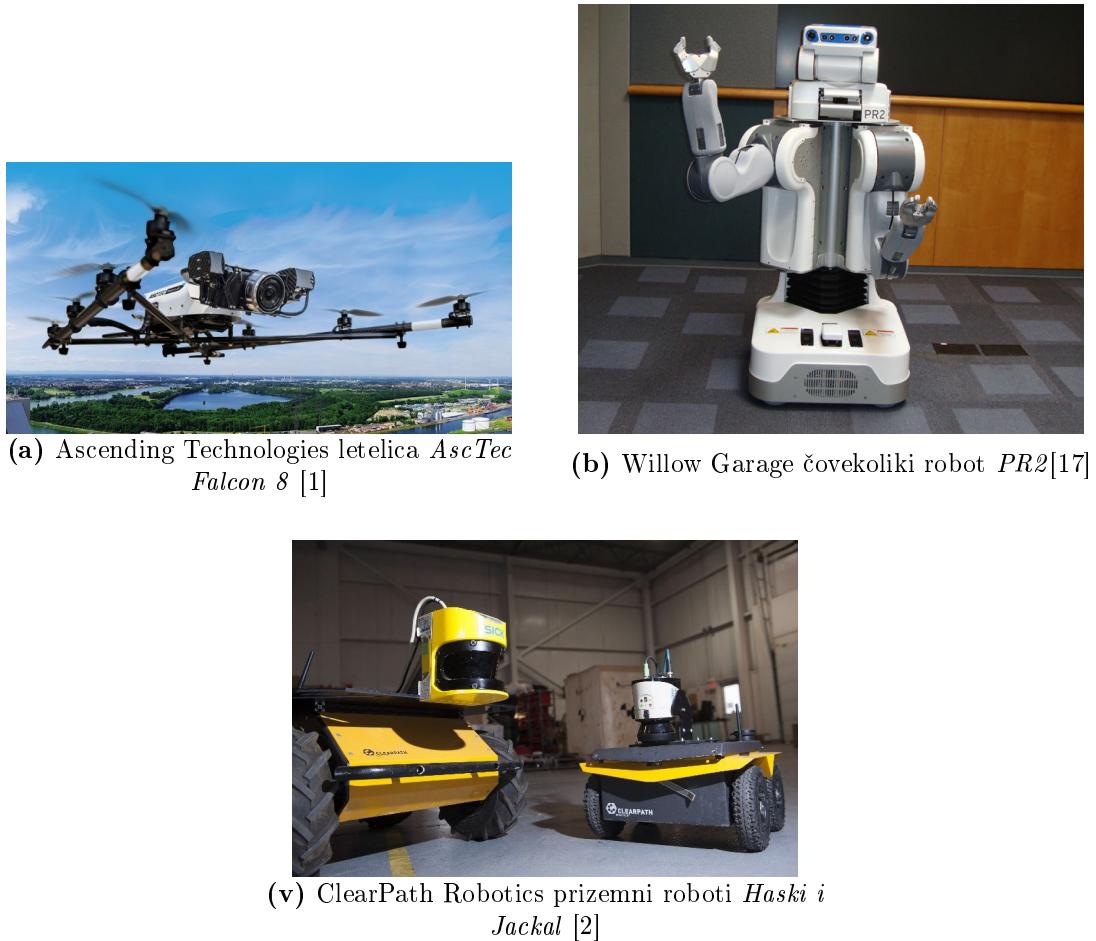
Novo i savremeno rešenje ovom problemu nas dovodi do glavne teme ovog poglavlja.

Jedan od standardnih rešenja koje se masovno koristi u industriji i naučnim istraživanje jeste Robotski Operativni Sistem – **ROS** (engl. *Robot Operating System*) je **meta-operativni sistem** koji je sačinjen od **softvera otvorenog koda** (engl. *open-source*)[15]. Meta-operativni sistem je sličan operativnom sistemu u klasičnom značenju pojma u šta spada: nivo apstrakcije pri korišćenju dostupnog hardvera, mogućnost korišćenja uređaja na niskom nivou, implementacija-ačesto korišćenih funkcionalnosti, komunikacija među procesima koje se izvršavaju, korišćenje gotovih paketa i njihovim upravljanjem... Međutim, striktno govoreći, ROS je skup softverskih frejmворкова (engl. *framework*) koji upravlja operativnim sistemima različitih robota. Kao takav, on nudi alate i biblioteke za prikupljanje, pisanje, kompajliranje i korišćenje programskog koda nad uređajima. Njegove glavne karakteristike su:

- **Skaliranje**–mogućnost lakog uvećanja ROS upravljanog sistema;

- **Nezavisnost jezika**—pisanje programskog koda je moguće na bilo kom jeziku za koji postoji odgovarajuća biblioteka. Postoje mnogobrojne biblioteke, a među njima spadaju biblioteke za *Pajton*, (engl. *Python*), *C++*, *MATLAB*, *Java*, *LISP* i mnogi drugi;
- **Testiranje**—ROS ima ugrađen frejmворк за testiranje koji se zove *rostest*;
- **P2P mreža**—svaki robot može započeti komunikaciju sa svakim drugim (engl. *peer-to-peer communication*);
- **Akcenat na alate**—različite alate omogućavaju povećanu pogodnost u čuvanju podataka, crtajući grafika, vizuelizacija, itd;
- **Besplatan, softver otvorenog koda, velika zajednica korisnika i programera koji razvijaju ROS, mnogobrojne gotove biblioteke.**

Mada ROS nije savršen sistem koji može ispuniti očekivanja današnjeg industrijskog i istraživačkog sektora koji se eksponencijalno razvijaju, ipak poseduje povoljne karakteristike za primenu u nebrojivo mnogo slučajeva.

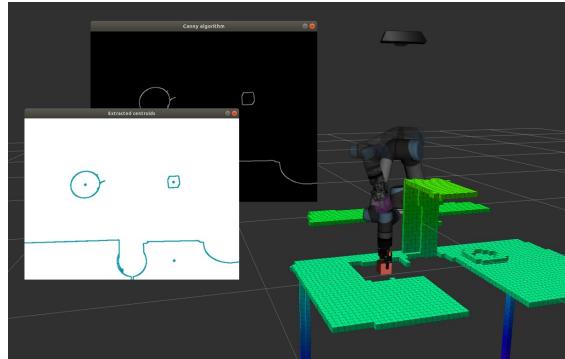


Slika 1.1: Primeri robotskih sistema koji koriste ROS

Pored ovih kompleksnih sistema, ROS je moguće implementirati i sa prostijim komponentama kao što su senzori. Među senzorima kojima ROS može spadaju GPS resiveri, kamere, laseri i za njih poseduje brojne drajvere koje su dostupne u okviru samog ROS sistema. U

slučaju da nisu dostupni drajveri za neki robot, s obzirom da je ROS softver otvorenog koda, poseduje određene beneficije.

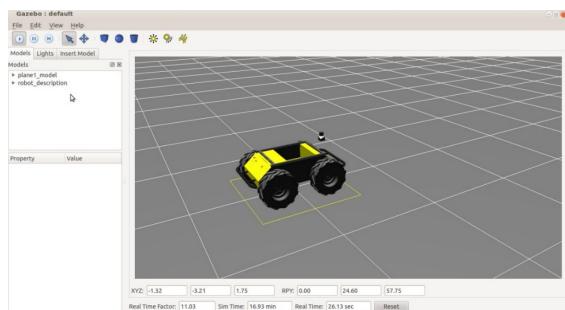
Programeri koji rade u ROS-u mogu podeliti svoj kôd za upotrebu ostalim korisnicima i zarad saradnje u usavršavanju već dostupnog kôda. S tim u vezi, logično je zašto se ROS uglavnom koristi kod operativnih koji su bazirani na *Unix*. ROS se najčešće testira na *Ubuntu* i *Mac OS X* sistemima dok je ROS zajednica takođe uticala na napredak podrške na sistemima kao što su *Fedora*, *Gentoo*, *Arch Linux* i druge *Linux* platforme. Mada je moguće koristiti ROS na *Microsoft Windows*-u, nije u potpunosti testirano pa bi trebalo izbegavati ovu opciju. Još jedna beneficija koju bi trebalo napomenuti jesu integracija ROS-a sa drugim softverskim bibliotekama otvorenog koda kao što su *OpenCV*–slika1.2a, *PCL* (eng. ***Point Cloud Library***)–slika1.2b, *Gazebo simulator*–slika1.2v i drugi.



(a) Integracija ROS-a sa *OpenCV* bibliotekom
[14]



(b) Vizuelizacije okoline pomoću ROS-a i
PCL-a [16]



(v) Gazebo simulator robota upravljanog
pomoću ROS-a [16]

Slika 1.2: ROS u saradnji sa softverima otvorenog koda

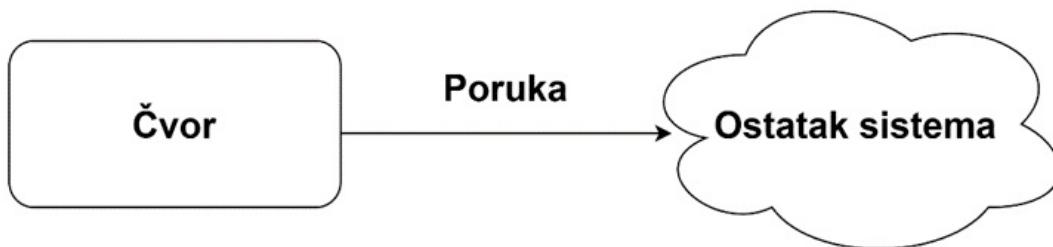
Pošto smo definisali šta je ROS, njegove karakteristike i način upotrebe, sledi objasniti kako je koncipiran njegov rad.

1.1 Princip rada ROS-a

Rad ROS-a se može opisati pomoću grafa koji sačinjava P2P mrežu određenih procesa (koji mogu biti raspodeljeni na različite uređaje) koji funkcioniše pomoću ROS-ove komunikacione infrastrukture. Ovo znači da komponentne mogu istovremeno obrađivati podatke na različitim uređajima i komunicirati međusobne te podatke na različite načine. Izvedba ROS-a je usko vezana za njegove ključne delove (način njihovog korišćenja zavisi od konkretnog problema koji se rešava), a među kojima spadaju čvorovi, poruke, teme, servisi, master, server parametara i torbe.

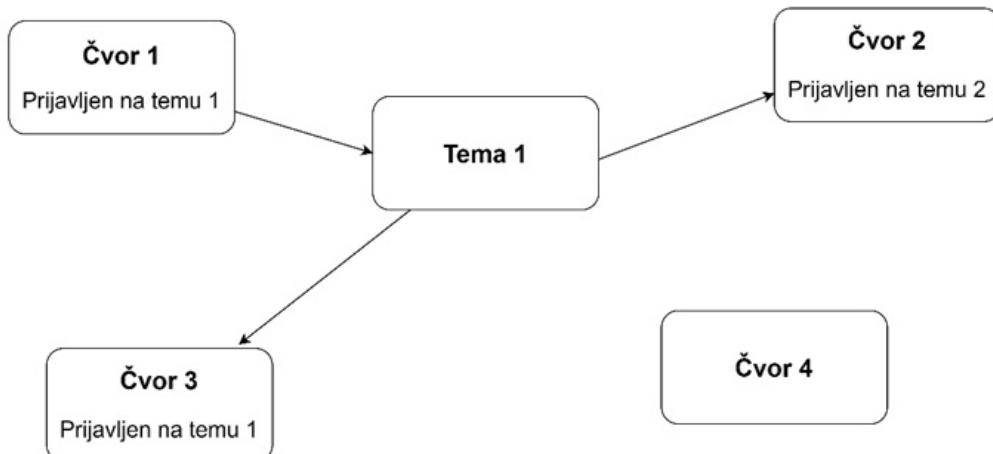
Čvorovi (engl. *Nodes*) jesu softverski procesi koji obavljaju određenu operaciju. Sistem je tipično sastavljen od mnoštvo čvorova gde svaki obavlja određenu operaciju i dozvoljava dobro skaliranje na veće sisteme. Zovu se čvorovi jer mogu se apstraktno opisati kao čvorovi grafa sistema. Primeri čvorova su drajveri senzora, drajveri aktuatora, korisnički interfejs, itd.

Poruke (engl. *Messages*) predstavljaju način komunikacije čvorova sa ostatkom sistema. Poseduju osnovne informacije o tipovima podataka (moraju biti predefinisani tipovi i to isključivo standardni koje ROS prepozna) i njihove vrednosti (slika 1.3).



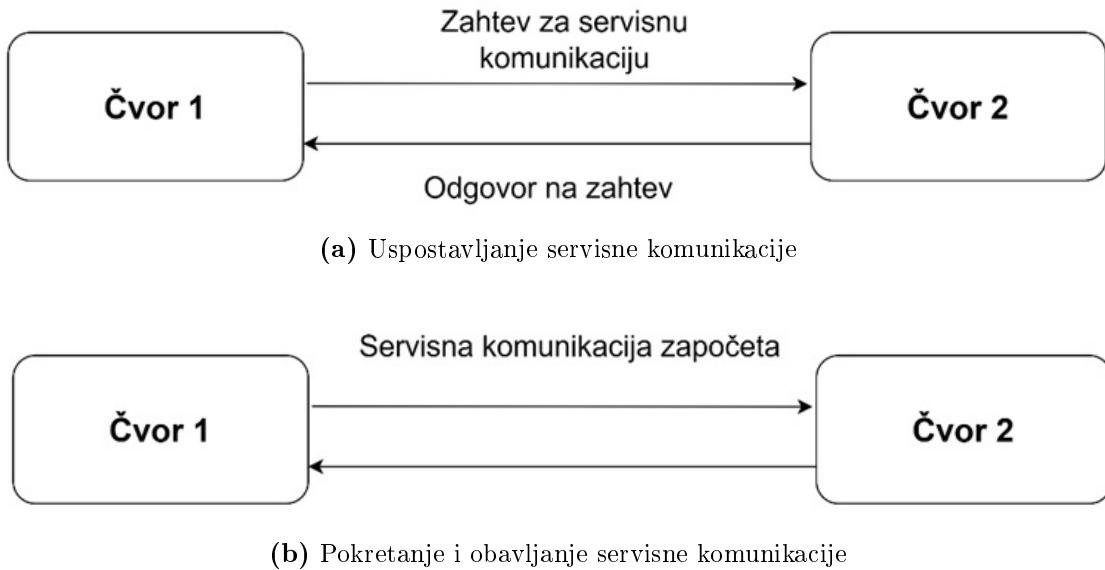
Slika 1.3: Slanje poruka od strane čvora ka ostatku sistema

Teme (engl. *Topics*) jesu način klasifikacije poruke rade lakšeg učitavanja i slanja prema više čvorova. Čvor kako bi primio određenu poruku od teme on mora biti prijavljen na određenu temu (engl. *subscribed*) i onda može primiti sve poruke koje se razvrstavaju u određenu temu. Takođe može slati poruke na određenu temu. Čvorovi nisu svesni postojanja drugih čvorova preko tema, samo su svesni same teme i poruke koje nose sa sobom (slika 1.4).



Slika 1.4: Komunikacija čvorova pomoću date teme

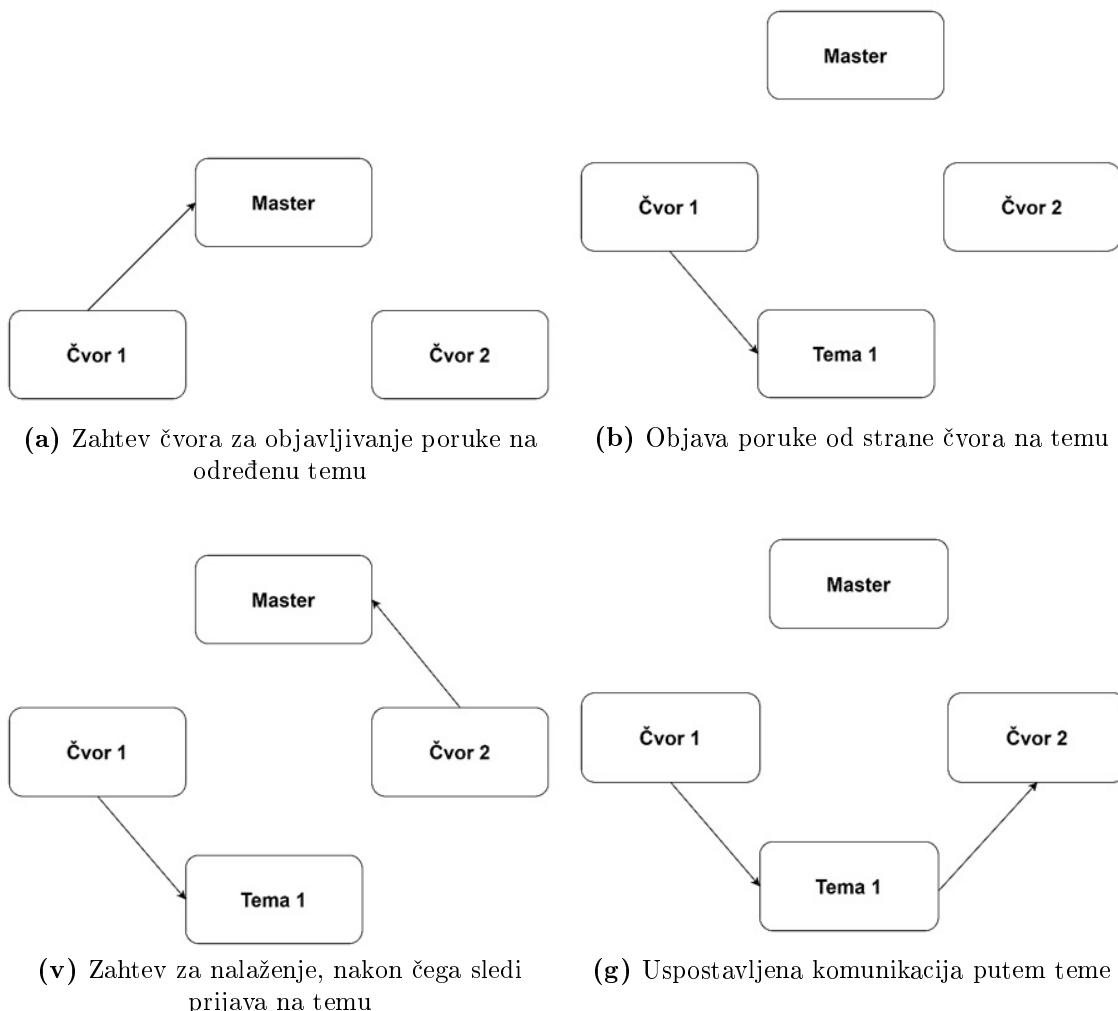
Servisi (engl. *Service*) omogućavaju direktnu sinhronizovanu komunikaciju među čvorovima. Za razliku od asinhronizovane komunikacije putem tema, komunikacija putem servisa je znatno brže i bolje u slučajevima kada je potreba konstantna razmena informacije između dva čvora. Dva čvora se uparuju za servisnu komunikaciju tako što jedan čvor šalje zahtev drugom za pristup takvoj vrsti komunikacije, drugi obradi zahtev i pošalje nazad informaciju da komunikacija može započeti (slika 1.5).



Slika 1.5: Servisna komunikacija

Master (engl. *Master*) sadrži informacije za uspostavljanje veze čvora sa ostatkom sistema. Npr. kako bi dva čvora započeli komunikaciju putem neke teme prvo mora tražiti informacije od Mastera kako bi objavilo nešto na toj temi (slika 6.a) nakon čega Master dozvoljava i nastaje tema sa porukama (slika 6.b), zatim drugi čvor traži informacije o toj temi kako bi se prijavilo (slika 6.v) i komunikacija tek onda teče nakon prijave (slika 6.g).

Torbe (engl. *Bags*) je format u kome se čuvaju i čitaju ROS poruke. Tipično se koristi za razvoj i testiranje algoritama sistema.



Slika 1.6: Primer uloge Mastera u komunikaciji

Glava 2

Niryo One manipulator

2.1 Osnovna razmatranja

Niryo Robotics je startup iz Francuske koji dizajniraju i proizvode industrijske robote, tačnije ti roboti su namenjeni za lične i edukativne svrhe te je cena znatno niža od prosečnog industrijskog robota čija se cena kreće između 20,000\$ i 200,000\$.[6] Model robota koji je primenjen u ovom radu je upravo od Niryo Robotics–**Niryo One** (slika 2.1).

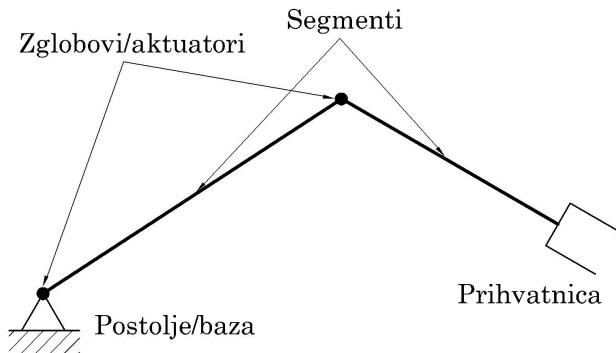


Slika 2.1: Izgled Niryo One robota[6]

Niryo One spada u manipulatore—“mašina u kojoj se mehanizam obično sastoji od niza segmenata, spojenih ili klizeći jedan u odnosu na drugi, u svrhu hvatanja i/ili pomeranja predmeta (komada ili alata) obično u nekoliko stepeni slobode”[4].

Sastav manipulatora (slika 2.2):

- **Segmenti**—Kruta ili gipka tela koja drže ostale elemente (u našem slučaju razmatraćemo isključivo kruta);
- **Zglobovi/aktuatori**—Vezivni elementi koji dozvoljavaju međusobno kretanje, tipično su aktuirana.
- **Postolje/baza**—Referentni segment (u ovde će biti reč o nepokretnom postolju);
- **Prihvatinica**—Izvršni element, uobičajeno smešten na poslednjem, tj. slobodnom segmentu.



Slika 2.2: Kinematska šema strukture standardnog manipulatora

Osim softvera koji je otvorenog koda [12], prednost ovog modela je da čitava dokumentacija može se besplatno naći onlajn na njihovom vebajtu [10] i na *GitHub*-u[11].

2.2 Tehničke specifikacije

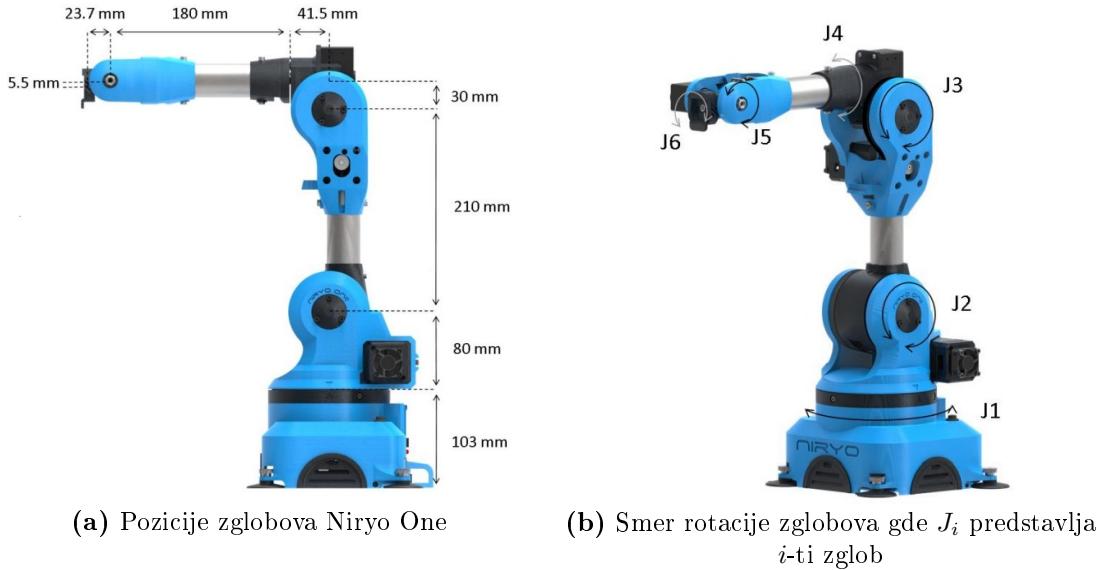
U tabeli 2.2 su date neke osnovne tehničke karakteristike. Robot je podeljen u par glavnih delova koji su prikazani na slici 2.3. Zglobovi i njihova pozicija su dati na slici 2.4. Dimenzije prihvavnice su date na slici 2.5.



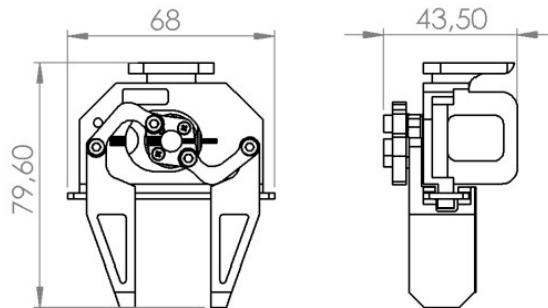
Slika 2.3: Delovi Niryo One manipulatora gde je narandžasta-baza, žuta-rame, zelena-nadlaktica, tirkizna-lakat, plava-podlaktica, ljubičasta-zglob šake, crvena-šaka [13]

Specifikacija	Vrednost
Mehaničke specifikacije	
Stepeni slobode	6
Masa	3,3 kg
Maksimalan dohvat	440 mm
Maksimalna nosivost	0,5 kg
Ponovljivost	±0,5 mm
Materijal	Aluminijum, PLA (3D štampano)
Aktuatori	5 koračna, 2 servo motora
Prihvavnica–hvataljka 1 (engl. <i>Gripper 1</i>)	
Masa	70 g
Dužina pri zatvorenom stanju	80 mm
Maksimalna širina otvaranja	27
Distanca dohvata od baze prihvavnice	60 mm
Opseg rotacije aktuatora	
Zglob 1	−175,00000° ÷ 175,00000°
Zglob 2	−109,44000° ÷ 36,670000°
Zglob 3	−79,890000° ÷ 89,960000°
Zglob 4	−174,76000° ÷ 175,00000°
Zglob 5	−100,00000° ÷ 110,01000°
Zglob 6	−144,96000° ÷ 144,96000°
Električne specifikacije	
Napajanje	11V ÷ 6A
Električna snaga	~ 60W
Hardver	Arduino Mega, Raspberry Pi 3
Senzori motora	Magnetni senzori
Portovi	4 USB, Ethernet
Softverske specifikacije	
Komunikacija	Ethernet, Wi-Fi, Bluetooth, USB
Korisnički interfejs	Veb aplikacija, Android aplikacija, iOS aplikacija, Gamepad
Programerski interfejs	ROS, višebrojni API, izvorni kôd

Tabela 2.2: Tehničke specifikacije gde je opseg rotacije aktuatora dobijem putem pyniryo interfejsa[13, 6]



Slika 2.4: Zglobovi Niryo One manipulatora [13]



Slika 2.5: Dimenzije prihvavnice-hvataljka 1 [13]

Specificirali smo manipulator i softver (ROS) primjenjen u radu. Obzirom da se možemo povezati i upravljati robotom na više načina (tabela 2.2), moramo naglasiti koji će način biti primjenjem za potrebe ovog rada:

- Komunikacija–Ethernet TCP/IP veza,
- Upravljanje–Pajton API pyniryo.

Ethernet je pouzdan i relativno brz način komunikacije koji zahtevni žičanu vezu (bežična nam je ionako nepotrebna), dok je pyniryo najnoviji dostupan Pajton API za Niryo One. Mogli smo i direktno primeniti ROS ali bi nam to uvelo dodatnu složenost koja je suvišna. Korišćeno razvojno okruženje je *VS Code* dok je verzija Pajtona 3.11.4.

Glava 3

Mehanika robota

Kretanje svakog robota je podvrgnuto zakonima mehanike. Klasična analitička mehanika se može podeliti na:

- **Statika** – izučava opterećenja sistema koji se nalazi u statičkoj, tj. nepokretnoj ravnoteži;
- **Kinematika** – određuje geometrijske karakteristike kretanja sistema;
- **Dinamika** – uključuje kinematiku i analizira vezu između sile i materije.

Koristeći klasičnu mehaniku kao osnovu, kao paralelna grana mehanike izrodila se **mehanika robota**. U okviru ove grane možemo navesti par osnovnih oblasti koje su uzajamno povezane i čija ćemo načela primenjivati:

- **Konfiguracioni prostor** – prostor svih mogućih konfiguracija robota. Pod **konfiguracijom robotskog sistema** smatra se pozicija svih tačaka robota;
- **Kretanje krutog tela** – matematički opis kretanja krutog tela u prostoru primenom teorije zavrtnjeva;
- **Direktna kinematika** – zadatak određivanja konfiguracije prihvavnice na osnovu generalisanih koordinata aktuatora;
- **Kinematika brzine i statika** – proračun brzine prihvavnice i statički proračun pri ravnoteži;
- **Inverzna kinematika** – zadatak određivanja generalisanih koordinata aktuatora na osnovu željene konfiguracije prihvavnice.

Druge oblasti koje nećemo izučavati, a spadaju pod okriljen mehanike robote jesu:

- **Kinematika zatvorenih lanca** – kinematika robota sa **zatvorenim kinematskim lancem**. Kinematski lanac predstavlja lanac segmenta i aktuatora robota. On je zatvoren ukoliko je poligon koji sačinjava lanac zatvoren;
- **Dinamika otvorenih lanca** – usložnjavanje problema direktne i inverzne kinematike analiziranjem međudejstva sila i materije;
- **Generisanje trajektorije** – određivanje trajektorije kretanja na osnovu potrebe izvršenja zadatka;

- **Kontaktna mehanika i manipulacija** – modelovanje kontakta između robota i objekta i njegova analiza;
- **Mobilni roboti** – analiza robota sa pokretnim postoljem.

Napomenimo još naučne discipline koje su usko vezane sa mehanikom robota i neizbežne su pri upravljanju jesu

- **Planiranje kretanja** – nalaženje načina kretanja robota koji izuzima kolizije u njegovom konfiguracijskom prostoru;
- **Teorija upravljanja robota** – primene **teorije upravljanja** na robotu na osnovu zadatka.

Po navedenim podelama, evidentno je da je upravljanje robota ozbiljan zadatak. Štaviše, unutar navedenih oblasti postoje više različitih interpretacija i rešenja koje poseduju određene prednosti i mane. Naš pristup biće zasnovan na **teoriji zavrtnjeva**. To je pristup koji podrazumeva analiziranje kretanje i sila analogno mašinskim zavrtnjevima. Glavni rezultati mehanike robota i teorije zavrtnjeva biće primenjeni i samo navedeni, dok se zainteresovanom čitaocu navodi knjiga od Linča i Parka [7].

Rezultati primene navedene teorije su dati na kraju ove glave.

3.1 Teorijska razmatranja

3.1.1 Konfiguracioni prostor

Prostor svih mogućih konfiguracija robota se naziva **konfiguracioni prostor**. Dva aspekta koji ga opisuju jesu

- **Dimenzije prostora** – minimalan broj generalisanih koordinata koji jednoznačno opisuju konfiguraciju robota. Poznato je pod nazivom **broj stepeni slobode robota**;
- **Oblik prostora** – oblik prostora u kom robot konfigurira. Izučava se pomoću matematičke grane **topologija**.

Broj stepeni slobode se može odrediti za robote na osnovu **Grublerove formule** (engl. *Grübler*) koja kaže

$$w = m(N - 1) - \sum_{i=1}^J c_i = m(N - 1 - J) + \sum_{i=1}^J f_i \quad (3.1.1)$$

gde je:

- w – broj stepeni slobode robota podvrgnut nezavisnim vezama,
- m – parametar vrste robota, tj. broj stepeni slobode segmenata robota. $m = 3$ za robote koje se kreću u ravni, a $m = 6$ za robote koje se kreću u prostoru,
- N – broj segmenata robota (postolje mora takođe biti uračunato),
- J – broj veza, aktuatora,

- c_i – broj stepeni slobode koje i -ti zglob ograničava robota,
- f_i – broj stepeni slobode koje i -ti zglob dozvoljava robotu, gde je $m = c_i + f_i$.

Pod jedinstvenim “oblikom” smatramo oblik nekog objekta koji se kontinualnim deformisanjem bez presecanja ne može opisati oblikom drugog tela. Na primer sferu i tetraedar smatramo **topološki ekvivalentnim**.

Oblik prostora se topološki univerzalno opisuje. Korisno je znati oblik kako bi se moglo odrediti određene karakteristike prostora. Međutim, za potrebe proračuna nama to nije pogodno i zato prostor opisujemo određenim reprezentacijama:

- **Eksplicitna reprezentacija** – način opisivanja prostora koristeći minimalan broj koordinata. Često poseduje određene probleme i tačke nepovoljnog ponašanja (*singulariteta*) pa je potrebno nekad više jednakih eksplicitnih reprezentacija različitih delova prostora,
- **Implicitna reprezentacija** – način opisivanja prostora veći broj koordinata ali koje su podvrgnute određenim ograničenja.

Posebni prostori koji su od interesa jesu:

- **Prostor zadatka** – prostor u kome se zadatak robota može opisati i
- **Radni prostor** – moguće konfiguracije prihvavnice robota.

Koristićemo implicitnu reprezentaciju u vidu višedimenzionih matrica i vektora sa posebnim karakteristikama koje će biti razjašnjene u narednom poglavlju.

3.1.2 Kretanje krutog tela

Svako kretanje krutog tela se može opisati pomoću rotacije oko neke ose i translacije duž nje (isto kao kod kretanje tačke na obodu zavrtnja). Ovo je osnova na kojoj se razvija teorija zavrtnja i poznata je kao Česls-Mocijeva teorema (engl. *Chasles-Mozzi*).

Postoje dva koordinatna sistema koja su nam od značaja (oba su fiksna i poklapaju se sa određenom tačkom prostora u određenom trenutku vremena). **Prostorni koordinatni sistem**, u oznaci $\{s\}$ koje je vezano za relativno nepokretan deo prostora i **koordinatni sistem tela od interesa** u oznaci $\{b\}$ koje je tipično vezano za prihvavnicu.

Takođe, zarad skraćenog prikaza, sinuse i kosinuse određenih uglova prikazaćemo kao:

$$\begin{aligned} \cos \theta &= c_\theta, & \sin \theta &= s_\theta, \\ \cos \theta_i &= c_i, & \sin \theta_i &= s_i, \\ \cos \left(\sum_{i=1}^n \theta_i \right) &= c_{12\dots n}, & \sin \left(\sum_{i=1}^n \theta_i \right) &= s_{12\dots n}. \end{aligned}$$

Razmatraćemo prvo čiste rotacije krutog tela, pa tek onda složeno kretanje jer postoje brojne analogije između ova dva kretanja.

Rotacije krutog tela

Rotacije krutog tela se prikazuju pomoću elemenata **specijalne ortogonalne grupe** $SO(3)$, takozvane **grupe ortogonalnih/rotacionih matrica** ili **grupe rotacije krutog tela**. Definisana je kao skup linearnih trasnformacija

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^T = \mathbf{R}^{-1}, \det \mathbf{R} = 1\}. \quad (3.1.2)$$

Ukoliko želimo da naglasimo da se radi o operateru koji transformiše oko određene jedinične ose $\hat{\omega}$ za neki ugao θ , možemo pisati i

$$\mathbf{R} = \text{Rot}(\hat{\omega}, \theta). \quad (3.1.3)$$

Tako, na primer, rotacije oko jediničnih ortogonalnih osa (nezavisno od koordinatnog sistema) su

$$\text{Rot}(\hat{x}, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & -s_\theta \\ 0 & s_\theta & c_\theta \end{bmatrix}, \quad (3.1.4)$$

$$\text{Rot}(\hat{y}, \theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad (3.1.5)$$

$$\text{Rot}(\hat{z}, \theta) = \begin{bmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.1.6)$$

Neka $\mathbf{R} = \text{Rot}(\hat{\omega}, \theta)$ bude operater, tada ukoliko imamo matricu koja predstavlja transformaciju i $\{b\}$ u $\{s\}$ koordinatni sistem \mathbf{R}_{sb} , važi

$$\mathbf{R}\mathbf{R}_{sb} \implies \hat{\omega} = \hat{\omega}_s, \quad (3.1.7)$$

$$\mathbf{R}_{sb}\mathbf{R} \implies \hat{\omega} = \hat{\omega}_b \quad (3.1.8)$$

odnosno u prvom slučaju je osa rotacije u $\{s\}$, a u drugom u $\{b\}$ koordinatnom sistemu.

Dalje, postavlja se pitanje, da li je moguće odrediti elemente rotacione matrice \mathbf{R} , ukoliko je poznata jedinična osa rotacije i ugao rotacije? Jeste i odgovor se može naći u **Rodrigezovoj formuli** (engl. *Rodrigues*) eksponencijacijom

$$\text{Rot}(\boldsymbol{\omega}) = \text{Rot}(\hat{\omega}, \theta) = e^{[\hat{\omega}]\theta} = \mathbf{I} + s_\theta[\hat{\omega}] + (1 - c_\theta)[\hat{\omega}]^2 \quad (3.1.9)$$

gde je

$$[\hat{\omega}] = \begin{bmatrix} 0 & -\hat{\omega}_3 & \hat{\omega}_2 \\ \hat{\omega}_3 & 0 & -\hat{\omega}_1 \\ -\hat{\omega}_2 & \hat{\omega}_1 & 0 \end{bmatrix} \in so(3) \quad (3.1.10)$$

gde je $so(3)$ grupa **eksponencijalnih koordinata**. Drugi, ekvivalentni, način prikazivanja eksponencijalnih koordinata jeste kao putem matričnog izvoda

$$\dot{\mathbf{R}}\mathbf{R}^{-1} = [\boldsymbol{\omega}_s] \quad (3.1.11)$$

$$\mathbf{R}^{-1}\dot{\mathbf{R}} = [\boldsymbol{\omega}_b] \quad (3.1.12)$$

Inverzna operacija **logaritmovanja**, tj. određivanja eksponencijalnih koordinata na osnovu zadate rotacione matrice se može algoritamski predstaviti.

Alogitam logaritmovanja rotacione matrice. Za $\mathbf{R} = [r_{ij}]_{3 \times 3}$,

- (a) Ako je $\mathbf{R} = \mathbf{I}$ onda je $\theta = 0$, a $\hat{\omega} = \mathbf{0}$, tj. neodređeno.
- (b) Ako je trag matrice $\text{tr } \mathbf{R} = -1$ onda je $\theta = \pi$, a $\hat{\omega}$ može imati bilo koji od navedenih mogućih rešenja (tako da $\hat{\omega} \neq \mathbf{0}$):

$$\hat{\omega} = \frac{1}{\sqrt{2(1 + r_{33})}} \begin{bmatrix} r_{13} \\ r_{23} \\ 1 + r_{33} \end{bmatrix} \quad (3.1.13)$$

$$\hat{\omega} = \frac{1}{\sqrt{2(1 + r_{22})}} \begin{bmatrix} r_{12} \\ 1 + r_{22} \\ r_{32} \end{bmatrix} \quad (3.1.14)$$

$$\hat{\omega} = \frac{1}{\sqrt{2(1 + r_{11})}} \begin{bmatrix} 1 + r_{11} \\ r_{21} \\ r_{31} \end{bmatrix}. \quad (3.1.15)$$

Takođe važi u ovom slučaju da je za svako $\hat{\omega}$ i $-\hat{\omega}$ validno rešenje.

- (c) Za sve ostale slučajeve je $\theta = \arccos((\text{tr } \mathbf{R} - 1)/2) \in [0, \pi]$ i važi

$$[\hat{\omega}] = \frac{1}{2s_\theta} (\mathbf{R} - \mathbf{R}^T). \quad (3.1.16)$$

Složeno kretanje krutog tela

Poznato je da linearnim transformacijama nije moguće opisati afinu transformaciju, u našem slučaju translaciju. Kako bismo rešili ovaj nedostatak, povećaćemo dimenziju naše transformacione matrice. Generalizacija naše specijalne ortogonalne grupe je **specijalna euklidova grupa $SE(3)$** , takođe poznata kao **grupa homogenih transformacionih matrica**,

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}^T & 1 \end{bmatrix} \mid \mathbf{R} \in SO(3), \mathbf{p} \in \mathbb{R}^{3 \times 1}, \mathbf{0}^T = [0 \ 0 \ 0] \right\}. \quad (3.1.17)$$

Kako bismo transformisali trodimenzionalni vektor, uvodimo **homogenu reprezentaciju vektora**

$$\bar{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{bmatrix}, \quad \mathbf{a} \in \mathbb{R}^{3 \times 1} \quad (3.1.18)$$

te afina transformacija homogenog vektora ima oblik

$$\mathbf{T}\bar{\mathbf{a}} = \begin{bmatrix} \mathbf{R}\mathbf{a} + \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}\mathbf{a} \\ 1 \end{bmatrix} \quad (3.1.19)$$

gde je $\mathbf{T} = (\mathbf{R}, \mathbf{p})$ i gde je $\mathbf{T}\mathbf{a}$ bez homogenog zapisa skraćena forma za $\mathbf{R}\mathbf{a} + \mathbf{p}$.

Gотов obrazac za inverznu matricu matrice \mathbf{T} glasi

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{p} \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (3.1.20)$$

Ukoliko želimo da naglasimo da se radi o operateru koji vrši homogenu transformaciju, u kontekstu $SE(3)$ grupe, oko određene jedinične ose $\hat{\omega}$ za neki ugao θ , možemo pisati i

$$\mathbf{T} = \text{Rot}(\boldsymbol{\omega}) = \text{Rot}(\hat{\omega}, \theta) = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.1.21)$$

ili ukoliko je čista translacija u pitanju definisano vektorom položaja $\mathbf{p} \in \mathbb{R}^{3 \times 1}$

$$\mathbf{T} = \text{Trans}(\mathbf{p}) = \text{Trans}(\hat{\mathbf{p}}, \|\mathbf{p}\|) = \begin{bmatrix} \mathbf{O} & \mathbf{p} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.1.22)$$

gde je $\mathbf{O} \in \mathbb{R}^{3 \times 3}$ nula matrica.

Neka \mathbf{T} bude operater, tada ukoliko imamo matricu koja predstavlja transformaciju i $\{b\}$ u $\{s\}$ koordinatni sistem \mathbf{T}_{sb} , važi

$$\mathbf{T}\mathbf{T}_{sb} = \text{Trans}(\mathbf{p}) \text{Rot}(\boldsymbol{\omega})\mathbf{T}_{sb}, \quad \boldsymbol{\omega} = \boldsymbol{\omega}_s, \quad \mathbf{p} = \mathbf{p}_s \quad (3.1.23)$$

$$\mathbf{T}_{sb}\mathbf{T} = \mathbf{T}_{sb} \text{Trans}(\mathbf{p}) \text{Rot}(\boldsymbol{\omega}), \quad \boldsymbol{\omega} = \boldsymbol{\omega}_b, \quad \mathbf{p} = \mathbf{p}_b. \quad (3.1.24)$$

Naveli smo da se kretanje može posmatrati kao rotacija i translacija, tj. istovremeno okretanje oko ose zavrtnja i kretanje duž iste, ovo se matematički opisuje kao takozvani **zakret**, (slika 3.1)

$$\boldsymbol{\mathcal{V}} = \begin{bmatrix} \boldsymbol{\omega}_{\mathcal{V}} \\ \mathbf{v}_{\mathcal{V}} \end{bmatrix} = \begin{bmatrix} \hat{s}\dot{\theta} \\ -\hat{s}\dot{\theta} \times \mathbf{q} + h\hat{s}\dot{\theta} \end{bmatrix} \in \mathbb{R}^{6 \times 1}, \quad \boldsymbol{\omega}_{\mathcal{V}}, \mathbf{v}_{\mathcal{V}} \in \mathbb{R}^{3 \times 1} \quad (3.1.25)$$

gde je

- \mathbf{q} – radijus vektor ose zavrtnja,
- \hat{s} – jedinični vektor ose,
- h – korak zavrtnja koji predstavlja odnos linearne i ugaone brzine zavrtnja,
- $\mathbf{v}_{\mathcal{V}}$ predstavlja brzinu kretanja koordinatnog početka (čak i kada on realno miruje) unutar koordinatnog sistema u kome je zakret definisan.

U slučaju da je zakret čista linearna brzina, tada je $h = \infty$ te ćemo reći da je

$$\boldsymbol{\mathcal{V}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_{\mathcal{V}} \end{bmatrix}. \quad (3.1.26)$$

Zakret možemo prikazati u pogodnijem obliku kao promenu generalisane koordinate kretanja $\dot{\theta}$ duž ose zavrtnja

$$\boldsymbol{\mathcal{V}} = \mathcal{S}\dot{\theta} \quad (3.1.27)$$

gde je

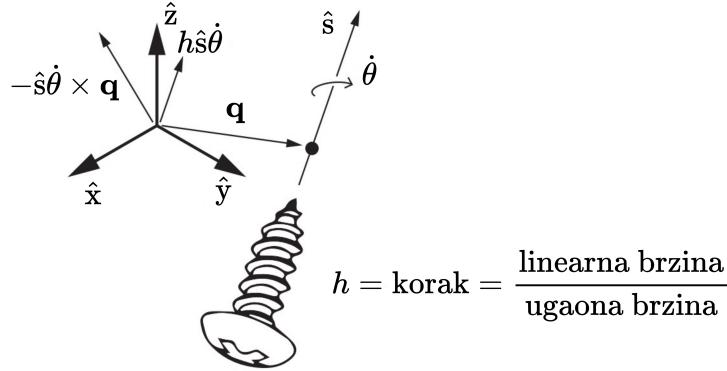
$$\mathcal{S} = \begin{bmatrix} \boldsymbol{\omega}_{\mathcal{S}} \\ \mathbf{v}_{\mathcal{S}} \end{bmatrix} \in \mathbb{R}^{6 \times 1}, \quad \boldsymbol{\omega}_{\mathcal{V}}, \mathbf{v}_{\mathcal{V}} \in \mathbb{R}^{3 \times 1}, \quad (3.1.28)$$

a važi (i) $\|\boldsymbol{\omega}_{\mathcal{S}}\| = 1$ ili za linearno kretanje (ii) $\boldsymbol{\omega}_{\mathcal{S}} = \mathbf{0}$ i $\|\mathbf{v}_{\mathcal{S}}\| = 1$.

Proširivanjem operatera $[\cdot]$ iz jednačine 3.1.10 na zakret i osu zavrtnja možemo napisati da je

$$[\boldsymbol{\mathcal{V}}] = \begin{bmatrix} [\boldsymbol{\omega}_{\mathcal{V}}] & \mathbf{v}_{\mathcal{V}} \\ \mathbf{0}^T & 0 \end{bmatrix} \in se(3), \quad (3.1.29)$$

$$[\mathcal{S}] = \begin{bmatrix} [\boldsymbol{\omega}_{\mathcal{S}}] & \mathbf{v}_{\mathcal{S}} \\ \mathbf{0}^T & 0 \end{bmatrix} \in se(3). \quad (3.1.30)$$



Slika 3.1: Osa zavrtnja \mathcal{S} opisana parametarski pomoću radijusa vektora ose zavrtnja \mathbf{q} , jediničnog vektora ose $\hat{\mathbf{s}}$ i korak zavrtnja h [7]

gde su $[\mathcal{V}]$ i $[\mathcal{S}]$ deo grupe **eksponencijalnih koordinata za složeno kretanje** $se(3)$.

Posebno značajan operater koji služi za promenu reprezentacije koordinatnog sistema određenih veličina jeste **adjungovano preslikavanje**. Za $\mathbf{T} = (\mathbf{R}, \mathbf{p}) \in SE(3)$ je adjungovano preslikavanje opisano kao

$$[\text{Ad}_{\mathbf{T}}] = \begin{bmatrix} \mathbf{R} & \mathbf{O} \\ [\mathbf{p}]\mathbf{R} & \mathbf{R} \end{bmatrix} \in \mathbb{R}^{6 \times 6}. \quad (3.1.31)$$

Za sada ćemo reći da možemo pomoću adjungovanog preslikavanja da promenimo koordinatni sistem ose zavrtnja i zakreta

$$\mathcal{V}_s = [\text{Ad}_{\mathbf{T}_{sb}}]\mathcal{V}_b, \quad (3.1.32)$$

$$\mathcal{S}_s = [\text{Ad}_{\mathbf{T}_{sb}}]\mathcal{S}_b, \quad (3.1.33)$$

dok ćemo kasnije videti još primena.

Eksponencijacijom možemo definisati \mathbf{T} :

(a) Za $\mathcal{S} = (\boldsymbol{\omega}_s, \mathbf{v}_s)$, $\|\boldsymbol{\omega}_s\|$ i $\theta \in \mathbb{R}$, eksponencijacija glasi

$$e^{[\mathcal{S}]\theta} = \begin{bmatrix} e^{[\boldsymbol{\omega}_s]\theta} & G(\theta)\mathbf{v}_s \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.1.34)$$

$$G(\theta) = \theta\mathbf{I} + (1 - c_\theta)[\boldsymbol{\omega}_s] + (\theta - s_\theta)[\boldsymbol{\omega}_s]^2. \quad (3.1.35)$$

(b) U slučaju da je $\|\boldsymbol{\omega}_s\| = 0$, eksponencijacija glasi

$$e^{[\mathcal{S}]\theta} = \begin{bmatrix} \mathbf{I} & \theta\mathbf{v}_s \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (3.1.36)$$

Ove jednačine matematički opisuju Česls-Mocijevu teoremu.

Inverzna operacija, logaritmovanje, definišemo algoritmom.

Alogritam logaritmovanja homogene matrice. Za $\mathbf{T} = (\mathbf{R}, \mathbf{p})$,

(a) Ako je $\mathbf{R} = \mathbf{I}$ onda je $\boldsymbol{\omega}_s = \mathbf{0}$, $\mathbf{v}_s = \mathbf{p}/\|\mathbf{p}\|$ i $\theta = \|\mathbf{p}\|$.

(b) Inače upotrebom algoritma logaritmovanja na \mathbf{R} odrediti $\boldsymbol{\omega}_S$ (tamo je napisano kao $\hat{\omega}$) i θ . \mathbf{v}_S odrediti kao

$$\mathbf{v}_S = G^{-1}(\theta)\mathbf{p}, \quad (3.1.37)$$

$$G^{-1}(\theta) = \frac{1}{\theta}\mathbf{I} - \frac{1}{2}[\boldsymbol{\omega}_S] + \left(\frac{1}{\theta} - \frac{1}{2}\operatorname{ctg}\frac{\theta}{2}\right)[\boldsymbol{\omega}_S]^2. \quad (3.1.38)$$

Dosada smo razmatrali brzine, sada ćemo pokazati kako ćemo reprezentovati sile koje deluju na robot. Vektor

$$\mathcal{F} = \begin{bmatrix} \mathbf{m} \\ \mathbf{f} \end{bmatrix} \in \mathbb{R}^{6 \times 1}, \quad \mathbf{m} = \mathbf{r} \times \mathbf{f}, \quad \mathbf{m}, \mathbf{f} \in \mathbb{R}^{3 \times 1}, \quad (3.1.39)$$

definisaćemo kao **ključ** ili **trzaj**. Njegove komponente su

- \mathbf{f} – sila koja deluje na telo,
- \mathbf{r} – moment koji deluje na radijus vektor napadne tačke sile \mathbf{f} ,
- \mathbf{m} – moment koji deluje na telo.

Primenom adjungovane transformacije možemo promeniti koordinatni sistem trzaja

$$\mathcal{F}_s = [\operatorname{Ad}_{\mathbf{T}_{bs}}]^T \mathcal{F}_b. \quad (3.1.40)$$

Prethodno opisani pojmovi i jednačine su ključne za teoriju zavrtnja. Iz nje, uz određene dodatke, definisaćemo ostatak mehanike robota.

3.1.3 Direktna kinematika

Direktna kinematika rešava problem određivanja pozicije i orijentacije prihvavnice na osnovu podataka o generalisanim koordinatama aktuatora. Drugačije rečeno, predstavlja funkciju $f : \mathbb{R}^n \rightarrow \mathbb{R}^{4 \times 4} \in SE(3)$,

$$f(\boldsymbol{\theta}) = \mathbf{T}.$$

Dva načina na koja možemo definisati direktnu kinematiku jeste u $\{s\}$ koordinatnom sistemu (jednačina 3.1.41) i $\{b\}$ koordinatnom sistemu (jednačina 3.1.42)

$$\mathbf{T} = \prod_{i=1}^n e^{[\mathcal{S}_i]\theta_i} \mathbf{M}, \quad (3.1.41)$$

$$\mathbf{T} = \mathbf{M} \prod_{i=1}^n e^{[\mathcal{B}_i]\theta_i}, \quad (3.1.42)$$

$$\mathcal{B}_i = [\operatorname{Ad}_{\mathbf{M}^{-1}}]\mathcal{S}_i, \quad (3.1.43)$$

gde je:

- θ_i – i -ta generalisana koordinata,
- \mathbf{M} – nulta konfiguracija robota kada važi $\theta_1 = \dots = \theta_n = 0$,
- \mathcal{S}_i – i -ta osa rotacije zgloba idući od postolja ka prihvavnici definisana u $\{s\}$ koordinatnom sistemu,
- \mathcal{B}_i – i -ta osa rotacije zgloba idući od prihvavnice ka postolju definisana u $\{b\}$ koordinatnom sistemu.

3.1.4 Kinematika brzine i statika

U robotici mi zadajemo generalisane koordinate i njihove izvode po vremenu, što rezultuje u određeno kretanje $\{b\}$ koordinatnog sistema. Transformacija $\mathbf{J} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ se naziva **Jakobijan** i predstavlja preslikavanja iz skupa izvoda generalisanih brzina aktuatora $\dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ u skup mogućih zakreta

$$\mathcal{V} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \quad (3.1.44)$$

Vektori kolone Jakobijana \mathbf{J}_{si} za $i = 1, \dots, n$ se mogu odrediti u $\{s\}$ koordinatama

$$\begin{aligned} \mathbf{J}_{s1} &= \mathcal{S}_1 \\ \mathbf{J}_{si}(\boldsymbol{\theta}) &= [\text{Ad}_{\exp([\mathcal{S}_1]\theta_1)} \dots \exp([\mathcal{S}_{i-1}]\theta_{i-1})] \mathcal{S}_i \end{aligned} \quad (3.1.45)$$

dok se u $\{b\}$ koordinatama definišu kao

$$\begin{aligned} \mathbf{J}_{bn} &= \mathcal{B}_n \\ \mathbf{J}_{bi}(\boldsymbol{\theta}) &= [\text{Ad}_{\exp(-[\mathcal{B}_n]\theta_n)} \dots \exp(-[\mathcal{B}_{i+1}]\theta_{i+1})] \mathcal{B}_i \end{aligned} \quad (3.1.46)$$

Ukoliko je poznat jedan oblik, u drugi se prelazi adjungovanom transformacijom

$$\mathbf{J}_s(\boldsymbol{\theta}) = [\text{Ad}_{\mathbf{T}_{sb}} \mathbf{J}_b(\boldsymbol{\theta})] \quad (3.1.47)$$

Razmatrajući manipulator sa n stepeni slobode u stanju statičke ravnoteže, sledi da je

$$\boldsymbol{\tau} = \mathbf{J}_b^T(\boldsymbol{\theta}) \mathcal{F}_b = \mathbf{J}_s^T(\boldsymbol{\theta}) \mathcal{F}_s \quad (3.1.48)$$

gde je $\boldsymbol{\tau} \in \mathbb{R}^n$ vektor momenta/sila koje je potrebno da generišu aktuatori kako bi manipulator održao ravnotežu.

Kinematska singularitet je konfiguracija robota u kojoj on onemogućen da se kreće u određenim pravcima. Govoreći matematičkim jeziok, to su sve konfiguracije robota $\boldsymbol{\theta} \in \mathbb{R}^n$ kada za Jakobijan $\mathbf{J}(\boldsymbol{\theta}) \in \mathbb{R}^{m \times n}$ važi

$$\text{rang } \mathbf{J} \neq \max\{m, n\}. \quad (3.1.49)$$

Postoje brojne konfiguracije za svaki manipulator koji može rezultovati u kinematsku singularitet, neki od najčešćih za rotoidne zglobove se javljaju kada:

- (a) Dve ose su kolinearne,
- (b) Tri ose su koplanarne i paralelne,
- (c) Četiri ose se seku u istoj tački,
- (d) Četiri ose su koplanarne,
- (e) Šest ose seku istu liniju.

Elipsoid manipulabilnosti i ellipsoid sile opisuju sa kojom težinom može robot da se kreće i izaziva silu u određenim pravcima respektivno. Ellipsoid manipulabilnosti je oblika

$$\dot{\mathbf{q}}^T \mathbf{A}^{-1} \dot{\mathbf{q}} = 1 \quad (3.1.50)$$

gde važi $\|\dot{\boldsymbol{\theta}}\| = 1$, a pomenute veličine u jednačini 3.1.50 su:

- $\mathbf{q} \in \mathbb{R}^m$ – koordinate prostora zadatka,

- $\mathbf{A} = \mathbf{J}\mathbf{J}^T \in \mathbb{R}^{m \times m}$ – simetrično pozitivno definitna matrica (isto važi i za matricu \mathbf{A}^{-1}).

Ellipsoid sile je oblika

$$\mathbf{f}^T \mathbf{B}^{-1} \mathbf{f} = 1 \quad (3.1.51)$$

gde važi $\|\dot{\boldsymbol{\theta}}\| = 1$, a pomenute veličine u jednačini 3.1.51 su:

- $\mathbf{f} \in \mathbb{R}^m$ – sile koje deluju na aktuator,
- $\mathbf{B} = (\mathbf{J}\mathbf{J}^T)^{-1} = \mathbf{A}^{-1} \in \mathbb{R}^{m \times m}$ – simetrično pozitivno definitna matrica (isto važi i za matricu \mathbf{B}^{-1}).

Kvantitativne mere ovih elipsoida, odnosno kvantitativne mere “blizine” robota singulitetu uključuju

$$\mu_1(\mathbf{A}) = \sqrt{\frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}} \geq 1, \quad (3.1.52)$$

$$\mu_2(\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \geq 1, \quad (3.1.53)$$

$$\mu_3(\mathbf{A}) = \sqrt{\det \mathbf{A}} = \sqrt{\prod_{i=1}^m \lambda_i}, \quad (3.1.54)$$

sa λ je obeležena sopstvena vrednost matrice \mathbf{A} . Isto važi i za elipsoid sile ako se u prethodnim jednačinama \mathbf{A} zameni za \mathbf{B} .

3.1.5 Inverzna kinematika

Ako direktna kinematika rešava problem određivanja pozicije i orijentacije prihvavnice na osnovu podataka o generalisanim koordinatama aktuatora, inverzna kinematika ima suprotan zadatak. **Inverzna kinematika** teži da odredi koje generalisane koordinate aktuatora odgovaraju određenoj poziciji i orijentaciji prihvavnice. Drugačije rečeno, predstavlja funkciju

$$f(\mathbf{T}) = \boldsymbol{\theta}.$$

Nalaženje rešenje ove funkcije nije nimalo naivan zadatak. Brojna istraživanja se i dan danas vrše na ovoj temi. Za razliku od direktne kinematike, u inverznoj kinematici ne postoji uvek analitičko rešenje ili je nepovoljno za praksu. Tipično za $w = 6$ stepeni slobode postoji konačan broj rešenja dok za $w > 6$ najčešće ima beskonačno mnogo rešenja.

U ovom radu razmatraćemo postupak analitičkog rešavanja koje je zavisi od geometrije robota i takođe ćemo izučiti iterativnu numeričku metodu.

Paden-Kahan podproblemi [9] predstavljaju rešene probleme koje se mogu iskoristiti da se inverzna kinematika manipulatora, koju delimo u podprobleme, reši. One koriste posebnu funkciju koja je slična $\text{arctg}(y/x)$ funkciji ali ima veći domen. Za $x, y \in \mathbb{R}$,

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0, \\ \arctan\left(\frac{y}{x}\right) + \pi & x < 0 \wedge y \geq 0, \\ \arctan\left(\frac{y}{x}\right) - \pi & x < 0 \wedge y < 0, \\ \frac{\pi}{2} & x = 0 \wedge y > 0, \\ -\frac{\pi}{2} & x = 0 \wedge y < 0, \\ \text{undefined} & x = 0 \wedge y = 0. \end{cases} \quad (3.1.55)$$

S tim u vezi, uslovi korišćenja Paden-Kahan podproblema i njihova rešenja glase:

Podproblem (1) Neka osa zavrtnja \mathcal{S} bude čista rotacija $h = 0$ od jedne do druge tačke $\mathbf{p}, \mathbf{q} \in \mathbb{R}^3$,

$$e^{[\mathcal{S}]\theta} \mathbf{p} = \mathbf{q}. \quad (3.1.56)$$

Ugao θ koji korespondira ovoj transformaciji ima oblik

$$\theta = \text{atan2}(\boldsymbol{\omega}_{\mathcal{S}} \cdot (\mathbf{u}' \times \mathbf{v}'), \mathbf{u}' \cdot \mathbf{v}') \quad (3.1.57)$$

gde je

- $\mathbf{u} = (\mathbf{p} - \mathbf{r})$;
- $\mathbf{v} = (\mathbf{q} - \mathbf{r})$;
- $\mathbf{r} \in \mathbb{R}^3$ tačka ose \mathcal{S} , a određuje se na sledeći način. Polazeći od jednačine 3.1.25 za $h = 0$ možemo napisati da je

$$\mathbf{v}_{\mathcal{S}} = -\boldsymbol{\omega}_{\mathcal{S}} \times \mathbf{r} = -[\boldsymbol{\omega}_{\mathcal{S}}] \mathbf{r} \quad (3.1.58)$$

gde smo primenili drugačiji zapis vektorskog proizvoda

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}] \mathbf{b}, \quad \mathbf{a}, \mathbf{b} \in \mathbb{R}^3. \quad (3.1.59)$$

Pošto ovaj linearan sistem ima beskonačno mnogo rešenja, a nama treba jedno rešenje (koje određujemo numerički), koristićemo izraz

$$\mathbf{r} = -[\boldsymbol{\omega}_{\mathcal{S}}]^{\dagger} \mathbf{v}_{\mathcal{S}} \quad (3.1.60)$$

odakle je sa $-[\boldsymbol{\omega}_{\mathcal{S}}]^{\dagger}$ obeležena **Mor-Penroz pseudoinverzna matrica**, a postupak nalaženja se može naći u prilogu D od knjige od Linča i Parka [7];

- $\mathbf{u}' = \mathbf{u} - \text{proj}_{\boldsymbol{\omega}_{\mathcal{S}}} \mathbf{u}$;
- $\mathbf{v}' = \mathbf{v} - \text{proj}_{\boldsymbol{\omega}_{\mathcal{S}}} \mathbf{v}$;
- $\text{proj}_{\mathbf{b}} \mathbf{a} = (\mathbf{a} \cdot \hat{\mathbf{b}}) \hat{\mathbf{b}}$ vektorska projekcija vektora \mathbf{a} na osu vektora \mathbf{b} .

Uslovi korišćenja podproblema su

$$\boldsymbol{\omega}_{\mathcal{S}} \cdot \mathbf{u} = \boldsymbol{\omega}_{\mathcal{S}} \cdot \mathbf{v}, \quad (3.1.61)$$

$$\|\mathbf{u}'\| = \|\mathbf{v}'\| \Leftrightarrow \|\mathbf{u}\| = \|\mathbf{v}\|. \quad (3.1.62)$$

U slučaju da je $\mathbf{u}' = \mathbf{0}$, postoji beskonačno mnogo rešenja.

Podproblem (2) Neka ose zavrtnjeva \mathcal{S}_1 i \mathcal{S}_2 budu čista rotacija $h = 0$ i presecaju se ili su paralelne i neka one predstavljaju dve sekvenčalne transformacije koje vode tačku \mathbf{p} do tačke \mathbf{q} gde $\mathbf{p}, \mathbf{q} \in \mathbb{R}^3$,

$$e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} \mathbf{p} = \mathbf{q}. \quad (3.1.63)$$

U slučaju da se ose poklapaju,

$$\mathcal{S}_1 = \pm \mathcal{S}_2 \implies \mathcal{S}_1 \cdot \mathcal{S}_2 = \pm \|\mathcal{S}_1\|^2 = \pm \|\mathcal{S}_2\|^2 \quad (3.1.64)$$

tada možemo reći da je

$$e^{[\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2]\theta_2} \mathbf{p} = e^{[\mathcal{S}_1]\theta_1} e^{[-\mathcal{S}_1]\theta_2} \mathbf{p} = e^{[\mathcal{S}_1](\theta_1 - \theta_2)} \mathbf{p} = \mathbf{q}. \quad (3.1.65)$$

i primenjujemo podproblem jedan gde ćemo reći da je

$$\theta = \theta_1 \pm \theta_2 \quad (3.1.66)$$

i svaka kombinacija generalisanih koordinata θ_1 i θ_2 koje zadovoljavaju prethodno naveden izraz, je rešenje. Napominjemo da smo razmatrali slučaj da je osa \mathcal{S}_1 referenta, tj. njen pravac je referentan, inače za slučaj da je druga osa referenta važi

$$\theta = \theta_2 \pm \theta_1. \quad (3.1.67)$$

Kada se ose samo sekut, jednačinu 3.1.63 svodimo na dve jednačine

$$\begin{aligned} e^{[\mathcal{S}_1]\theta_1} \mathbf{p} &= \mathbf{c} \\ e^{-[\mathcal{S}_2]\theta_2} \mathbf{q} &= \mathbf{c} \end{aligned} \quad (3.1.68)$$

i rešavamo ih isto kao podproblem 1. Veličine koje treba odrediti su

- $\mathbf{c} = \mathbf{z} + \mathbf{r}$;
- $\mathbf{r} \in \mathbb{R}^3$ tačka presecanja osa. Pošto, u opštem slučaju, primena jednačine 3.1.60 dobijamo

$$\mathbf{r}_i = -[\boldsymbol{\omega}_{\mathcal{S}_i}]^\dagger \mathbf{v}_{\mathcal{S}_i}, \quad i = 1, 2$$

dobijamo $\mathbf{r}_1 \neq \mathbf{r}_2$, moramo primeniti drugu metodu nalaženja vektora $\mathbf{r} = \mathbf{r}_1 = \mathbf{r}_2$. Svojstvo vektorskog proizvoda da $\mathbf{a} \times \mathbf{a} = \mathbf{0}$ nam omogućuje da jednačinu 3.1.58 zapišemo kao (gde ćemo umesto \mathbf{r} pisati \mathbf{w})

$$-[\boldsymbol{\omega}_{\mathcal{S}}] \mathbf{w} = -[\boldsymbol{\omega}_{\mathcal{S}}] (\mathbf{r} + \xi \boldsymbol{\omega}_{\mathcal{S}}) = \mathbf{v}_{\mathcal{S}}, \quad \xi \in \mathbb{R} \quad (3.1.69)$$

Prema uslovu da je $\mathbf{w}_1 = \mathbf{w}_2$ dobijamo

$$\mathbf{r} + \xi_1 \boldsymbol{\omega}_{\mathcal{S}_1} = \mathbf{r} + \xi_2 \boldsymbol{\omega}_{\mathcal{S}_2}, \quad (3.1.70)$$

manipulacijom dobijamo linearan sistem

$$[\boldsymbol{\omega}_{\mathcal{S}_1} \quad -\boldsymbol{\omega}_{\mathcal{S}_2}] \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \mathbf{w}_1 - \mathbf{w}_2 \quad (3.1.71)$$

koje rešavamo primenom pseudoinversa

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = [\boldsymbol{\omega}_{\mathcal{S}_1} \quad -\boldsymbol{\omega}_{\mathcal{S}_2}]^\dagger (\mathbf{w}_1 - \mathbf{w}_2). \quad (3.1.72)$$

Proverom da li važi jednakost 3.1.70 za vrednosti ξ_1 i ξ_2 iz jednačine 3.1.72 možemo utvrditi da li se ose zavrtnja presecaju ili ne;

- $\mathbf{z} = \alpha \boldsymbol{\omega}_{\mathcal{S}_1} + \beta \boldsymbol{\omega}_{\mathcal{S}_2} + \gamma (\boldsymbol{\omega}_{\mathcal{S}_1} \times \boldsymbol{\omega}_{\mathcal{S}_2})$;
-

$$\begin{aligned} \alpha &= \frac{(\boldsymbol{\omega}_{\mathcal{S}_1}^T \boldsymbol{\omega}_{\mathcal{S}_2}) \boldsymbol{\omega}_{\mathcal{S}_2}^T \mathbf{u} - \boldsymbol{\omega}_{\mathcal{S}_1}^T \mathbf{v}}{(\boldsymbol{\omega}_{\mathcal{S}_1}^T \boldsymbol{\omega}_{\mathcal{S}_2})^2 - 1}, \\ \beta &= \frac{(\boldsymbol{\omega}_{\mathcal{S}_1}^T \boldsymbol{\omega}_{\mathcal{S}_2}) \boldsymbol{\omega}_{\mathcal{S}_1}^T \mathbf{v} - \boldsymbol{\omega}_{\mathcal{S}_2}^T \mathbf{u}}{(\boldsymbol{\omega}_{\mathcal{S}_1}^T \boldsymbol{\omega}_{\mathcal{S}_2})^2 - 1}, \\ \gamma &= \pm \sqrt{\frac{\|\mathbf{u}\|^2 - \alpha^2 - \beta^2 - 2\alpha\beta \boldsymbol{\omega}_{\mathcal{S}_1}^T \boldsymbol{\omega}_{\mathcal{S}_2}}{\|\boldsymbol{\omega}_{\mathcal{S}_1} \times \boldsymbol{\omega}_{\mathcal{S}_2}\|^2}}; \end{aligned}$$

- $\mathbf{u} = \mathbf{p} - \mathbf{r}$;
- $\mathbf{v} = \mathbf{q} - \mathbf{r}$.

Podproblem može imati po dva rešenja, po jedno rešenje i nijedno rešenje za par uglova θ_1 i θ_2 . Dva para rešenja dobijamo kada $\gamma \neq 0$, dok jedan par kada važi jednakost 3.1.64 ili $\gamma = 0$. Uslovi korišćenja podproblema, pored uslova presecanja osa, su (implikacija ovih uslova sledi iz jednačina 3.1.68)

$$\boldsymbol{\omega}_{\mathcal{S}_2} \cdot \mathbf{u} = \boldsymbol{\omega}_{\mathcal{S}_2} \cdot \mathbf{z}, \quad (3.1.73)$$

$$\boldsymbol{\omega}_{\mathcal{S}_1} \cdot \mathbf{v} = \boldsymbol{\omega}_{\mathcal{S}_1} \cdot \mathbf{z}, \quad (3.1.74)$$

$$\|\mathbf{u}\|^2 = \|\mathbf{z}\|^2 = \|\mathbf{v}\|^2. \quad (3.1.75)$$

Podproblem (3) Neka osa zavrtnja \mathcal{S} bude čista rotacija $h = 0$ od jedne tačke \mathbf{p} dok ne bude bila za $\delta \geq 0$ daleko od tačke \mathbf{q} , gde $\mathbf{p}, \mathbf{q} \in \mathbb{R}^3$, tada tražimo rešenje za θ u jednačini

$$\|\mathbf{q} - e^{[\mathcal{S}]\theta} \mathbf{p}\| = \delta. \quad (3.1.76)$$

Ukoliko je $\delta = 0$, tada se jednačina 3.1.76 svodi na 3.1.56, tj. na podproblem jedan

$$\|\mathbf{q} - e^{[\mathcal{S}]\theta} \mathbf{p}\| = 0 \implies \mathbf{q} - e^{[\mathcal{S}]\theta} \mathbf{p} = \mathbf{0} \implies e^{[\mathcal{S}]\theta} \mathbf{p} = \mathbf{q}.$$

Ugao θ koji korespondira ovoj transformaciji ima oblik

$$\theta = \text{atan2}(\boldsymbol{\omega}_{\mathcal{S}} \cdot (\mathbf{u}' \times \mathbf{v}'), \mathbf{u}' \cdot \mathbf{v}') \pm \arccos \left(\frac{\|\mathbf{u}'\|^2 + \|\mathbf{v}'\|^2 - \delta'^2}{2\|\mathbf{u}'\| \|\mathbf{v}'\|} \right) \quad (3.1.77)$$

gde je

- $\mathbf{u} = (\mathbf{p} - \mathbf{r})$;
- $\mathbf{v} = (\mathbf{q} - \mathbf{r})$;
- $\mathbf{r} \in \mathbb{R}^3$ tačka ose \mathcal{S} (videti podproblem jedan za više detalja);
- $\mathbf{u}' = \mathbf{u} - \text{proj}_{\boldsymbol{\omega}_{\mathcal{S}}} \mathbf{u}$;
- $\mathbf{v}' = \mathbf{v} - \text{proj}_{\boldsymbol{\omega}_{\mathcal{S}}} \mathbf{v}$;
- $\delta'^2 = \delta^2 - |\boldsymbol{\omega}_{\mathcal{S}} \cdot (\mathbf{p} - \mathbf{q})|$

Ako je δ veće od naše željene dužine ili ako se leva i desna strana jednačine 3.1.76 ne poklapaju za ugao θ prema jednačini 3.1.77, onda podproblem nema rešenje.

Kada su nam analitička rešenja nedostupna ili je njihov oblik nepogodan za kompjutersku obradu, možemo pribeci numeričkim rešenjima. Metod koji ćemo razjasniti je **Njutn-Rapsonova metoda** i on glasi za $\{s\}$ koordinatni sistem:

- Neka je data željena konačna konfiguracija \mathbf{T}_{sk} i početni nagadaj $\boldsymbol{\theta}^0 \in \mathbb{R}^n$, neka je takođe $i = 0$.
- Neka je $[\mathcal{V}_b] = \log(\mathbf{T}_{sb}^{-1}(\boldsymbol{\theta}^i) \mathbf{T}_{sk})$. Dokle god važi $\|\boldsymbol{\omega}_{\mathcal{V}_b}\| > \epsilon_{\boldsymbol{\omega}}$ i $\|\mathbf{v}_{\mathcal{V}_b}\| > \epsilon_{\mathbf{v}}$ za male vrednosti $\epsilon_{\boldsymbol{\omega}}, \epsilon_{\mathbf{v}}$:
 - Neka je $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i + \mathbf{J}_b^\dagger(\boldsymbol{\theta}^i) \mathcal{V}_b$.

Redni broj	Dužina L u mm
1	103
2	80
3	210
4	30
5	41.5
6	180
7	23.7
8	5.5

Tabela 3.1: Karakteristične dužine određene prema slici 2.4a

- Povećati i za 1.

U prostornim koordinatama je moguće koristiti metodu ako napišemo da je

$$\begin{aligned}\mathcal{V}_s &= [\text{Ad}_{\mathbf{T}_{sb}}]\mathcal{V}, \\ \theta^{i+1} &= \theta^i + \mathbf{J}_s^\dagger \mathcal{V}_s.\end{aligned}$$

Navedena metoda neće uvek konvergirati nekom rešenju. Kako bismo osigurali konvergenciju rešenja možemo:

- Započeti kretanje robota od poznate konfiguracije i polako menjati željenu konfiguraciju \mathbf{T}_{sk} u odnosu na frekvenciju korišćenja metode,
- Koristiti aproksimativno analitičko rešenje za parametar početni nagadaj θ^0 .

3.2 Mehaničke karakteristike Niryo One manipulatora

Manipulator se može uprostiti i grafički prikazati kao prema slici 3.2 gde smo prostorni koordinatni sistem $\{s\}$ i koordinatnom sistemu prihvavnice $\{b\}$ postavili na standardnim pozicijama sa našim tipom robota (što se takođe slaže sa pyniryo interfejsom).

Nulta konfiguracija Niryo One manipulatora je prema tome

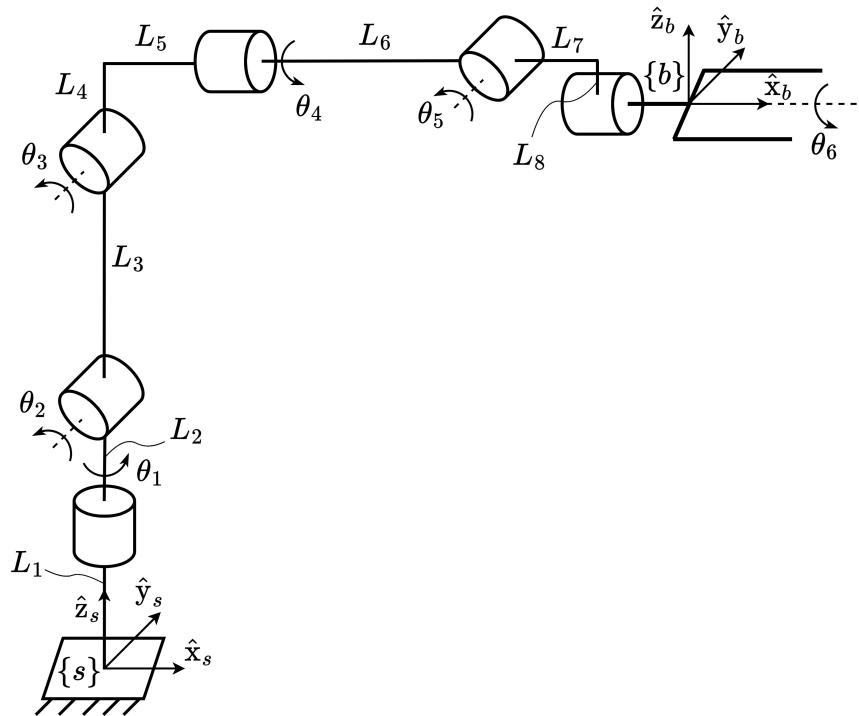
$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & L_{57} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{14} - L_8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.1)$$

gde je skraćeni zapis dužine definisan kao

$$L_{kn} = L_{k, k+1, \dots, n} = \sum_{i=k}^n L_i, \quad (3.2.2)$$

a vrednosti su, prema slici 2.4a, date u tabeli 3.1.

Ose zavrtnja u $\{s\}$ i $\{b\}$ koordinatnom sistemu date su u tabeli 3.2.



Slika 3.2: Kinematski dijagram Niryo One manipulatora

Redni broj	Prostorni koordinatni sistem \mathcal{S}	Koordinatni sistem prihvavnice \mathcal{B}
1	(0, 0, 1, 0, 0, 0)	(0, 0, 1, 0, L_{57} , 0)
2	(0, -1, 0, L_{12} , 0, 0)	(0, -1, 0, $L_8 - L_{34}$, 0, L_{57})
3	(0, -1, 0, L_{13} , 0, 0)	(0, -1, 0, $L_8 - L_4$, 0, L_{57})
4	(1, 0, 0, 0, L_{14} , 0)	(1, 0, 0, 0, L_8 , 0)
5	(0, -1, 0, L_{14} , 0, - L_{56})	(0, -1, 0, L_8 , 0, L_7)
6	(1, 0, 0, 0, $L_{14} - L_8$, 0)	(1, 0, 0, 0, 0, 0)

Tabela 3.2: Ose zavrtanja Niryo One manipulatora u $\{s\}$ i $\{b\}$ koordinatnom sistemu

Homogene transformacione matrice na osnovu tabele 3.2 date su kao

$$e^{[\mathcal{S}_1]\theta_1} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.3)$$

$$e^{[\mathcal{S}_2]\theta_2} = \begin{bmatrix} c_2 & 0 & -s_2 & L_{12}s_2 \\ 0 & 1 & 0 & 0 \\ s_2 & 0 & c_2 & -L_{12}(1 - c_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.4)$$

$$e^{[\mathcal{S}_3]\theta_3} = \begin{bmatrix} c_3 & 0 & -s_3 & L_{13}s_3 \\ 0 & 1 & 0 & 0 \\ s_3 & 0 & c_3 & -L_{13}(1 - c_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.5)$$

$$e^{[\mathcal{S}_4]\theta_4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_4 & -s_4 & L_{14}s_4 \\ 0 & s_4 & c_4 & L_{14}(1 - c_4) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.6)$$

$$e^{[\mathcal{S}_5]\theta_5} = \begin{bmatrix} c_5 & 0 & -s_5 & L_{14}s_5 - L_{56}(1 - c_5) \\ 0 & 1 & 0 & 0 \\ s_5 & 0 & c_5 & -L_{14}(1 - c_5) - L_{56}s_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.7)$$

$$e^{[\mathcal{S}_6]\theta_6} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_6 & -s_6 & (L_{14} - L_8)s_6 \\ 0 & s_6 & c_6 & (L_{14} - L_8)(1 - c_6) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2.8)$$

Inverznu kinematiku robota odredićemo numerički, a prvi nagađaj odredićemo analitički u prostornim koordinatama. Primetimo da ukoliko izvršimo aproksimaciju i kažemo da je $L_8 \approx 0$, naš manipulator ima sličnu strukturu kao i PUMA robot za koje analitičko rešenje i postupak njegovog nalaženja je dato u knjizi od Mareja, Lia i Sastrija [9]. Prema tome ćemo, analogijom, odrediti približno analitičko rešenje. Krećemo od izraza za direktnu kinematiku prema jednačini 3.1.41, ukoliko je pomnožimo sa desne strane sa \mathbf{M}^{-1} odvojićemo poznate sa jedne i nepoznate transformacije sa druge strane

$$\prod_{i=1}^6 e^{[\mathcal{S}_i]\theta_i} = \mathbf{T}_k \mathbf{M}^{-1} =: \mathbf{T}_1. \quad (3.2.9)$$

gde je

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -L_{57} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_8 - L_{14} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.2.10)$$

Odredićemo generalisane koordinate zglobova u četiri koraka.

Korak 1 (odrediti generalisanu koordinatu θ_3). Pomnožiti obe strane jednačine 3.2.9 sa vektorom koji dodiruje osu \mathcal{S}_4 , \mathcal{S}_5 i \mathcal{S}_6

$$\prod_{i=1}^3 e^{[\mathcal{S}_i]\theta_i} \mathbf{q}_{456} = \mathbf{T}_1 \mathbf{q}_{456} \quad (3.2.11)$$

gde je vektor preseka:

$$\mathbf{q}_{456} = \begin{bmatrix} L_{56} \\ 0 \\ L_{14} \end{bmatrix}. \quad (3.2.12)$$

Jednačina 3.2.11 sledi iz činjenice da ukoliko neki vektor \mathbf{q}_S dodiruje osu S onda važi jednačina

$$e^{[S]\theta} \mathbf{q}_S = \mathbf{q}_S. \quad (3.2.13)$$

Oduzeti sa obe strane jednačine 3.2.11 vektorom preseka ose S_1 i S_2

$$\mathbf{q}_{12} = \begin{bmatrix} 0 \\ 0 \\ L_{12} \end{bmatrix} \quad (3.2.14)$$

koja dodiruje tačku presecanja ose S_1 i S_2

$$e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} \mathbf{q}_{456} - \mathbf{q}_{12} = e^{[S_1]\theta_1} e^{[S_2]\theta_2} (e^{[S_3]\theta_3} \mathbf{q}_{456} - \mathbf{q}_{12}) = \mathbf{T}_1 \mathbf{q}_6 - \mathbf{q}_{12}. \quad (3.2.15)$$

Normiranjem obe strane jednačine 3.2.15 dobija se

$$\|e^{[S_3]\theta_3} \mathbf{q}_{456} - \mathbf{q}_{12}\| = \|\mathbf{T}_1 \mathbf{q}_6 - \mathbf{q}_{12}\| =: \delta \quad (3.2.16)$$

čiji oblik odgovara podproblemu tri, te određujemo θ_3 .

Korak 2 (odrediti generalisane koordinate θ_1 i θ_2). Na osnovu izraza 3.2.11 dobijamo

$$e^{[S_1]\theta_1} e^{[S_2]\theta_2} (e^{[S_3]\theta_3} \mathbf{q}_{456}) = \mathbf{T}_1 \mathbf{q}_{456} \implies e^{[S_1]\theta_1} e^{[S_2]\theta_2} \mathbf{q}^{(1)} = \mathbf{p}^{(1)} \quad (3.2.17)$$

gde je

$$\mathbf{q}^{(1)} := e^{[S_3]\theta_3} \mathbf{q}_{456}, \quad (3.2.18)$$

$$\mathbf{p}^{(1)} := \mathbf{T}_1 \mathbf{q}_{456}, \quad (3.2.19)$$

a koje se rešava prema šablonu podproblema dva za uglove θ_1 i θ_2 .

Korak 3 (odrediti generalisane koordinate θ_4 i θ_5). Prema jednačini 3.2.9 na osnovu naših poznatih izraza

$$e^{[S_4]\theta_4} e^{[S_5]\theta_5} = e^{-[S_3]\theta_3} e^{-[S_2]\theta_2} e^{-[S_1]\theta_1} \mathbf{T}_1 \quad (3.2.20)$$

množenjem obe strane dobijamo jednačinu koja se rešava kao podproblem dva

$$e^{[S_4]\theta_4} e^{[S_5]\theta_5} \mathbf{q}_6 = \mathbf{T}_2 \mathbf{q}_6 =: \mathbf{p}_6 \quad (3.2.21)$$

gde je \mathbf{q}_6 vektor koji ne dodiruje osu S_6 ali nije jednako tački preseka triju osa \mathbf{q}_{456}

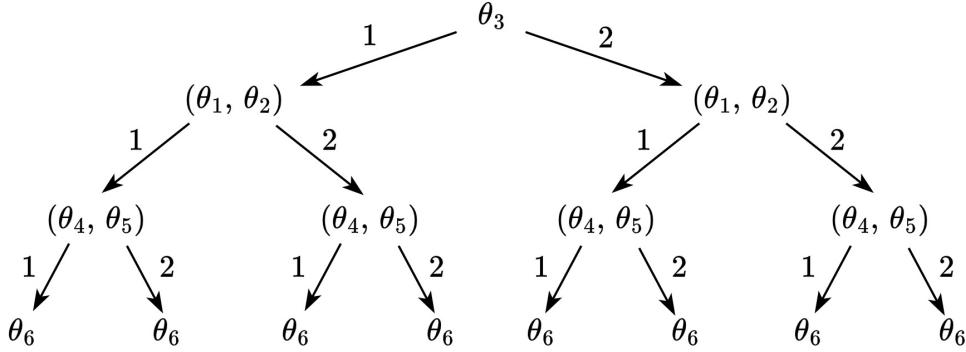
$$\mathbf{q}_6 = \begin{bmatrix} q_6 \\ 0 \\ L_{14} \end{bmatrix}, \quad q_6 \in \mathbb{R} \setminus \{L_{56}\} \quad (3.2.22)$$

i gde je matrica \mathbf{T}_2 ,

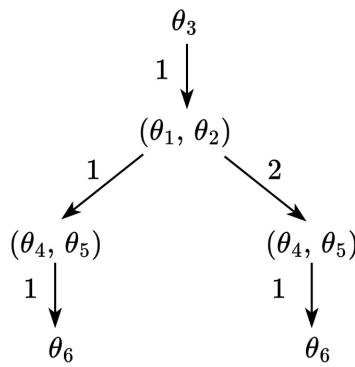
$$\mathbf{T}_2 =: e^{-[S_3]\theta_3} e^{-[S_2]\theta_2} e^{-[S_1]\theta_1} \mathbf{T}_1. \quad (3.2.23)$$

Korak 4 (odrediti generalisanu koordinatu θ_6). Pošto nam ostaje samo jedan nepoznata generalisana koordinata, određujemo je tako što prvo prebacimo poznate na jednu i nepoznate na drugu stranu jednačine 3.2.9

$$e^{[S_6]\theta_6} = \prod_{i=1}^5 \exp(-[S_{6-i}]\theta_{6-i}) \mathbf{T}_1, \quad (3.2.24)$$



Slika 3.3: Graf validnih aproksimativnih rešenja gde su čvorovi grafa generalisane koordinate čije rešenje određujemo, a grane vode ka različitim skupovima rešenja generalisanih koordinata



Slika 3.4: Graf validnih aproksimativnih rešenja u slučaju da θ_3 ima jedno rešenje i iz njega sledi po jedno rešenje za θ_4 i θ_5

zatim množimo obe strane jednačine vektorom koji ne dodiruje osu \mathcal{S}_6

$$e^{[\mathcal{S}_6]\theta_6} \mathbf{q}^{(2)} = \prod_{i=1}^5 \exp(-[\mathcal{S}_{6-i}]\theta_{6-i}) \mathbf{T}_1 =: \mathbf{p}^{(2)} \quad (3.2.25)$$

Na kraju naše analize naglašavamo da je maksimalan broj različitih generalisanih koordinata koje postižu istu željenu konfiguraciju osam zbog višestrukosti rešenja u jednačinama 3.2.16, 3.2.17 i 3.2.21 što se prikazati grafom kao na slici 3.3. U slučaju da svaka generalisana koordinata osim θ_6 ima po dva rešenja onda je graf rešenja isti kao na slici, inače graf rešenja ima manje čvorova i grana. Na primer, ako θ_3 ima jedno rešenje, a θ_4 i θ_5 po jedno rešenje i koje se računa na osnovu jednog rešenja θ_3 , graf bi izgledao kao na slici 3.4.

Pomoću Paden-Kahanovih podproblema, dobijamo između 0 i 8 aproksimativnih rešenja. Ukoliko imamo 0 rešenja, onda je željena konfiguracija van radnog prostora, inače je unutar. Skup rešenja koja se dobijaju Njutn-Rapsonovim algoritmom, pomoću početnih uslova dobijena prema Paden-Kahanovim podproblemima, a da su unutar opsega aktuiranja (tabela 2.2), predstavljaju prava rešenja inverzne kinematike Niryo One manipulatora.

Na osnovu ove glave namenjene mehanici Niryo One manipulatora i mehanici robota, napisana je biblioteka u Pajtonu koja je otvorenog koda i može se naći na Github-u [5].

Glava 4

Eksperiment

Zadatak. Premestiti objekat iz tačke $(148, -148, 0)$ mm u tačku $(148, 148, 0)$ mm pomoću Niryo One manipulatora primenom inverzne kinematike.

Resenje. Rešenje je dano u programskom kôdu 4.1

```
"""Upravljanje Niryo One manipulatorom"""

"""
*** BIBLIOTEKE ***
"""

from mehanika_robota import niryo_one as n_one
import pyniryo as pn
import os
import logging
from msvcrt import getch

"""
*** KONSTANTE ***
"""

NIRYO_ONE_ETHERNET_IP = "169.254.200.200"

POZICIJA1 = n_one.inv_kin(
    [[ 0.0, 0.0, 1.0, 148e-3],
     [ 0.0, 1.0, 0.0, -148e-3],
     [-1.0, 0.0, 0.0, 150e-3],
     [ 0.0, 0.0, 0.0,      1.0]],
    0.1,
    0.1
) [0]

POZICIJA2 = n_one.inv_kin(
    [[ 0.0, 0.0, 1.0, 148e-3],
     [ 0.0, 1.0, 0.0, -148e-3],
     [-1.0, 0.0, 0.0,   88e-3],
     [ 0.0, 0.0, 0.0,      1.0]],
    0.001,
    0.001
) [0]

POZICIJA3 = n_one.inv_kin(
    [[ 0.0, 0.0, 1.0, 148e-3],
     [ 0.0, 1.0, 0.0, 148e-3],
     [-1.0, 0.0, 0.0, 150e-3],
     [ 0.0, 0.0, 0.0,      1.0]],
    0.1,
    0.1
```

```

) [0]

POZICIJA4 = n_one.inv_kin(
    [[ 0.0,  0.0,  1.0,  148e-3],
     [ 0.0,  1.0,  0.0,  148e-3],
     [-1.0,  0.0,  0.0,   88e-3],
     [ 0.0,  0.0,  0.0,    1.0]],
    0.001,
    0.001
) [0]

"""
*** FUNKCIJE ***
"""

def ocisti_terminal():
    """Cisti terminal na COMMAND.COM i cmd.exe CLI
    """
    os.system("cls")

def gasenje_robota(robot: pn.NiryoRobot, izuzetak: Exception = None) -> None:
    """Procedura sigurnog gasenja robota

Parametri
-----
robot : pn.NiryoRobot
    Robot koji treba ugasiti
izuzetak : Exception, opcionalno
    Ukoliko je nastala greska, pribeleziti izuzetak koji je nastao
    (automatska vrednost je None)

Primeri
-----
    >>> robot = pn.NiryoRobot("169.254.200.200")
    >>> gasenje_robota(robot)
    print("Robot je uspesno ugasen")
    >>> robot = pn.NiryoRobot("169.254.200.200")
    >>> gasenje_robota(robot, ValueError)
    print("Doslo je do neocekivane greske, proverite greske.log za detalje")
"""
    ocisti_terminal()

    if izuzetak is not None:
        logging.error(
            f"{robot.get_hardware_status()}\n"
            f"-----\n"
            f"Pozicija zglobova = {robot.get_joints()}\n"
            f"-----\n"
            f"{izuzetak}\n\n"
        )

    robot.set_learning_mode(True)
    robot.close_connection()

    if izuzetak is not None:
        print(
            "Doslo je do neocekivane greske, proverite greske.log za vise "
            "detalja"
        )
    else:
        print("Robot je uspesno ugasen")

def main():

```

```

# Log podešavanje
logging.basicConfig(
    filename      = f"../log/greske.log",
    level        = logging.ERROR,
    format       = "%(asctime)s %(message)s",
    datefmt      = "Date: %d-%m-%Y    Time: %H:%M:%S"
)
# Povezivanje standardnom adresom robota za Ethernet TCP/IP komunikaciju
try:
    robot = pn.NiryoRobot(NIRYO_ONE_ETHERNET_IP)
except Exception as izuzetak:
    logging.error(f"{izuzetak}\n\n")
    print(
        "Nije moguce povezati se sa Niryo One robotom, videti greske.log "
        "za vise detalja"
    )
    return

try:
    # Kalibracija
    robot.calibrate(pn.CalibrateMode.AUTO)
    print("Pritisnite taster [d] da potvrdite rucnu kalibraciju: ")

    while getch().decode() != 'd':
        pass

    robot.calibrate(pn.CalibrateMode.MANUAL)
    print("Rucna kalibracija je uspesna! Molim Vas, udaljite se od robota")
    robot.wait(3)

    # Priblizi se objektu
    robot.set_arm_max_velocity(70)
    robot.move_joints(POZICIJA1)

    # Pokupi objekat
    robot.set_arm_max_velocity(30)
    robot.move_joints(POZICIJA2)
    robot.wait(1)
    robot.close_gripper(hold_torque_percentage=60)
    robot.wait(1)
    robot.move_joints(POZICIJA1)

    # Pomeri se iznad konacne pozicije
    robot.set_arm_max_velocity(70)
    robot.move_joints(POZICIJA3)

    # Ispusti objekat
    robot.set_arm_max_velocity(30)
    robot.move_joints(POZICIJA4)
    robot.wait(1)
    robot.open_gripper()
    robot.wait(1)
    robot.move_joints(POZICIJA3)

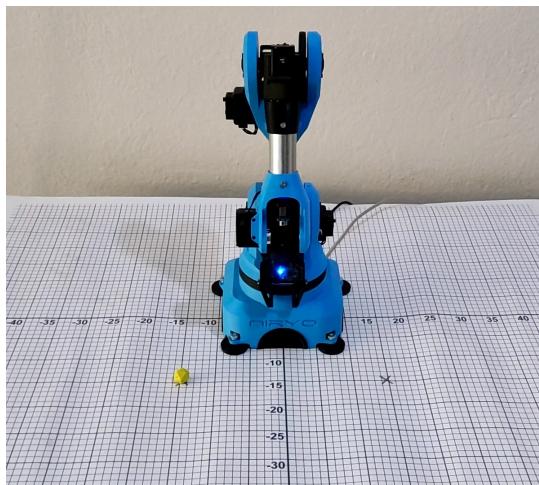
    # Vrati se u pocetnu konfiguraciju
    robot.set_arm_max_velocity(70)
    robot.move_to_home_pose()

except Exception as izuzetak:
    gasenje_robota(robot, izuzetak)
else:
    print("Program je uspesno zavrsen")
    gasenje_robota(robot)

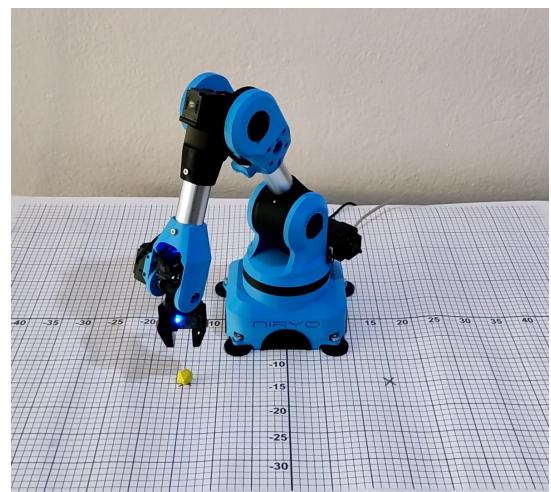
```

```
if __name__ == "__main__":
    ocisti_terminal()
    main()
```

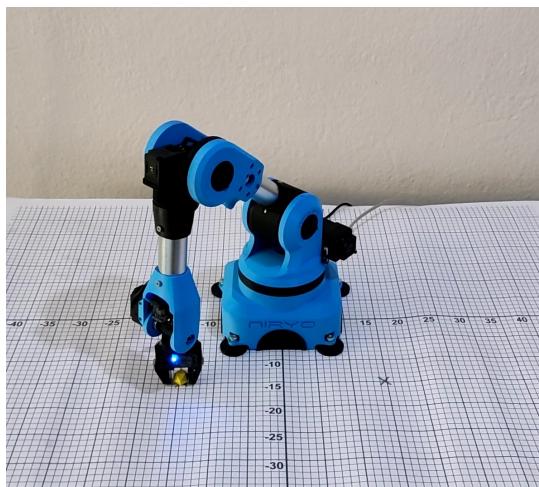
Programski kôd 4.1: Zadatak premeštanja objekta



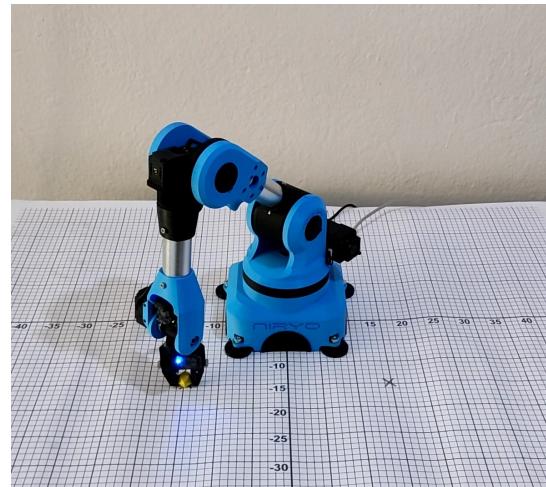
(a) Pozicija 1–početni položaj robota



(b) Pozicija 2–prilazak iznad objekta

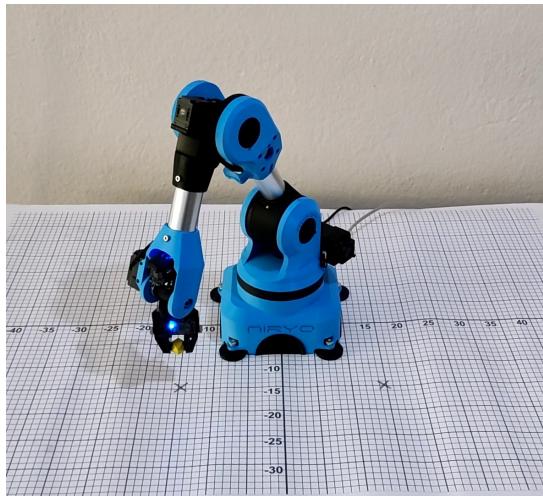


(v) Pozicija 3–spuštanje do objekta

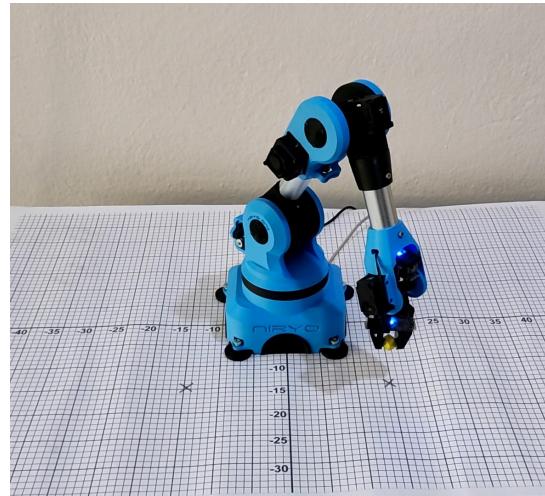


(g) Pozicija 4–hvatanje objekta

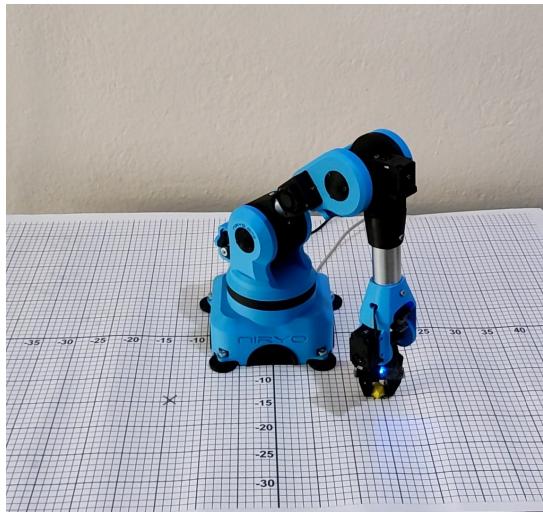
Slika 4.1: Eksperiment upravljanja Niryo One manipulatorom



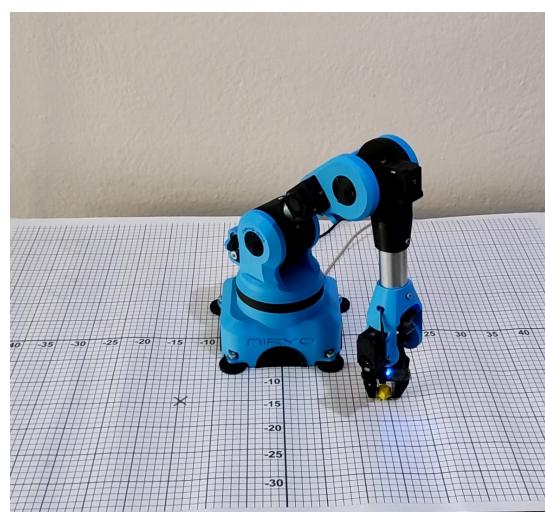
(d) Pozicija 5–podizanje objekta



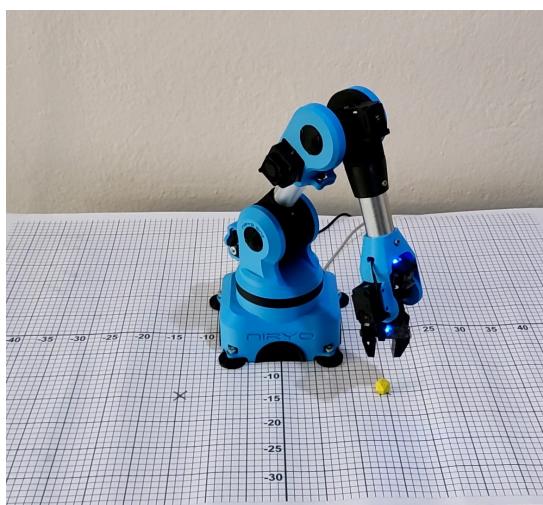
(d) Pozicija 6–prilazak iznad odredišta



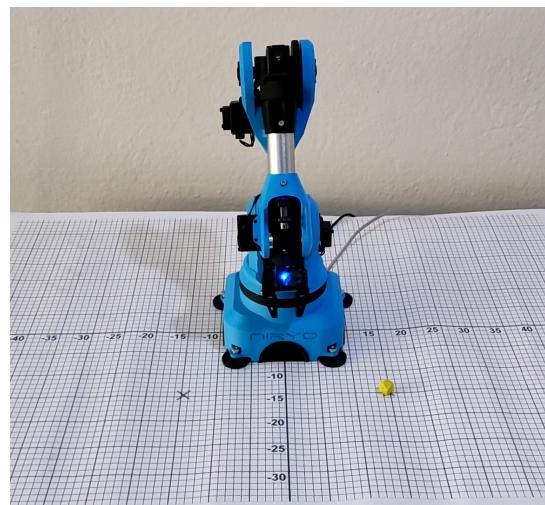
(e) Pozicija 7–spuštanje objekta do odredišta



(ž) Pozicija 8–ispuštanje objekta



(z) Pozicija 9–podizanje od objekta



(i) Pozicija 10–početni položaj robota

Slika 4.1 (nastavak): Eksperiment upravljanja Niryo One manipulatorom

Zaključak

Integracija različitih sinergijskih elemenata robotskog sistema na visokom teorijskom i praktičnom nivou može služiti kao dobar primer je inženjerskog duha, što je upravo izvedeno u radu. Efikasan proračun, jasna formulacija, laka implementacija, nedostatak singulariteta reprezentacije i eksperimentalno izvođenje na Niryo One robotu su evidentni pokazatelji prednosti primene teorije zavrtnjeva u upravljanju manipulatorima.

Na osnovu prethodno izloženog i ličnom sudu, želeo bih ukazati na put koji inspirisan čitalac može pratiti kao prirodan naredni korak nakon pročitanog rada. Proširivanje teorijskog i praktičnog aspekta rada je uvek dobar izbor. Teorija zavrtnjeva poseduje dalekosežne mogućnosti sama po sebi čak i ako se ne uzme u obzir da je zasnovana na linearnoj algebru koja je matematički gigant u tehniči. Pajton je dobar izbor za brzu i relativno prostiju implementaciju koja se može doraditi u prilogu rada na vebajtu [5] kako bi izvorni kôd poprimio još kompletnejji, sigurniji i efikasniji oblik, a da je u skladu sa duhom Pajtona. Pored toga, implementacija u *C++* programskom jeziku bi bilo idealno za oblik kôda koji bi više odgovarao modernim zahtevima praktične robotike. Pre toga, naravno, moguće je i zalaženje dublje u mogućnosti ROS-a kako bi se potpunije iskoristio potencijal Niryo One robota, ukoliko čitalac želi da se fokusira na ovaj robot.

Inovacija ne sme da stane! Svaka kritika i pokušaj izmene ili upotpunjavanja priloženog rada je dobrodošla.

Literatura

- [1] *Ascending Technologies letelica AscTec Falcon 8.* URL: <https://linuxgizmos.com/linux-based-hexacopter-avoids-obstacles%20-using-intel-realsense/>.
- [2] *ClearPath Robotics prizemni roboti Haski i Jackal.* URL: [https://www.theglobeandmail.com/business/article-waterloos%20-clearpath-robotics-sold-to-rockwell-automation-for-us600/..](https://www.theglobeandmail.com/business/article-waterloos%20-clearpath-robotics-sold-to-rockwell-automation-for-us600/)
- [3] Vlatko Doleček i Isak Karabegović. *Robotika.* Bihać: Tehnički fakultet, 2002.
- [4] *Robots and robotic devices–Vocabulary.* Standard. Geneva, CH: International Organization for Standardization, 2012.
- [5] *Izvorni kôd mehanike robota.* URL: https://github.com/VuckoT/mehanika_robota.
- [6] *Kickstarter Niryo One projekat.* URL: <https://www.kickstarter.com/projects/niryo/niryo-one-an-open%20-source-6-axis-robotic-arm-just-f>.
- [7] Kevin M. Lynch i Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control.* Cambridge University Press, 2017.
- [8] *Modern Robotics Pajton biblioteka.* URL: <https://github.com/NxRLab/ModernRobotics/tree/master/packages/Python>.
- [9] Richard M. Murray, Zexiang Li i S.Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.
- [10] *Niryo One vebsajt.* URL: <https://niryo.com/>.
- [11] *Niryo Robotics na GitHub-u.* URL: <https://github.com/NiryoRobotics>.
- [12] *pyniryo biblioteka od Niryo Robotics na GitHub-u.* URL: <https://github.com/NiryoRobotics/pyniryo>.
- [13] Niryo Robotics. *Niryo One Mechanical Specifications.* 2018.
- [14] *Robotics Casual.* URL: <https://roboticscasual.com/ros-tutorial-how-to-use-opencv-in-a%20-robot-pick-and-place-task-for-computer-vision/>.
- [15] *ROS vebsajt.* URL: <https://wiki.ros.org>.
- [16] Daniel Serrano. *Introduction to ROS (Robot Operating System).* STO-EN-SCI-271, 2015.
- [17] *Willow Garage čovekoliki robot PR2.* URL: <https://www.wired.com/2010/05/robot-graduation-party/>.