

Univerza v Ljubljani

Fakulteta za elektrotehniko

Timotej Gašpar

# **Vodenje robota v stiku s podajnim objektom**

Magistrsko delo

Mentor: prof. dr. Matjaž Mihelj

Somentor: dr. Leon Žlajpah

Ljubljana, 2015



## Zahvala



# Vsebina

<b>1</b>	<b>Uvod</b>	<b>5</b>
<b>2</b>	<b>Robotski manipulator PA-10</b>	<b>7</b>
2.1	Kinematični model robota PA-10 . . . . .	7
2.2	Servo Krmilnik . . . . .	10
2.2.1	Komunikacija z ARCNET vmesnikom . . . . .	10
2.3	Senzor sile in navorov - JR3 . . . . .	12
<b>3</b>	<b>UDP strežnik za komunikacijo s krmilnikom robota ter senzor-jem JR3</b>	<b>15</b>
3.1	ARCNET strežnik . . . . .	15
3.2	JR3 strežnik . . . . .	16
3.3	Združitev ARCNET in JR3 strežnika . . . . .	17
3.4	Razlogi za strežnik . . . . .	17
<b>4</b>	<b>Kratek uvod v redundatne mehanizme in pseudoinverz Jacobi-jeve matrike</b>	<b>19</b>
<b>5</b>	<b>Admitančno krmiljenje</b>	<b>23</b>
5.1	Vodenje preko inverzne kinematike . . . . .	23

---

5.1.1	Vodenje v notranjih koordinatah . . . . .	23
5.1.2	Vodenje v zunajih koordinatah . . . . .	24
5.1.3	Upraba zunanjih sil . . . . .	25
5.2	Implementacija . . . . .	26
5.3	Rezultati . . . . .	27
<b>6</b>	<b>Krmiljenje z inverzno dinamiko</b>	<b>29</b>
6.1	Teorija . . . . .	29
6.1.1	Vodenje v notranjih koordinatah . . . . .	30
6.2	Implementacija . . . . .	31
6.3	Rezultati . . . . .	32
<b>7</b>	<b>Simulink knjižice</b>	<b>33</b>
7.1	Blok PA-10 . . . . .	33
7.2	Blok Inverse Dynamics Control . . . . .	33
7.3	Blok Follow Trajectory . . . . .	34
7.4	Blok Admittance Control . . . . .	34
<b>8</b>	<b>Zaključek</b>	<b>35</b>
<b>A</b>	<b>Appendix 1</b>	<b>41</b>
<b>B</b>	<b>Appendix 2</b>	<b>43</b>
<b>C</b>	<b>Appendix 3</b>	<b>45</b>
<b>D</b>	<b>Predloge za navajanje literature - baza BibTex</b>	<b>47</b>

## Seznam slik

2.1	Skica robotskega mehanizma PA 10 . . . . .	9
2.2	Način za krmiljenje robota . . . . .	12
5.1	Povratna zanka za krmiljenje kota v sklepu . . . . .	24
5.2	Bločna shema krmiljenja referenčne lege vrha robota . . . . .	25
6.1	Bločna shema krmiljenja referenčne lege vrha robota z inverzno dynamiko . . . . .	31





## Seznam tabel

1	Veličine in simboli . . . . .	xi
2.1	D-H parametri . . . . .	9



## Seznam uporabljenih simbolov

V pričujočem zaključnem delu so uporabljeni naslednje veličine in simboli:

Veličina / oznaka		Enota	
Ime	Simbol	Ime	Simbol
čas	$t$	sekunda	s
frekvenca	$f$	Hertz	Hz
sila	$F$	Newton	N
masa telesa	$m_t$	kilogram	kg
kot v sklepu	$q$	radijan na sekundo	rad/s
lega vrha robota	$\mathbf{x}$	metri	m
Jacobijeva matrika	$\mathbf{J}$	-	-

Tabela 1: Veličine in simboli

Pri čemer so vektorji in matrike napisani s poudarjeno pisavo. Natančnejši pomen simbolov in njihovih indeksov je razviden iz ustreznih slik ali pa je pojasnjen v spremljajočem besedilu, kjer je simbol uporabljen.



## Povzetek

**Ključne besede:** beseda1, beseda2, beseda3



## **Abstract**

The thesis addresses ...

**Key words:** word1, word2, word3





# 1 Uvod



## 2 Robotski manipulator PA-10

Podjetje Mitsubishi Heavy Industries je leta 1992 izdelalo prvega industrijskega redundatnega robota [1]. Podjetje je robota poimenovalo Portable General-purpose Intelligent Arm PA-10, krajše PA-10. Gre za serijskega robota s sedmimi stopnjami prostosti. Prvi trije sklepi so označeni kot ramenski sklepi (shoulder), S1, S2, S3. Naslednja dva sta označena kot komolčni sklepi (elbow), E1, E2. Zadnja dva sklepa pa sta označena kot zapestna (wrist), W1, W2. Glede na zgradbo se ga lahko uvrsti v tako imenovane antropomorfne robote [2]. Izraz antropomorfnost izhaja iz dveh grških besed *antropo* - človek, *morphe* - oblika. Značilnost takih robotov je spretnost saj so vsi sklepi rotacijski [3].

Robotska roka PA-10 tehta 36 kg in ima nosilnost 10 kg. Servo motorji v sklepih se napajajo preko izmenične napetosti. Prenosi med v sklepih so realizirani z mehanskim sistemom harmonic drive. Proizvajalec navaja, da se lahko bazo robota pritrdi v katerokoli lego. To pomeni, da se ga lahko fiksira bodisi na tla, na steno ali na strop. Robota PA-10 odlikuje relativno majhna konstrukcija, enostavno rokovanje (menjava konfiguracije) ter odprtost njegovega krmilnika. Prav ti razlogi so povod, da je mnogo institucij vzelo tega robota kot eksperimentalni sistem za razvijanje raznih algoritmov vodenja oz. identificiranja ([4], [5], [6], [7]).

### 2.1 Kinematični model robota PA-10

Robotski mehanizem se obravnava kot kinematično verigo  $n$  med seboj povezanih togih teles. Ker je en konec kinematične verige povezan v bazo robota je mogoče

reči, da se premika le vrh robota. Z opisom kinematične relacije med dvema zaporednima segmentoma je mogoče to znanje združiti za definiranje kinematične relacije med bazo in vrhom robota. Homogena transformacijska matrika med dvema sosednjima segmentoma je definirana kot:

$$\mathbf{T}_{i+1}^i = \begin{bmatrix} \cos(\Theta_i) & -\sin(\Theta_i) & 0 & a_i \\ \sin(\Theta_i)\cos(\alpha_i) & \cos(\Theta_i)\cos(\alpha_i) & -\sin(a_i) & -\sin(a_i)d_i \\ \sin(\Theta_i)\sin(\alpha_i) & \cos(\Theta_i)\sin(\alpha_i) & \cos(a_i) & \cos(a_i)d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

V homogeni transformacijski matriki  $\mathbf{T}_{i+1}$  so  $a_i$ ,  $\alpha_i$ ,  $d_i$  in  $\Theta$  parametri tako imenovane Denavit - Hartenberg sistema opisovanja relacij med različnimi koordinatnimi sistemi. Sklicujoč na sliko ?? se opiše parametre kot:

- $a_i$  razdalja med  $O_i$  in  $O_{i+1}$  v smeri  $x_i$ ,
- $\alpha_i$  kot med  $O_i$  in  $O_{i+1}$  glede na os  $x_i$ ,
- $d_i$  razdalja med  $O_{i-1}$  in  $O_i$  glede na os  $z_i$ ,
- $\Theta$  razdalja med  $O_{i-1}$  in  $O_i$  glede na os  $z_i$ ,

Zgornje parametre se d

Tako je mogoče zapisati homogeno transformacijo od baze do vrha robota kot produkt posameznih transformacij

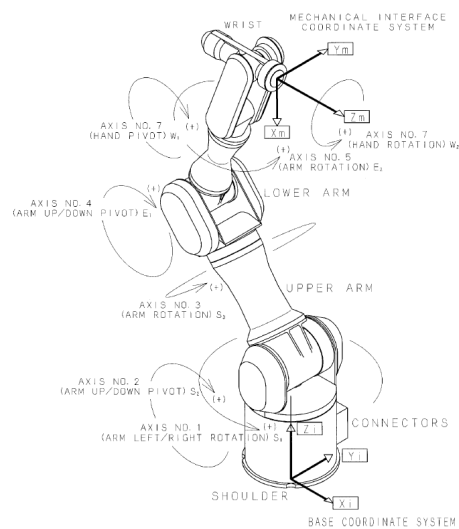
$$T_7^0 = T_1^0 \cdot T_2^1 \cdot \dots \cdot T_7^6. \quad (2.2)$$

S pomočjo D-H parametrov lje moč definirati geometrijski model robota oz. transformacijsko matriko  $A_{PA10} = A_1 \cdot A_2 \cdot \dots \cdot A_7$ , ki preslika koordinate izražene v koordinatnem sistemu baze ( $x_{base}$ ) koordinate izražene v koordinatnem sistemu na vrhu robota ( $x_{PA10}$ ) v odvisnosti od kotov ( $q$ ) v sklepih. Tako velja relacija

$$x_{PA10} = A_{PA10}(q) \cdot x_{base}$$

$i$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	0.315	$q1$
2	$-\frac{\pi}{2}$	0	0	$q2$
3	$\frac{\pi}{2}$	0	0.45	$q3$
4	$-\frac{\pi}{2}$	0	0	$q4$
5	$\frac{\pi}{2}$	0	0.5	$q5$
6	$-\frac{\pi}{2}$	0	0	$q6$
7	$\frac{\pi}{2}$	0	0.08	$q7$

Tabela 2.1: D-H parametri



Slika 2.1: Skica robotskega mehanizma PA 10

## 2.2 Servo Krmilnik

Proizvajalec ob robotskem mehanizmu dostavi tudi krmilnik. V krmilniku so štiri ločeni krmilniki za servo motorje. Vsak servo krmilnik krmili servo motorja, razen enega. Krmilnik omogoča vodenje na dva načina. Prvi način predvideva, da motorje krmilimo navorno, drugi pa hitrostno. Razlika je tudi v regulatorjih. Navorno krmiljenje je realizirano z analognim P regulatorjem toka, hitrostna regulacija pa je realizirana z digitalnim PI regulatorjem s frekvenco približno 1500 Hz.

Na zadnji strani krmilnika je stikalo s katerim lahko preklopimo med dvema načina delovanja: "Teach Mode" ter "Run mode". Ko je krmilnik nastavljen na "Teach Mode", so hitrosti v sklepih omejene tako, da se lahko vrh robota premika s maksimalno hitrostjo 250 mm/s.

Ob vžigu krmilnika se parametri za vodenje robota naložijo iz EEPROM tabele v RAM. Med temi parametri so tudi ojačanje proporcionalnega ter integracijskega dela regulatorja, omejitve posameznih stopenj, razmerje prenosa zobnikov, itd.

Referenčne navore ali hitrosti v sklepih na servo krmilniku se nastavljajo preko zunanjega vmesnika. Krmilnik ima ARCNET vmesnik, ki podpira serijsko komunikacijo po ARCNET protokolu. Krmilnik ima ločen konektor za vhod (Rx) ter izhod (Tx). Uporabljena so optična vodila. Za komunikacijo s servo krmilnikom je bil v namen tega dela narejen vmesnik, ki je podrobneje opisan v poglavju 3.

### 2.2.1 Komunikacija z ARCNET vmesnikom

ArcNET vmesnik na krmilniku skrbi za komunikacijo med servo krmilniki ter visokonivojskim vmesniku. Dovoli hitrost komunikacije do 5 Mb/s. Njegova naloga je interpretirati dobljene pakete, primerno nastaviti parametre na servo krmilniku ter posredovati primeren odgovor višjenivojskemu vmesniku. Ko višjenivojski vmesnik ne pošilja nobenih ukazov je ARCNET vmesnik v čakanju. V tem stanju

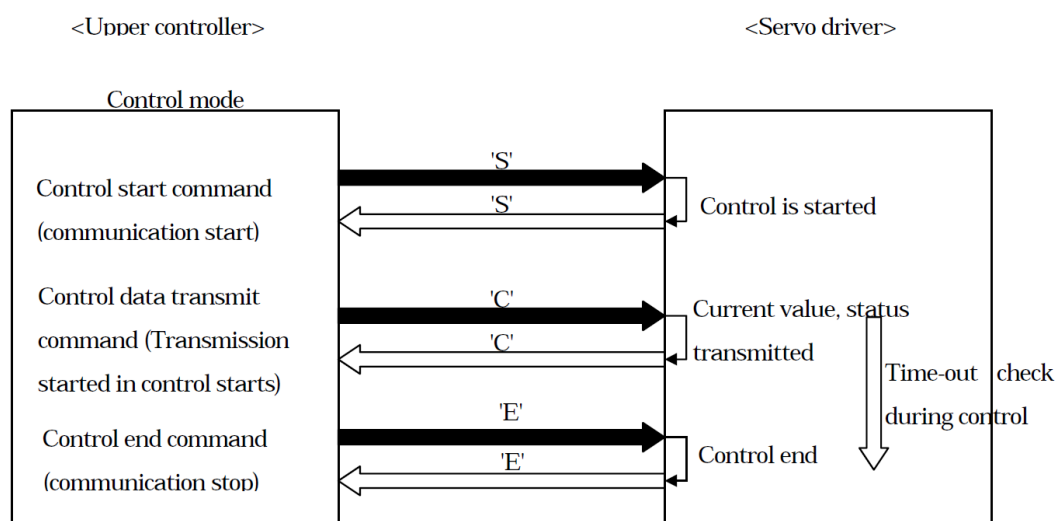
čaka na primerno sestavljen paket. Vsak paket se začne z veliko tiskano črko zapisano v 8-bitnem ASCII formatu. Z različnimi črkami lahko izberemo različna stanja oz. načine delovanja ARCNET vmesnika:

- Izpis EEPROM tabele
- Vpis v EEPROM tabelo
- Vpis v RAM tabelo
- Kopiranje iz RAM v EEPROM tabelo
- Postavitev kotov na ničelno vrednost
- Sporstitev zavor
- Vodenje sklepov

Tako lahko uporabnik poljubno spremeni razne parametre, ki se uporabljajo za vodenje robota. Možno je na primer spremeniti ojačanja PI regeulatorja, vendar se teh funkcionalosti v tem delu ni podrobneje raziskalo saj ni bilo potrebe po tem.

V tem delu se je v večini uporabljalo način za vodenje sklepov. Ta način se izbere tako, da se ARCNET vmesniku najprej pošlje ukaz, ki se začne z velikim tiskanim S (ASCII DEC 65), nadaljuje se s pošiljanjem paketov, ki se začnejo s veliko tiskano črko C (ASCII DEC 67) ter zaključi s tem, da se pošlje paket, ki se začne z velikim tiskanim E (ASCII 69). Pri tem načinu delovanja je potrebno posvetiti nekaj pozornosti frekvenci pošiljanja. V kolikor paketi ne pridejo do ARCNET vmesnika v intervalu specificiranem na krmilniku, bo ARCNET vmesnik šel iz tega načina delovanja nazaj na način čakanje.

Paket, ki se začne s C in nosi podatke o krmiljenju sklepov robota je velik 35 bajtov, 5 za vsak sklep. Prvi bajt je kontrolni in samo prvi trije biti nosijo podatke o vodenju:



Slika 2.2: Način za krmiljenje robota

1. vklop ali izklop mehanske zavore (1 - vklop, 0 - izklop)
2. vklop ali izklop servo motorja (1 - vklop, 0 - izklop)
3. izbira navornega ali hitrostnega načina (1 - navorni, 0 - hitrostni)

Drugi in tretji bajt nosijo podatek o referenčnem navoru na sklepu. Ta podatek bo uporabljen le, če je servo motor nastavljen na navorni način delovanja. Dvobajtni podatek pa je v formatu 0.001 Nm/digit. Četrty in peti bajt nosijo podatek o referenčni hitrosti na sklepu. Podatek je tako kot pri navornem delovanju v uporabi le, ko vodimo servo motor na hitrostnem načinu. Dvobajtni podatek je v formatu 0.0002 rad/s/digit.

## 2.3 Senzor sile in navorov - JR3

Za namene vodenja opisanega v poglavju 5 je med vrhom robota ter prijemalom pritrnilo senzor sil JR3. Senzor nam omogoča merjenje sil in navorov v X, Y in Z oseh. Senzor je narejen iz uporovnih lističev porazdeljenih po notranjosti. Iz specifikacij senzorja [8] je razvidno, da uporablja 8 uporovnih lističev. S tem, ko



na senzor deluje zunanja sila ali navor, se upornost na uporovnih lističih spremeni proporcionalno s silo.

Senzor sile JR3 ima v svoji notranjosti poleg uporovnih lističev še AD-pretvornik tako, da iz senzorja pridejo podatki o navoru v digitalnem formatu. Senzor se lahko ž priključi preko 6 ali 8 žičnega kabla na osebni računalnik, na katerem je kartica, ki skrbi za digitalno obdelavo signala. Kartice na osebni računalniku so vključene bodisi na ISA ali PCI vodilo, odvisno od izvedbe. Za potrebe tega dela se je uporabljala kartica na ISA vodilu. Na kartici je vezje za digitalno procesiranje signalov. V dokumentaciji kartice [9] so opisani trije nizkoprepustni filtri z različnimi mejnimi frekvencami. V tem delu se je uporabilo filter z mejno frekvenco 500 Hz.



## **3 UDP strežnik za komunikacijo s krmilnikom robota ter senzorjem JR3**

V prejšnjem poglavju so bile opisane mehanske (robotski mehanizem PA-10 in krmilnik mehanizma) in senzorne (senzor sile JR3) komponente, ki se jih je za to delo uporabljalo. Vendar pa ni nobena od uporabljenih mehanskih ali senzornih komponent imela delujočega programskega dela. Za potrebe izvedbe tega dela je tako bila narejena primerna programska oprema, da je omogočala uporabo mehanske ter senzorne opreme. Nastal strežnik za komunikacijo s krmilnikom robota po ARCNET mreži. Sledil je program za prebiranje vrednosti izmerjenih sil ter navorov iz senzorja JR3.

V naslednjih poglavjih se bo podrobneje opisalo delovanje obeh programov ter končni izdelek, ki predstavlja UDP strežnik. Vsa programska oprema, ki bo opisana je bila zgrajena v programskem jeziku C ter na operacijskem sistemu CentOS Linux z jedrom Kernel 2.6. Razlog za odločitev, ki je privedla do izbire omenjenega operacijskega sistema, je obstoj dokumentacije [10] za točno ta sistem, ki je bila najdena na internetu in opisuje uporabo ARCNET kartic.

### **3.1 ARCNET strežnik**

V poglavju 2.2.1 je opisan ARCNET vmesnik na krmilniku robota. Opisana so stanja vmesnika in kakšne so zahteve za preklapljanje med njimi. V tem poglavju bo opisano vse potrebno za vzpostaviti komunikacijo z krmilnikom robota po

ARCNET protokolu.

Uporabnik, ki bi želel uporabljati PCI kartico na računalniku za komunikacijo po ARCNET omrežju mora najprej zagotoviti ustrezne gonilnike. Tej so na razpolago v [10]. Gonilnik se doda v Linux okolju kot modul. Po uspešnem zagonu gonilnika lahko uporabnik vzpostavi ARCNET mrežo s standardno Linux komando `ifconfig` in sicer na sledeč način:

```
ifconfig arc0 192.168.1.1
```

Od tukaj dalje lahko uporabnik razvija programsko opremo. V primeru tega dela so avtorji uporabljali programski jezik C. Avtorji tega dela so od tega koraka dalje začeli razvijati strežnik za posredovanje ukazov pridobljenih iz UDP omrežja na ARCNET omrežje. UDP omrežje se je uprabilo, ker lahko omogoča visoke hitrosti prenosa podatkov, hkrati pa omogoča stabilnost tudi v primeru izguba paketov. Avtorji tega dela so razvili strežnik, ki deluje s frekvenco 500 Hz. To se zagotavlja z onemogočanjem uporabe namizja ter uporabo Linux C funkcije `clock_gettime(clockid_t clk_id, struct timespec *tp);`. Funkcija vrne vrednost systemskega časa. Z večkratnim klicom te funkcije lahko vidimo kolko systemskega časa je minilo od enega do drugega klica. S tem lahko zagotovimo, da program deluje z neko konstantno frekvenco in zelo majhnim odstopanjem od nje.

## 3.2 JR3 strežnik

Senzor sil in navorov JR3 ima se lahko povezuje z osebnim računalnikom preko kartice, ki se priključi na vodilo ISA (starejši standard) ter kartice, ki se priključi na vodilo PCI (novejši standard). Na uporabljenem računalniku za potrebe tega dela je nameščena kartica na vodilu ISA. Ker je standard starejši je podjetje prenehalo s podporo tega standarda. Kot posledica tega so avtorji tega dela razvili svojo programsko opremo za komunikacijo z notranjimi registri kartice. Osnovne informacije o podatkih do katerih lahko dostopamo po registrih se nahajajo v

dokumentaciji [9], ki jo priloži proizvajalec. Dokumentacija navaja, da v kolikor želi uporabnik odčitati vrednost sile ali navora  $F_x$  mora tako prebrati osnovno vrednost  $F_s$  na registru imenovanem (`full scale`), jo pomnožiti z vrednostjo  $F_f$ , ki je na registru izbranega filtra (`current force`). Na zadnje more to vrednost delit s konstanto  $K_p$ , da pretvori v prave enote (N ali Nm).

$$F_x = \frac{F_s \cdot F_f}{K_p} \quad (3.1)$$

Ker senzor meri silo in navor v treh smereh je potrebno opraviti izračun za vsako dimenzijo posebej.

### 3.3 Združitev ARCNET in JR3 strežnika

### 3.4 Razlogi za strežnik



## 4 Kratek uvod v redundatne mehanizme in pseudoinverz Jacobijeve matrike

V poglavju ?? je opisan robotski mehanizem PA-10 in omenjeno je tudi, da ima sedem sklepov. Večina industrijskih robotov jih ima običajno šest. Število sklepov bo v nadaljevanju označeno s  $n$ . V kolikor je naloga robota pozicioniranje vrha v neko lego je govora o šest stopnjah prostosti, tri za pozicijo in tri za orientacijo. Stopnje prostosti naloge bodo označene z  $m$ . Če bi veljalo  $n = m$  bi število rešitev inverzne kinematike bilo končno. V kolikor velja  $n > m$  je mogoče reči, da je mehanizem redundanten za dano nalogo in je število rešitev inverzne kinematike neskončno.

Smiselno je toraj boljše definirati nalogo. Ena opcija je definiranje primarne ( $\mathbf{p}$ ) in sekundarne ( $\mathbf{s}$ ) naloge. Z dodatkom naloge je mogoče koristno izkoristiti kinematično redundanco robota. Vprašanje je le, kako definirati sekundarno nalogo. V primeru tega dela, je bila definirana sekundarna naloga tako, da je minimizirala premike v sklepih.

Naj bodo premiki za izvedbo primarne naloge označeni kot  $d\mathbf{q}_p$ , sekundarni pa  $d\mathbf{q}_s$ . Tako se lahko zapiše premik za izvedbo obeh nalog kot

$$d\mathbf{q} = d\mathbf{q}_p + d\mathbf{q}_s. \quad (4.1)$$

Za izračun  $d\mathbf{q}_p$  in  $d\mathbf{q}_s$  je potrebno rešiti problem inverzne kinematike. Izračun inverzne kinematike veleva poznanje inverza Jacobijeve matrike. Vendar pa je v

primeru, ko velja  $n > m$  Jacobijeva matrika nesimetrična. Posledično ni mogoče izračunati njenega inverza. Mogoče pa je definirati pseudoinverzno Jacobijevo matriko po sledeči enačbi:

$$\mathbf{J}^* = (\mathbf{J}^T \cdot \mathbf{J})^{-1} \cdot \mathbf{J}^T. \quad (4.2)$$

Pseudoinverz primarne naloge bo označen z  $\mathbf{J}_p^*$ , inverz sekundarne pa  $\mathbf{J}_s^*$ . Smiselno je tudi definirati neke uteži tako, da so kvadrati hitrosti najmanjši [11]. Definira se neizrojeno pozitivno definitno matriko uteži  $\mathbf{A}$ . Zapiše se utežen pseudoinverz primarne naloge kot

$$\mathbf{J}_{pA}^* = \mathbf{A}^{-1} \cdot \mathbf{J}_p^T \cdot (\mathbf{J}_p \cdot \mathbf{A}^{-1} \cdot \mathbf{J}_p^T)^{-1}. \quad (4.3)$$

Sedaj velja enakost

$$d\mathbf{q} = \mathbf{J}_{pA}^* \cdot d\mathbf{p} + \mathbf{J}_s^* \cdot d\mathbf{s}, \quad (4.4)$$

ki pa ima eno pomankljivost. Sekundarna naloga ni utežena. Želja je, da sekundarna naloga vpliva na primarno le toliko, kolikor želimo. Definira se tako matriko, ki preslika rezultat množenja  $\mathbf{J}_s^* \cdot d\mathbf{s}$  v ničelni prostor matrike  $\mathbf{J}_p$ . Definira se matrika  $\mathbf{N}_{pA}$  na sledeč nač način [11]:

$$\mathbf{N}_{pA} = \mathbf{I} - \mathbf{J}_{pA}^* \cdot \mathbf{J}_p \quad (4.5)$$

Z množenjem sekundarnega dela enačbe 4.4 z matriko  $\mathbf{N}_{pA}^*$  dobimo tako končno enačbo za izračun premikov v notranjih koordinatah z upoštevanjem primarne in sekundarne naloge.

$$d\mathbf{q} = \mathbf{J}_{pA}^* \cdot d\mathbf{p} + \mathbf{N}_{pA} \cdot \mathbf{J}_s^* \cdot d\mathbf{s}, \quad (4.6)$$

V poglavjih, ki sledijo se bo predpostavljalo, da je sekundarna naloga neupoštevana in kadarkoli bo zapisana Jacobijeva matrika se bo predpostavilo, da



gre za pseudoinverz te matrike.



## 5 Admitančno krmiljenje

Robotski mehanizem PA-10 je industrijski robot s segmenti iz pretežno litega železa. Segmenti imajo relativno veliko maso, sklepi pa relativno visoko trenje. Lahko predpostavimo, da ima manipulator veliko lastno impendanco. Avtor v [12] za take mehanizme predlaga admitančno vodenje. To pomeni, da se silo, ki deluje na vrh robota meri s senzorjem sil. To bi lahko naredili tudi posredno z merjenjem navorov v sklepih vendar bo o tem več v poglavju 6.

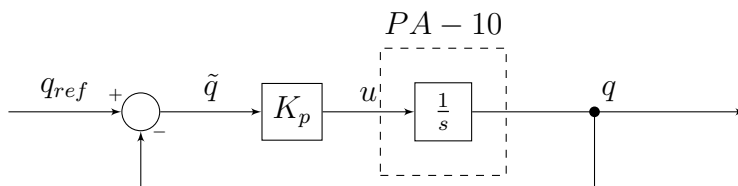
V nadaljevanju bo opisana potrebna teorija za admitančno vodenje mehanizmov. Začelo se bo z vodenjem pozicije v notranjih koordinatah, prešlo na vodenje pozicije v zunanjih koordinatah nato pa še opisalo admitančno vodenje.

### 5.1 Vodenje preko inverzne kinematike

#### 5.1.1 Vodenje v notranjih koordinatah

V poglavju 2.2 je opisano, da ima servo krmilnik proporcionalno diferencirni regulator za regulacijo hitrosti v sklepih. Za namen poenostavitve bo robotski mehanizem aproksimiran kot integrator  $\frac{1}{s}$ . Spomniti je potrebno, da krmilnik robota prejme kod vhodne veličine hitrosti v sklepih ( $\omega$ ), vrne pa kote v sklepih ( $\mathbf{q}$ ). Iz tega sledi enostavna regulacijska enačba:

$$\mathbf{u} = \mathbf{K}_p \cdot \tilde{\mathbf{q}} \quad (5.1)$$



Slika 5.1: Povratna zanka za krmiljenje kota v sklepu

$$\tilde{\mathbf{q}} = \mathbf{q}_{ref} - \mathbf{q} \quad (5.2)$$

Slika 5.1 prikazuje enostavno regulacijsko shemo, ki je uporabljena za krmiliti robota v notranjih koordinatah. Čas, ki ga motor porabi, da pride iz začetne v referenčno lego je proporcionalen ojačanju  $K_p$ .

### 5.1.2 Vodenje v zunajih koordinatah

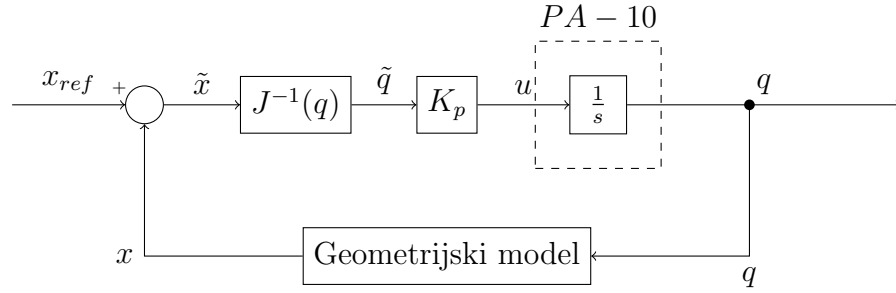
Iz tega znanja je sedaj moč realizirati regulacijo v zunanjih koordinatah. Najprej opisati napako vrha robota. Naj bo  $x_{ref}$  željena lega vrha robta in naj bo  $x$  trenutna lega. Iz tega sledi enačba napake:

$$\tilde{\mathbf{x}} = \mathbf{x}_{ref} - \mathbf{x} \quad (5.3)$$

Vendar pa je potrebno pred tem poznati kinematični model robota. V poglavju ?? so bili opisani D-H parametri robota. Avtorji v [13] opišejo kako zapisati kinematiko robota. Potrebno je zapisati Jacobijevo matriko  $J$ . Z Jacobijevo matriko lahko zapišemo relacijo med spremembo zunanjih koordinatah (vrh robota) ter spremembo notranjimi (koti v sklepih):

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}} \quad (5.4)$$

Ker je pri Jacobijevi matriki govora o relaciji med spremembami, se lahko kot spremembo tudi uporabi napako. Z uporabo enačb 5.1, 5.3 in 5.4 je moč zapisati enostaven proporcionalni regulator pozicije vrha robota.



Slika 5.2: Bločna shema krmiljenja referenčne lege vrha robota

Z množenjem enačbe 5.4 z inverzom Jacobijeve matrike se lahko izpostavi  $\tilde{q}$ :

$$\tilde{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \cdot \tilde{\mathbf{x}} \quad (5.5)$$

Sedaj je mogoče regulirni signal, ki krmili hitrost v sklepih robota tako, da bo vrh robota dosegel referenčno lego:

$$\mathbf{u} = \mathbf{K}_p \cdot \mathbf{J}^{-1}(\mathbf{q}) \cdot (\mathbf{x}_{ref} - \mathbf{x}) \quad (5.6)$$

Avtor v [14] navaja, da se mehanski sistem, ki je voden na ta način obnaša kot mehanski sistem z  $n$  - dimenzionalno vzmetjo v notranjih koordinatah. Togost omenjene vzmeti pa določa ojačanje  $K_p$ .

### 5.1.3 Upraba zunanjih sil

Za doseči admitančno vodenje je na tem mestu potrebno še upoštevati podatek o silah in navorih, ki delujejo na vrhu robota. Kot omenjeno v poglavju 3 je bil na vrh robota pritrjen senzor sil in navorov JR3. Odčitki senzorja sile bodo označeni z črko  $\mathbf{h}_{JR3}$ . V nadaljevanju bo opisano vodenje pozicije vrha robota z upoštevanjem željene sile na vrhu robota.

Naj bo napaka v željeni sili definirana kot:

$$\tilde{\mathbf{h}} = \mathbf{h}_{ref} - \mathbf{h} \quad (5.7)$$

Napako bi bilo mogoče sedaj ojačati in direktno pripeljati v regulator. Vendar pa avtorji v tem delu predlagajo, da se to napako superponira referenčni legi robota. Na ta način se doseže to, da je robot postavljen neko lego in šele tam deluje z neko silo. V nadaljevanju bo predpostavljeno, da je referenčna sila enaka nič. Na ta način se enačba napake sile poenostavi na  $\tilde{\mathbf{h}} = \mathbf{h}_{JR3}$ .

Napako sile prištejemo referenčni pozicijo po tistem, ko jo ojačamo z matriko ojačanj  $\mathbf{K}_h$ . Matrika  $\mathbf{K}_h$  je diagonalna in pove, koliko je vrh robota podajen v posamezne smeri. Novo referenčno pozicijo se bo označilo z  $\mathbf{x}_{ref}^*$ .

$$\mathbf{x}_{ref}^* = \mathbf{x}_{ref} + \mathbf{K}_h \cdot \tilde{\mathbf{h}}_{JR3} \quad (5.8)$$

Z vstavitvijo enačbe 5.8 v 5.6 je mogoče zapisati celotno regulacijsko enačbo za vodenje robota v zunanjih koordinatah z upoštevanjem delovanja zunanjih sil na vrh robota.

$$\mathbf{u} = \mathbf{K}_p \cdot \mathbf{J}^{-1}(\mathbf{q}) \cdot (\mathbf{x}_{ref} + \mathbf{K}_h \cdot \tilde{\mathbf{h}}_{JR3} - \mathbf{x}) \quad (5.9)$$

## 5.2 Implementacija

Avtorji so za implementacijo na robotskem mehanizmu PA-10 uporabili strežnik opisan v poglavju 3 ter programsko okolje MATLAB<sup>®</sup> SIMULINK<sup>®</sup>. Program za vodenje robota je bil v celoti zgrajen v okolju SIMULINK<sup>®</sup>. Razlogi za izbiro omenjenega programskega paketa so bile dobre predhodne izkušnje avtorjev. Komunikacija z strežnikom je bila realizirana z integriranimi funkcijami za pošiljanje paketov po UDP protokolu. S pravim formiranjem paketov je bilo mogoče pošiljati referenčne hitrosti sklepov ter prebirati trenutne kote v sklepih. Ker strežnik pošilja tudi izmerjene sile ter navore na vrhu robota preko JR3 senzorja je mogoče admitančno vodenje opisano v 5.1.

Avtorji tega dela so pripravili SIMULINK<sup>®</sup>shemo, ki se izvaja na posvečenem

računalniku na katerem teče operacijski sistem Simulink Target. Frekvenca izvajanja programa je nastavljena na 500 Hz tako, da se ujema s frekvenco strežnika. Pripravljen je bil grafični vmesnik, da lahko bodoči uporabniki uporabljajo enak program brez predhodnega poznavanja katere ukaze je potrebno izvesti.

Najprej se je implementiralo admitančno vodenje s referenčno silo enako 0 N. S spreminjanjem matrike ojačanj  $\mathbf{K}_h$  se je spreminjalo podajnost vrha robota. Večje kot je ojačanje, večja je podajnost. Vendar je pri tem zelo pomembno, da sistem ne postane nestabilen. Potrebno je vzeti v zakup, da izmerjene vrednosti senzorja sil JR3 lezejo. Posledično lahko robot počasi začne lesti nekam kamor to ni zaželeno. Ojačanje je bilo večinokrat nižje od 0.1. Kar v praksi pomeni, da se je vrh robota obnašal kot vzmet z koeficientom 10. Iz enačbe vzmeti  $F = \Delta x \cdot k$  je videt, da v kolikor deluje na vrh robota sila 10 N bo premik vrha enak 0.1 m.

Implementirano je bilo tudi vodenje v katerem se je signal senzorja sile integriralo in prištelo na referenčno pozicijo. S tem načinom je bilo dosežena možnost spreminjanja pozicije vrha robota. V praksi to pomeni, da lahko robot opravlja neko nalogo in v kolikor uporabniku ni všeč lega lahko z roko potisne na vrh robota tako, da bo robot bil v drugačni legi. To je bilo tudi preverjeno z eksperimentom.

## 5.3 Rezultati

V tem poglavju bodo predstavljeni rezultati v obliki grafov. Opisana bodo tudi ojačanja, ki so se uporabljala. Dejanske številske vrednosti.





## 6 Krmiljenje z inverzno dinamiko

Admitačno krmiljenje industrijskega robota PA-10 ni zahtevno za realizacijo. Delovanje je močno odvisno od signala pridobljenega iz senzorja sil ter ojačanj. Za realizacijo osnovnega krmiljenja po notranjih koordinatah je potrebno poznati le kinematičen model, ki pa je relativno enostaven za zgraditi. Pri krmiljenju na podlagi inverzne dinamike pa ni tako. Potrebuje se dinamičen model robota, ki pa je večinoma zakompliciran za zgraditi. Potrebno je dobro poznati dinamične parametre robotskega mehanizma. Če proizvajalec ni pripravljen te parametre posredovati jih je potrebno identificirati. Avtor v [7] predlaga in zgradi dinamičen model v namen kompenzacije gravitacije s pomočjo nevronske mreže. Avtorji tega dela so ta model uporabili v namene realizacije vodenja. Za kompenzacijo trenja pa so avtorji tega dela uporabili Stribeckov model trenja. Parametre za model pa so pridobili na podlagi eksperimenta.

Zgrajen je bil program v programskem okolju MATLAB® SIMULINK®, ki deluje s frekvenco 500 Hz in pošilja ukaze o željenih navorov po UDP protokolu na strežnik opisan v poglavju 2.2.1.

### 6.1 Teorija

Z uprabo Newton - Eulerjeve metode lahko zapišemo relacijo med navori v sklepih in pospeški, hitrostmi ter kotih v sklepih na sledeč način:

$$\tau = B(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + F_f \cdot \dot{q} + g(q) \quad (6.1)$$

kjer matrika  $\mathbf{B}(\mathbf{q})$  predstavlja matriko vztrajnosti, matrika  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  predstavlja matriko Coriolisovih in centrifugalnih vplivov, matrika  $\mathbf{F}_f$  predstavlja matriko parametrov trenja, matrika  $\mathbf{g}(\mathbf{q})$  pa vpliv gravitacije. Enačba predstavlja dinamičen model robota.

V nadaljevanju bodo opisani koraki, ki privedejo do vodenja robota v zunanjih koordinatah preko inverzne dinamike, začenši z vodenjem v notranjih koordinatah.

### 6.1.1 Vodenje v notranjih koordinatah

V poglavju 5.1.1 je bilo opisano vodenje robota v notranjih koordinatah. Ker je bila v tistem primeru vhodna veličina v krmilnik robota hitrost, je bilo to relativno enostavno realizirati. Ker pa je v primeru vodenja z inverzno kinematiko potrebno kot vhodno veličino podati navor, je pred tem potrebno narediti primeren regulator. Regulator prejme kot referenco pospešek in ven da potreben navor za to doseči. Ta se lahko nato pošlje na krmilnik robota. Izhodiščna enačba naj bo 6.1. Hitrost ter trenutni kot se lahko obravnava kot posledica vodenja, zato je regulirna veličina pospešek. V nadaljevanju bo definiran takšen  $\ddot{\mathbf{q}}$ , da bodo sklepi robota sledili referenčni legi.

Iz enačbe 6.1 je videt, da je potrebno pospešek z modelom matrike vztrajnostne matrike. V nadaljevanju bodo za lažje branje matrike, ki opisujejo uporabljeni model za označene z znakom  ${}^{\wedge}$ . Želen pospešek bo označen z  $\ddot{\mathbf{q}}_{ref}$

V kolikor je naloga preprosto sledenje nekih točk v prostoru lahko referenčni pospešek reguliramo na sledeč način:

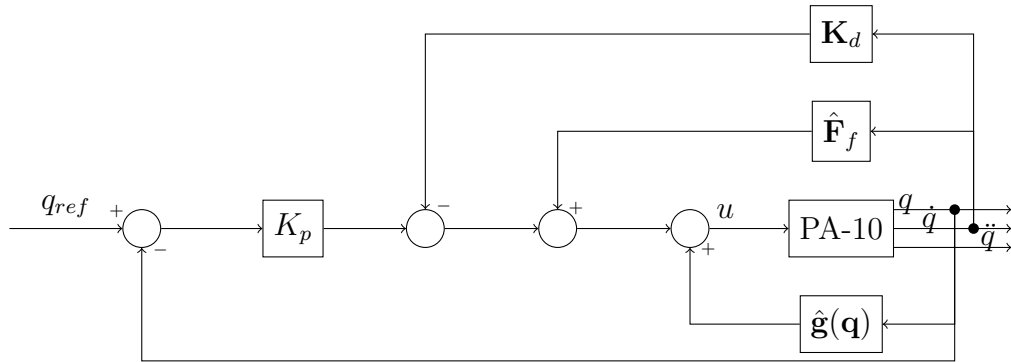
$$\ddot{\mathbf{q}}_{ref} = \mathbf{K}_p \cdot (\mathbf{q}_{ref} - \mathbf{q}) - \mathbf{K}_d \cdot (\dot{\mathbf{q}}) \quad (6.2)$$

Matrika  $\mathbf{K}_p$  predstavlja proporcionalno ojačanje,  $\mathbf{K}_d$  pa diferencirno ojačanje. Za dobro vodenje je potrebno upoštevati tudi napako, ki nastane zaradi trenja ter vpliva gravitacije in to tudi kompenzirati. Tako postane enačba navora za

sledenje referenč legi sledeča:

$$\mathbf{u} = \hat{\mathbf{B}}(\mathbf{q}) \cdot (\mathbf{K}_p \cdot (\mathbf{q}_{ref} - \mathbf{q}) - \mathbf{K}_d \cdot (\dot{\mathbf{q}})) + \hat{\mathbf{F}}_f \cdot \dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) \quad (6.3)$$

Shemo takega vodenja prikazuje shema 6.1



Slika 6.1: Bločna shema krmiljenja referenčne lege vrha robota z inverzno dinamiko

## 6.2 Implementacija

Postopek implementacije je bil v tem primeru podoben tistemu opisanemu v poglavju 5.2. Tudi tokrat so avtorji uporabili strežnik opisan v poglavju 3 ter programski paket MATLAB® SIMULINK®. Program je deloval s frekvenco 500 Hz na posevečenem računalniku na katerem je potekal operacijski sistem Simulink Target.

Implementacija je potekala postopoma za vsako stopjo posebaj. Začelo se je z zadnjim sklepom saj je najmanj odvisen od ostalih in njegovo premikanje ni preveliko. Testiralo se je najprej kompenzacijo trenja. Izključilo se je zavoro za posamezen sklep ter se ga je z roko porinilo. Teza je bila, da v kolikor sklep ohrani hitrost gibanja je navor dobro kompenziran. Težava je, da mora tukaj biti ojačano na meji stabilnosti. V kolikor je kompenzacija prevelika, bo sklep pospeševal, če pa je premajhna bo zaviral.

Sledila je implementacija kompenzacije gravitacije. Ponovno se je to poskušalo za vsak sklep posebej. Teza je bila, da v kolikor kompenzacija pravilo deluje, bo sklep miroval v kateri koli legi kljub temu, da so zavore izklopljene. Potrebno je bilo nekatera ojačanja prilagoditi, za pravilno delovanje.

Z dobrim modelom trenja in dobro kompenzacijo gravitacije se je testiralo še vodenje v notranjih koordinatah. Spet za posamezne stopnje. Kot referenco se je podalo neko sinusno nihanje. Opazovalo se je graf reference ter graf dejanskega kota. Prilagajalo se je ojačanja regulatorja tako, da se je graf dejanskega kota čimbolje ujemal z referenčnim.

Z dobro delujočim vodenjem v zunanjih koordinatah je sledila implementacija vodenja v zunanjih koordinatah. To je bilo opravljeno z enakimi bloki za izračun hitrosti v sklepih kot v primeru admitančnega vodenja. Ojačanja so bila drugače izbrana tako, da so odzivi bili hitri in, da je sistem še vedno bil stabilen.

## 6.3 Rezultati

V tem poglavju bodo predstavljeni rezultati v obliki grafov. Opisana bodo tudi ojačanja, ki so se uporabljala. Dejanske številske vrednosti.

## 7 Simulink knjižice

Avtorji tega dela so za visoko nivojsko vodenje uporabljali programski paket MATLAB<sup>®</sup> SIMULINK<sup>®</sup>. Ker je želja avtorjev, da se raziskovalno in razvojno delo na robotu PA-10 v prihodnosti nadaljuje, so nastale SIMULINK<sup>®</sup> knjižnice. Knjižnice bodočim uporabnikom omogočajo uporabo robota brez, da poznajo do podrobnosti vse, kar je bilo opisano do sedaj v tem delu. V nadaljevanju bodo opisani posamezni bloki, ki jih lahko uporabniki dodajo v njihovo SIMULINK<sup>®</sup> shemo.

### 7.1 Blok PA-10

Blok imenovan PA-10 je glavni blok za komunikacijo z ARCNET strežnikom, ki teče na Linux računalniku. Blok omogoča prebiranje vseh statusnih bajtov za vse stopnje robota in glavni statusni bajt. Omogoča tudi prebiranje vrednosti sil izmerjenih na senzorju JR3 ter kote in navore v sklepih robota. Hkrati pa uporabniku omogoča izbiro na kakšen način bo robota vodil, navorno ali hitrostno. Uporabnik ima tudi možnost vklopa ter izklopa zavor posameznih sklepov ter vklop ali izklop posameznih servo motorjev.

### 7.2 Blok Inverse Dynamics Control

Blok imenovan Inverse Dynamics Control je nastal z željo, da uporabnik enostavno pripelje v vhod željeno pozicijo v sklepih ali hitrost sklepov v blok in ven

dobi potreben navor za to hitrost oz. za ohranjanje pozicije. Uporabnik ima možnost nastavljanja proporcionalnega ter diferencirnega ojačanja.

### **7.3 Blok Follow Trajectory**

Blok imenovan Follow Trajectory implementira vodenje v zunajih koordinatah. Kot vhod prejme zunanje koordinate, trenutne kote in hitrosti v sklepih. Kot rezultat vrne potrebne hitrosti, za doseganje referenčne pozicije. V bloku je realizirano vodenje na način, ki je opisan v 5.1.2.

### **7.4 Blok Admitance Control**

Blok imenovan Admitance Control je namenjen admitačnemu vodenju robota. Blok prejme kot vhod signal želeno referenco izraženo v zunajih koordinatah ter odčitke senzorja sil. Kot izhod pa poda novo referenčno pozicijo v zunajih koordinatah, ki jo lahko uporabnik posreduje naprej v blok Follow Trajectory. Bloku Admitance Control se lahko natavlja ojačanja v posameznih smereh. Na ta način se doseže, da je po različnih oseh različno podajen.

## 8 Zaključek





## Literatura

- [1] M. H. I. Ltd., “Mitsubishi clean room robot,” *Seminarska naloga*, 2003.
- [2] J. J. Craig, *Introduction to Robotics*. Z.D.A.: Pearson Education International, 2005.
- [3] B. Siciliano, L. Sciavicco, L. Villani in G. Oriolo, *Robotics, Modelling, Planning and Control*. London: Springer, 2009.
- [4] N. D. Vuong in M. H. J. Ang, “Dynamic model identification for industrial robots,” *Acta Polytechnica Hungarica*, 2009.
- [5] R. van der Aalst, M. H. J. Ang in H. Nijmeijer, “Dynamic identification of a mitsubishi pa-10 robotic manipulator,” *ICAR*, 1999.
- [6] R. Jamisola, M. J. Ang, T. M. Lim, O. Khatib in S. Y. Lim, “Dynamics identification and control of an industrial robot,” *ICAR*, 1999.
- [7] T. Petrič in M. Munih, “Kompenzacija trenja in gravitacije na robotu mitsubishi pa-10,” *Seminarska naloga*, 2010.
- [8] I. JR3, “JR3, Installation manual for force - torque sensors with external electronics.” Dosegljivo: <http://www.jr3.com/product-manuals.html>. [Dostopano: 6. 17. 2015].
- [9] I. JR3, “JR3, DPS-based force sensor receivers, software and installation manual.” Dosegljivo: <http://www.jr3.com/product-manuals.html>. [Dostopano: 6. 17. 2015].

- [10] A. Pennarun, “Linux ARCnet HOWTO.” Dosegljivo: <http://apenwarr.ca/arcnet/howto/index.html>. [Dostopano: 6. 17. 2015].
- [11] J. Lenarčič in T. Bajd, *Robotski mehanizmi*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2003.
- [12] M. Mihelj, *Vodenje robotov*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2011.
- [13] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik in M. Munih, *Robotika*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2008.
- [14] M. Mihelj, T. Bajd in M. Munih, *Vodenje robotov*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2011.

# Dodatek



## A Appendix 1



## B Appendix 2





## C Appendix 3

Postopek dela:



## D Predloge za navajanje literature - baza BibTex

```
@ARTICLE{clanek1,  
  author = "L[eslie] A. Aamport",  
  title = "The Gnats and Gnus Document Preparation System",  
  journal = "\mbox{G-Animal's} Journal",  
  year = 1986,  
  volume = 41,  
  number = 7,  
  pages = "73-77",  
  month = jul,  
}
```

```
@BOOK{knjiga1,  
  author = "Donald E. Knuth",  
  title = "Seminumerical Algorithms",  
  publisher = "Addison-Wesley",  
  address = "Reading, Massachusetts",  
  year = "1981",  
}
```

```
@INPROCEEDINGS{vzborniku,  
  author = "Alfred V. Oaho and Jeffrey D. Ullman and Mihalis Yannakakis",  
  title = "On Notions of Information Transfer in {VLSI} Circuits",  
  editor = "Wizard V. Oz and Mihalis Yannakakis",  
}
```

```
booktitle = "Proc. Fifteenth Annual ACM" # STOC,  
pages = "133--139",  
month = mar,  
year = 1983,  
address = "Boston",  
publisher = "Academic Press",  
}
```

```
@misc{spletna_stran,  
author = "LLC",  
title = "{MS Windows NT Kernel Description [Online]}",  
howpublished = "Dosegljivo: \url{http://web.archive.org}",  
note = "[Dostopano: 19. 4. 2013]"  
}
```