

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

1	Uvod	5
2	Robotski manipulator PA-10	7
2.1	Kinematični model robota PA-10	7
2.2	Dinamičen model robota	12
2.3	Inverzna kinematika	14
3	ARCNET - UDP strežnik za komunikacijo s krmilnikom robota ter senzorjem sile	17
3.1	Krmilnik servo motorjev robota	18
3.2	ARCNET vmesnik	19
3.3	Senzor sile in navorov - JR3	20
3.4	Strežnik	22
3.4.1	Delovanje programa	23
3.4.2	Varnostni ukrepi	25
4	Vodenje robota PA-10	27
4.1	Kompenzacija mase prijemala ter merilnega odmika na senzorju sile JR3	27
4.2	Vodenje preko inverzne kinematike	30

4.2.1	Vodenje v notranjih koordinatah	31
4.2.2	Vodenje v zunajih koordinatah	32
4.3	Admitančno vodenje	33
4.3.1	Upraba zunanjih sil	34
4.4	Implementacija	35
4.5	Rezultati	36
5	Krmiljenje z inverzno dinamiko	37
5.1	Teorija	37
5.1.1	Vodenje v notranjih koordinatah	38
5.2	Implementacija	39
5.3	Rezultati	40
6	Simulink knjižice	41
6.1	Blok PA-10	41
6.2	Blok Inverse Dynamics Control	41
6.3	Blok Follow Trajectory	42
6.4	Blok Admittance Control	42
7	Zaključek	43
A	Appendix 1	49
B	Appendix 2	51
C	Appendix 3	53
D	Predloge za navajanje literature - baza BibTex	55

Seznam slik

2.1	Skica robotskega mehanizma PA 10.	8
2.2	Dva zaporedna člena kinematične verige, povzeto po [1]	9
2.3	Tehnična risba robota PA-10. Iz nje se definira D-H parametre. Povzeto po [2].	10
3.1	Tehnična risba ohišja krmilnika servo motorjev. Povzeto po [2]. . .	18
3.2	Način za krmiljenje robota	21
4.1	Koordinatni sistem senzorja sile ter koordinatni sistem telesa prijemala	28
4.2	Povratna zanka za krmiljenje kota v sklepu	31
4.3	Bločna shema krmiljenja referenčne lege vrha robota	33
5.1	Bločna shema krmiljenja referenčne lege vrha robota z inverzno dinamiko	39

Seznam tabel

1	Veličine in simboli	xi
2.1	D-H parametri	10
3.1	Opis prejetega UDP paketa.	23
3.2	Opis ARCNET paketa.	24
3.3	Opis poslanega UDP paketa.	25

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

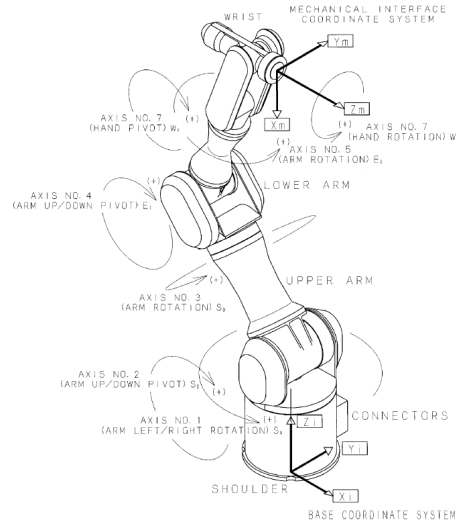
2 Robotski manipulator PA-10

Podjetje Mitsubishi Heavy Industries je leta 1992 izdelalo prvega katalogiranega industrijskega redundatnega robota [3]. Podjetje je robota poimenovalo Portable General-purpose Intelligent Arm PA-10, krajše PA-10. Gre za serijskega robota s sedmimi stopnjami prostosti (slika 2.1). 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 [4]. Značilnost takih robotov je spretnost saj so vsi sklepi rotacijski [1].

Masa robotske roke PA-10 je 36 kg in ima nosilnost 10 kg. Servo motorji v sklepih se napajajo preko izmenične napetosti. Prenosi med v sklepih so realizirani z harmoničnimi gonili. Baza robota da se lahko pritrdi v katerokoli lego. To pomeni, da se ga lahko fiksira bodisi na tla, na steno ali na strop. Robota PA-10 odlikuje relativno lahka konstrukcija, enostavno rokovanje ter odprtost njegovega krmilnika. Prav ti razlogi so povod, da je mnogo instituciji vzelo tega robota kot eksperimentalni sistem za razvijanje raznih algoritmov vodenja ([5], [6], [7], [8]).

2.1 Kinematični model robota PA-10

Robotski mehanizem obravnavamo kot kinematično verigo n med seboj povezanih togih teles, rečemo jim tudi kinematični pari. Ker je en konec kinematične verige, t.j. baza robota, toga povezan v bazo se premika le vrh kinematične verige. Z opisom kinematične relacije med dvema zaporednima segmentoma je



Slika 2.1: Skica robotskega mehanizma PA 10.

mogoče definirati kinematično relacijo med bazo in vrhom robota. Homogena transformacijska matrika je operator, ki opisuje translacijo in rotacijo med dvema koordinatnima sistemoma. Definiramo jo kot [1]

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

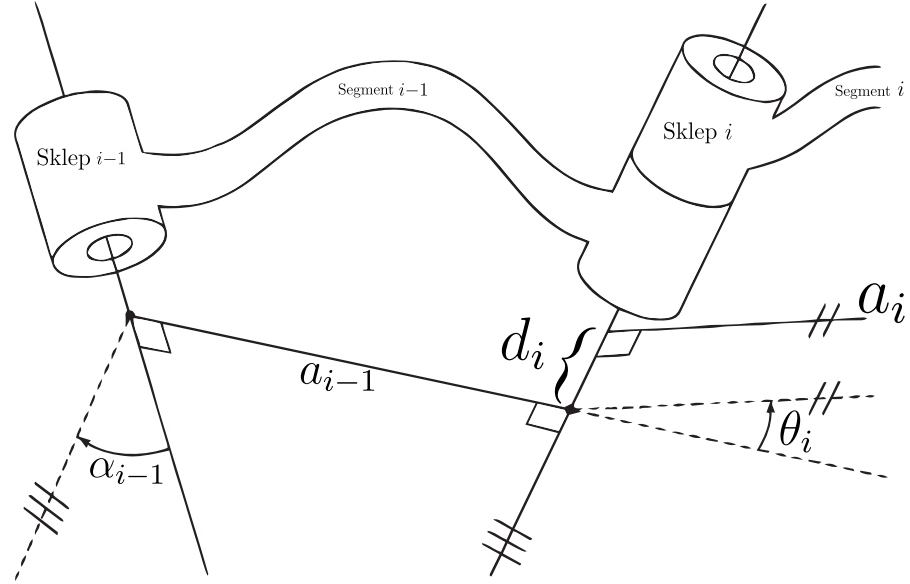
kjer so a_i , α_i , d_i in Θ D-H parametri in opisujejo relacijo med dvema različnima koordinatnima sistemoma.

Z množenjem homogenih transformacijskih matrik posameznih sklepov med seboj dobimo homogeno transformacijo med vrhom robota in njegovo bazo:

$$\mathbf{T}_7^0 = \mathbf{T}_1^0 \mathbf{T}_2^1 \dots \mathbf{T}_7^6. \quad (2.2)$$

Sklicujoč na sliko 2.2 se D-H parametre parametre opiše kot:

- a_i razdalja med O_i in O_{i+1} v smeri x_i ,

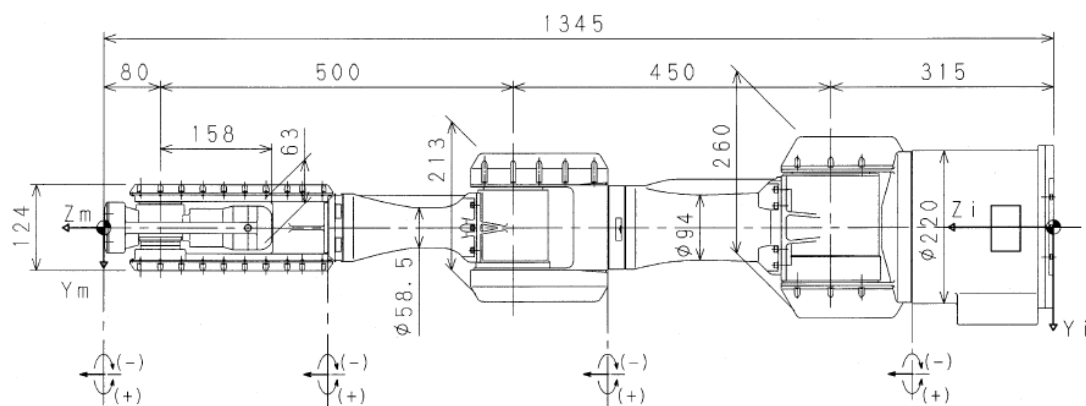


Slika 2.2: Dva zaporedna člena kinematične verige, povzeto po [1]

- α_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 ,
- θ razdalja med O_{i-1} in O_i glede na os z_i ,

Te parametre se za primer robota PA-10 definira s pomočjo dokumentacije [2] oziroma iz slike 2.3. V tabeli 2.1 so zapisani D-H parametri za vsak kinematični par. Koti v sklepih robota so zapisani kot q_n in jih imenujemo tudi notranje koordinate. Glede na D-H sistem bodo v našem primeru ustrezali parametru θ_i . Z upoštevanjem tega postane matrika \mathbf{T}_{i+1}^i funkcija kotov v sklepih $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_7]^T$.

Opis lege vrha robota se da skrajšati in zapisati z minimalnim številom koordinat. Iz matrike homogene transformacije je mogoče izpisati tri koordinate, ki opisujejo pozicijo vrha robota in tri koordinate, ki opisujejo njegovo orientacijo. Pozicija se enostavno izpiše iz prvih treh vrstic zadnjega stolpca. Za zapis orientacije pa se je potrebno sklicati na Eulerjeve kote. Prvi trije stolpci in prve tri



Slika 2.3: Tehnična risba robota PA-10. Iz nje se definira D-H parametre. Povzeto po [2].

i	α_{i-1}	a_{i-1}	d_i	θ_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

vrstice v matriki homogene transformacije opisujejo spremembo orientacije. Če omenjeno podmatriko zapišemo kot

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (2.3)$$

lahko definiramo tri Eulerjeve kote na sledeč način

$$\varphi = \arctan_2(r_{21}, r_{11}), \vartheta = \arctan_2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}), \psi = \arctan_2(-r_{32}, -r_{33}). \quad (2.4)$$

Z združitvijo treh koordinat za opis pozicije in treh koordinat za opis orientacije se lahko definira vektor, ki opisuje neko točko v prostoru baze robota:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \varphi & \vartheta & \psi \end{bmatrix}^T. \quad (2.5)$$

Vektor x se imenuje tudi vektor zunanjih koordinat. Sedaj je mogoče opisati enačbo direktne kinematike kot funkcijo \mathbf{q} :

$$\mathbf{x} = \mathbf{k}(\mathbf{q}). \quad (2.6)$$

Funkcija $\mathbf{k}(\mathbf{q})$ je vektorska funkcija notranjih koordinat, v katerih so zajete kinematične enačbe pozicije in orientacije mehanizma. Problem direktne kinematike je pri serijskih mehanizmih enostavno rešljiv in ima eno rešitev [9].

Potrebno je še opisati kakšna je hitrost vrha robota v odvisnosti od hitrosti v sklepov. Z odvajanjem enačbe 2.6 po q dobimo

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{K}(q)}{\partial \mathbf{q}} \mathbf{q} = \mathbf{J} \dot{\mathbf{q}}, \quad (2.7)$$

kjer je \mathbf{J} Jacobijeva matrika in predstavlja parcialne odvode zunanjih koordinat.

Z odvajanjem enačbe ?? je mogoče zapisati še relacijo med pospeški vrha robota ter pospeški sklepov

$$\ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}, \quad (2.8)$$

kjer je $\dot{\mathbf{J}}$ odvod Jacobijeve matrike.

2.2 Dinamičen model robota

Vodenje robota z referenčnimi navori veleva poznavanje njegovega dinamičnega modela. S poznavanjem dinamičnega modela je mogoče opisati silo s katero bo vrh robota deloval v kontaktu z okolico. Z razliko od kinematičnega modela je pri dinamičnem modelu je število parametrov, ki vplivajo na vodenje večje. V nadaljevanju bo predstavljen splošen dinamičen model in relacija med silami, ki delujejo na vrhu robota in navori v sklepih.

Enačbe gibanja kot posledica delovanja sil in navorov se lahko zapišejo z uporabo Lagrangevih formulacij

$$\begin{aligned} \mathcal{L}(q, \dot{q}) &= \mathcal{T}(q, \dot{q}) - \mathcal{U}(q), \\ \boldsymbol{\tau} &= \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} \end{aligned} \quad (2.9)$$

kjer \mathcal{T} in \mathcal{U} opisujeta kinetično in potencialno energijo.

Lagrangeve formulacije so orodje za sistematičen opis dinamike posameznih segmentov [1]. Z uporabo te formulacije se lahko zapiše enačbo navorov v sklepih v odvisnosti od kotov, hitrosti in pospeškov v sklepih na sledeč način:

$$\boldsymbol{\tau}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_f(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\mathbf{h}_o. \quad (2.10)$$

Posamezni členi zgornje enačbe bodo opisani v nadaljevanju.

Matrika $\mathbf{B}(\mathbf{q})$ predstavlja zapis vztrajnosti posameznih segmentov. Odvisna je od trenutne konfiguracije robota. Posamezni členi pa so izračunani kot

$$h_{i,j} = \sum_{i=\max(j,k)}^n Tr[(\frac{\partial}{\partial q_k} \mathbf{T}_i^0)(\frac{\partial}{\partial q_j} \mathbf{T}_i^0)]. \quad (2.11)$$

Operator Tr je sled matrike, to je vsota diagonalnih členov matrike.

Matrika $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ vključuje podatke o Coriolisovih in centripetalnih silah. Njeni členi izraženi s pomočjo kot

$$c_{i,j} = \sum_{k=1}^n c_{i,j,k} \dot{q}_k, \quad (2.12)$$

kjer so

$$c_{i,j,k} = \frac{1}{2} \left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right), \quad (2.13)$$

Christoffelovi simboli in dodatno velja še $i, j, k = 1, \dots, n$.

Vektor $\mathbf{g}(\mathbf{q})$ predstavlja navor proizveden v sklepu manipulatorja i zaradi vpliva gravitacije v trenutni konfiguraciji [1]. Posamezen člen vektorja $\mathbf{g}(\mathbf{q})$ je podan kot

$$g_i = \sum_{i=j}^n (-m_i \mathbf{g}_0 (\frac{\partial}{\partial q_j} \mathbf{T}_i^0)^T \mathbf{r}_i), \quad (2.14)$$

kjer je \mathbf{g}_0 vektor gravitacijskega pospeška $\mathbf{g}_0 = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$. Vektor \mathbf{r}_i opisuje mesto težišča v segmentu, m_i pa opisuje maso segmenta.

Vektor \mathbf{F}_f opisuje vpliv trenja. Vektor zajema tako Coulombovo trenje kot tudi viskozno trenje. Posamezni členi vektorja, f_i so prispevki trenja v i -tem sklepu. Podrobneje se bo ta prispevek obravnaval v poglavju 5.

Vektor \mathbf{h}_o vsebuje sile in navore, ki delujejo na vrh robota, $\mathbf{h}_o = \begin{bmatrix} f_x & f_y & f_x & m_x & m_y & m_z \end{bmatrix}$, kjer prvi trije členi vektorja opisujejo sile, drugi trije pa navore.

2.3 Inverzna kinematika

Nalogo, ki jo opravlja robot tipično opišemo s časovnim potekom vektorja x , kot je podan v enačbi 2.6. Da bi lahko izvedli želeno gibanje, je potrebno poiskati ustrezne vrednosti notranjih koordinat \mathbf{q} , kar predstavimo z enačbo

$$\mathbf{q} = \mathbf{K}^{-1}(\mathbf{x}), \quad (2.15)$$

kjer je \mathbf{K}^{-1} inverz funkcije \mathbf{K} . Ta funkcija predstavlja inverzno kinematiko. Čeprav je bila rešitev enačbe 2.6 enolična, pa enačba 2.15 nima vedno enolične rešitve. Še več, rešitev obstaja le, če \mathbf{x} leži v delovnem prostoru robota.

Pri vodenju robotov večinoma ne rešujemo inverzne dinamike direktno iz enačbe 2.15 ampak preko hitrosti, torej iz enačbe 2.7. Če predpostavimo, da je dimenzionalnost prostora naloge m in če velja, da je $m = n$, potem lahko določimo hitrost v sklepih iz relacije

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}}, \quad (2.16)$$

kjer je \mathbf{J}^{-1} inverz Jacobijeve matrike.

V primeru, da pa ima robot več stopenj prostosti, kot jih je potrebno za izvedbo naloge, torej če je $n > m$, potem inverz \mathbf{J}^{-1} ne obstaja in je potrebno poiskati rešitev inverzne kinematike drugače. Enačbo 2.7 lahko invertiramo z uporabo naslednje zveze

$$\dot{\mathbf{q}} = \mathbf{J}^*\dot{\mathbf{p}} + \mathbf{N}\dot{\mathbf{q}}, \quad (2.17)$$

kjer je \mathbf{J}^* generaliziran inverz matrike \mathbf{J} , matrika \mathbf{N} pa predstavlja projekcijo v ničelni prostor matrike \mathbf{J} ,

$$\mathbf{N} = (\mathbf{I} - \mathbf{J}^\# \mathbf{J})$$

.

Prvi člen enačbe 2.17 predstavlja partikularno rešitev iz zadosti osnovni nalogi, to je zagotovi, da je vrh robota v \mathbf{x} . Drugi člen enačbe 2.17 pa predstavlja homogeno rešitev in omogoča rekonfiguracijo robotskega mehanizma brez, da se

spremeni pozicija vrha robota \mathbf{x} . Zaradi tega se hitrost \dot{q}_n lahko uporabi za realizacijo dodatnih nalog nižje prioritete.

Kot generaliziran inverz matrike \mathbf{J}^* se pogosto uporablja Moore-Penroseov pseudoinverz

$$\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}. \quad (2.18)$$

ali uteženi Moore-Penroseov pseudoinverz

$$\mathbf{J}^* = \mathbf{J}^\# = \mathbf{W}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T)^{-1}, \quad (2.19)$$

kjer je \mathbf{W} utežnostna matrika.

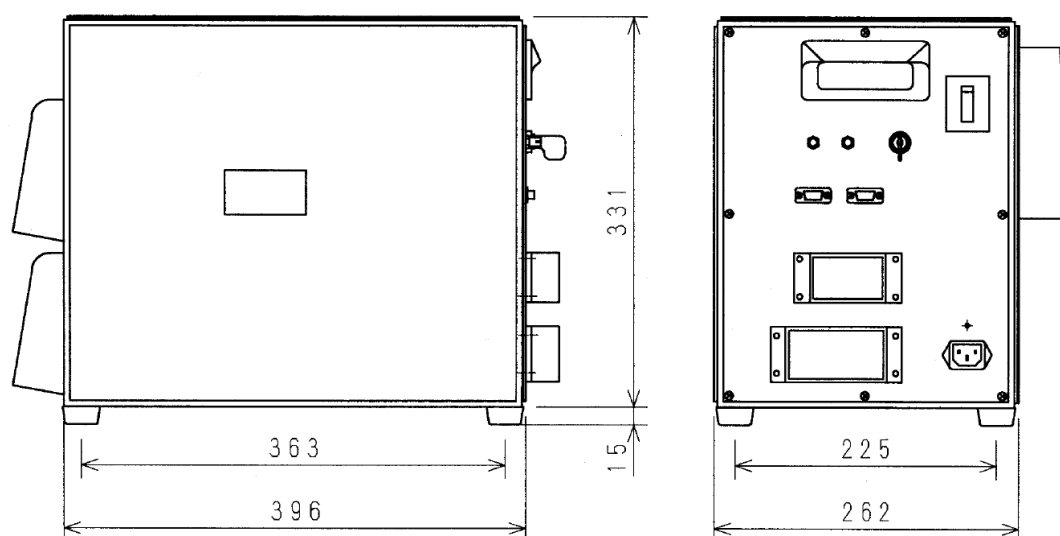
Z odvajanjem enačbe 2.17 dobimo relacijo med pospeški v sklepih in pospeških na vrhu robota.

3 ARCNET - UDP strežnik za komunikacijo s krmilnikom robota ter senzorjem sile

V prejšnjem poglavju je bil opisan kinematični in dinamičen model robotskega mehanizma. V sklepih robota PA-10 so servo motorji na izmenični tok. Krmiljenje motorjev se izvaja v krmilniku, ki ga proizvajalec priloži ob nakupu robota. Krmilnik omogoča vodenje servo motorjev bodisi preko referenčne hitrosti ali referenčnega navora. V krmilniku je vgrajen ARCNET vmesnik, ki omogoča priključitev krmilnika v ARCNET omrežje. Na ta način je omogočena komunikacija s krmilnikom [2].

Z uspešnim vodenjem robota brez taktilne povratne zanke je mogoče izvesti mnogo različnih nalog. V zgornjem poglavju je bil opisan dinamičen model robota. S poznavanjem vseh parametrov je v teoriji mogoče izračunati sile, ki delujejo na vrh robota \mathbf{h}_o . V praksi se izkaže, da te parametre ni vedno mogoče dovolj natančno definirati. Iz teh razlogov se vgradi senzor sil in navorov med vrhom robota ter prijemalom ([10], [11], [12]). Senzor sil in navorov JR3 je bil uporabljen pri izvedbi tega dela.

Želja po združitvi vodenja robota in merjenja sil je privedla do tega, da so avtorji tega dela naredili strežnik ki to omogoča. Dodaten razlog je bil to, da prejšnja programska oprema za komunikacijo s servo krmilnikom, ni zadoščala časovnemu kriteriju delovanja s frekvenco 500 Hz.



Slika 3.1: Tehnična risba ohišja krmilnika servo motorjev. Povzeto po [2].

3.1 Krmilnik servo motorjev robota

V prvem poglavju je bilo zapisano, da robota PA-10 med drugim odlikuje prenosnost. To se velja tako za samo robotsko roko kot tudi za krmilnik servo motorjev. Krmilnik z ohišjem ima dimenzije $262 \times 331 \times 396$ mm in tehta 22 Kg. Za sprednji strani krmilnika sta dva zračnika, na zadnji pa so priključki in stikala.

V ohišju servo krmilnika so vgrajeni štirje krmilniki servo motorjev. Vsak servo krmilnik krmili dva servo motorja, razen enega. Krmilniki omogočajo vodenje motorjev na dva načina. Prvi način je navorno, drugi pa hitrostno. Različna vodenja sta v krmilnikih drugače realizirana. Navorno vodenje je realizirano z analognim P regulatorjem toka, hitrostna regulacija pa je realizirana z digitalnim PI regulatorjem s frekvenco približno 1500 Hz.

Krmilnik vsebuje dva pomnilnika. Delovni pomnilnik (RAM) in nastavitveni pomnilnik (EEPROM). V EEPROM tabeli so zapisani razni parametri za nastavitve in vodenje servo krmilnikov. Ob vžigu krmilnika se parametri naložijo iz EEPROM tabele v RAM. Med temi parametri so tudi ojačanje proporcional-

nega ter integracijskega dela regulatorja, omejitve posameznih stopenj, razmerje prenosa zobnikov, itd.

3.2 ARCNET vmesnik

Referenčne navore in hitrosti se na krmilniku nastavlja na preko zunanjega vmesnika, ki je povezan na isto ARCNET omrežje kot servo krmilnik. Priključitev na ARCNET omrežje servo krmilniku omogoča v ohišje vgrajen ARCNET modul. ARCNET je omrežje, ki vključuje podatkovni in fizični nivo po OSI modelu. Njegova prednost pred Ethernet omrežjem je velika stopnja determinističnosti [13]. Krmilnik ima dva priključka za optična vodila, vhod (Rx) ter izhod (Tx). Zgornja meja hitrosti komunikacije za robota PA-10 je 5 Mb/s [2].

ARCNET komunikacija je serijska in paketno zastavljena. Za komunikacijo z modulom je potrebno vsak paket primerno sestaviti. Po [2] je potrebno paket začeti z naslovom modula, sledi bajt, ki označuje veliko črko po ASCII formatu, v nadaljevanju. ARCNET modul dodatno vsebuje logiko za preklapljanje med različnimi stanji. Med stanji preklapljam s pošiljanjem ustrezne črke v paketu. Različna stanja pa so:

- 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,
- Čakanje

Ko višje nivojski vmesnik ne pošilja nobenih ukazov je ARCNET vmesnik v čakanju. V tem stanju čaka na primerno sestavljen paket.

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) (slika 3.2). Pri tem načinu delovanja je potrebno posvetiti nekaj pozornosti frekvenci pošiljanja. V kolikor paketi ne pridejo do ARCNET vmesnika v intervalu specificiranem v EEPROM tabeli na krmilniku, bo vmesnik šel iz tega načina delovanja nazaj na način čakanje.

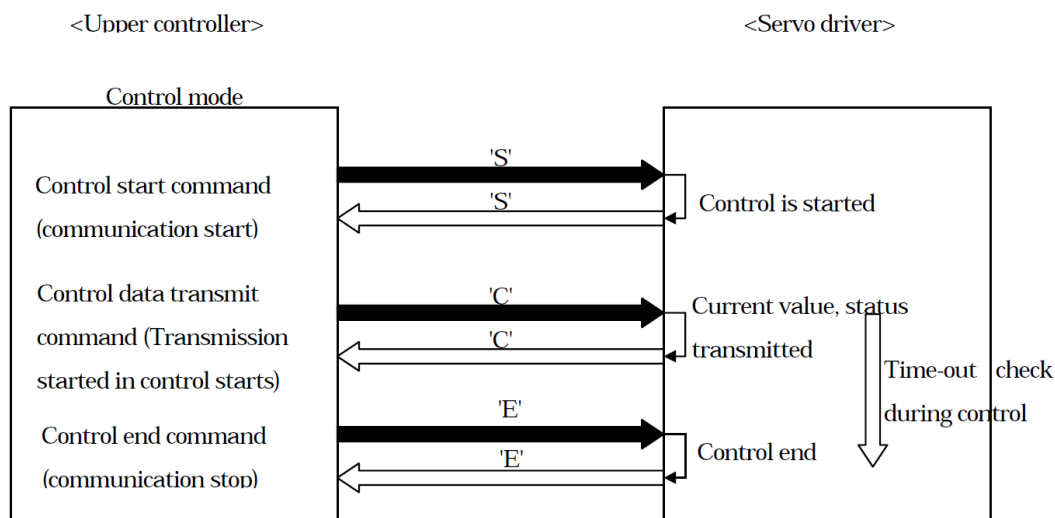
Paket, ki se začne s C in nosi podatke o krmiljenju sklepov robota je velik 35 bajtov, 5 za vsak sklep. Prvi bajt je kontrolni in samo prvi trije biti nosijo podatke o vodenju:

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.

3.3 Senzor sile in navorov - JR3

Za namene vodenja opisanega v poglavju ?? je med vrhom robota ter prijemalom pritrdilo senzor sil JR3. Senzor nam omogoča merjenje sil in navorov v X, Y in



Slika 3.2: Način za krmiljenje robota

Z oseh. Senzor je narejen iz uporovnih lističev porazdeljenih po notranjosti v obliki križa. Iz specifikacij senzorja [14] je razvidno, da uporablja 8 uporovnih lističev. S tem, ko na senzor deluje zunanja sila ali navor, pride do deformacije materiala na katerem so nameščeni uporovni lističi in se upornost na uporovnih lističih spremeni proporcionalno s silo in togostjo senzorja. Sprememba upornosti se izmeri posredno preko spremembe napetosti na AD-pretvorniku, ki je vgrajen v sam senzor. Napetost se pretvori v silo preko množenja s kalibracijsko matriko \mathbf{K} . Naj bo \mathbf{h}_{JR3} vektor sil, ki delujejo na senzor in naj bo ω_u vektor napetosti na AD-pretvorniku. Enačba za izračun sil je

$$\mathbf{h}_{JR3} = \mathbf{K}\omega_u \quad (3.1)$$

Senzor se priključi preko 6 ali 8 žičnega kabla na računalniško kartico, ki je bodisi na ISA ali PCI vodilu. Kabel serijsko pošilja podatke o izmerjenih silah na kartico. Kartica vsebuje vezje za digitalno obdelavo signalov. V dokumentaciji kartice [15] so opisani trije nizkoprepustni filtri z različnimi mejnimi frekvencami. V tem delu se je uporabilo filter z mejno frekvenco 500 Hz.

Na računalniku, ki se je uporabljal za to delo, 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 se lahko dostopa po registrih se nahajajo v dokumentaciji [15]. Dokumentacija navaja, da v kolikor želi uporabnik odčitati vrednost sile ali navora F 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 deliti s konstanto K_p , da pretvori v primerne enote (N ali Nm).

$$F = \frac{F_s F_f}{K_p} \quad (3.2)$$

Ker senzor meri silo in navor v treh smereh je potrebno opraviti izračun za vsako od izmerjenih veličin potrebno opraviti omenjeno operacijo.

3.4 Strežnik

Proizvajalec Mitsubishi Heavy Industries je za upravljanje in programiranje robota pripravil posebno programsko opremo. Iz dokumentacije je razvidno kaj je vse njihova programska oprema omogočala. Težava pa je, da omenjena programska oprema ni več dostopna. Proizvajalec je s prodajo robotov PA-10 prenehal v letu 2009, s podporo pa v marcu 2014 (vir: osebno dopisovanje s proizvajalcem). Iz tega razloga je nastala potreba, po lastnem programu, ki bi lahko komuniciral s servo krmilnikom.

Pred začetkom izdelave lastnega programskega orodja se je definiralo dve zahtevi. Prva je bila, da program deluje kar se le da hitro in kar se le da v realnem času. To izhaja iz vidika stabilnosti vodenja. Avtor v [16] namreč navaja, da vzorčni čas močno vpliva na stabilnost vodenja robota v kontaktu z okolico. Večja kot je vzorčna frekvenca večja je lahko togost okolice s katero je robot v kontaktu.

Druga zahteva pa je bila enostavna uporaba. Želeli smo, da uporaba pro-

gramske opreme, ki jo naredimo, zahteva čim manj predznanja o komunikacijskih protokolih po ARCNET mreži. Hkrati pa mora program omogočati neko mero svobode pri realizaciji visoko nivojskega vodenja.

Dodatno se je kasneje pojavila še želja, da bi programska oprema omogočala tudi posredovanje meritev izvedenih na senzorju sile JR3. Tako bi nastal nek program, ki bi komuniciral tako s servo kontrolerjem kot s ISA kartico za senzor sile JR3.

Naštete zahteve so privedle do izbire primerne operacijskega sistema in programskega jezika. Tako je bil razvit strežnik v programskem jeziku C na operacijskem sistemu Linux. Strežnik deluje kot posrednik med dvema različnima omrežjema, ARCNET in UDP. V nadaljevanju bo opisana zgradba tega strežnika, delovanje ter uporaba.

3.4.1 Delovanje programa

Kot povedano, je nastali program strežnik, kar pomeni, da deluje na klientovo zahtevo. V tem primeru je klient kakršen koli program, ki lahko pošilja UDP pakete po Ethernet omrežju. Klient mora tako formirati ustrezen UDP paket za začetek komunikacije. Paket je velik 99 bajtov. Vsebina paketa pa je opisana v tabeli 3.1.

Zaporedna številka bajta	Opis podatka	Tip podatka	Količina
0	Kontrolni bajt za posamezen motor	char	×7
6	Referenčni navor za posamezen motor	signed short int	×7
20	Referenčna hitrost za posamezen motor	signed short int	×7
34	Rezervirani / neuporabljeni bajti	char	×62
96	Štartna sekvenca	char	×1
97	Trenuten čas na klientu	float	×1

Tabela 3.1: Opis prejetega UDP paketa.

Kontrolni bajt, referenčni navori ter hitrosti za posamezen motor se posre-

dujejo naprej na krmilnik robota. Ostali bajti pa so namenjeni zagotavljanju varnosti med komunikacijo. Referenčni navori so v Nm, hitrosti pa v rad/s.

S prispelim UDP paketom strežnik pretvori dane navore in hitrosti v format, ki je naveden v dokumentaciji in opisan v 3.2. Kontrolni bajti za posamezne motorje ostanejo nespremenjeni in se posredujejo enaki kot so prispeli. Strežnik prispele podatke zapakira v ARCNET paket naveden v tabeli 3.2. Paket pošlje po ARCNET omrežju

Zaporedna številka bajta	Opis podatka	Količina
0	ID pošiljatelja	×1
1	ID prejemnika	×1
2	Prazen / neuporabljen bajt	×1
3	Mesto sekvencijskega števila	×1
4	Sekvencijsko število	×1
5	Tip podatka	×1
6	Statusni bajt za prvi servo	×1
7	Referenčni navor za prvi servo	×2
9	Referenčna hitrost za prvi servo	×2
	⋮	
36	Statusni bajt za sedmi servo	×1
37	Referenčni navor za sedmi servo	×2
39	Referenčna hitrost za sedmi servo	×2

Tabela 3.2: Opis ARCNET paketa.

Strežnik nato počaka na odgovor servo krmilnika. V kolikor odgovor ne prispe v vnaprej predpisanem času, se program zaustavi. Strežnik odgovor servo krmilnika najprej pretvori v zeleno obliko. Hitrosti in navori v sklepih so ponovno zapisani v formatu navedenem v 3.2. Te podatke pa nato zapakira v UDP paket. Zraven doda še kontrolne bajte. UDP paket poslan klientu je opisan v tabeli 3.3.

Po tistem, ko strežnik prejme in pošlje vse podatke, gre v stanje čakanja, da mine od začetka programskega cikla do konca točno 2 ms. Ohranjanje frekvence časovnega cikla je kritičnega pomena.

Zaporedna številka bajta	Opis podatka	Tip podatka	Količina
0	Kontrolni bajt za posamezen motor	char	×7
7	Trenuten kot v sklepu	float	×7
35	Trenutni proizveden navor v sklepu	signed short int	×7
63	Izmerjene sile in navori na JR3	float	×6
73	Rezerviran prostor	float	×1
112	Čas, ki je bil porabljen za programski cikel	float	×1
116	Čas, ki ga je poslal klient na bajtu 60	float	×1

Tabela 3.3: Opis poslanega UDP paketa.

3.4.2 Varnostni ukrepi

V tabelah, ki opisujejo poslane in prejete UDP pakete je mogoče razbrat, da so nekateri bajti namenjeni ohranjanju komunikacije oz. preverjanju ali sploh še poteka. Eden od takih ukrepov je, da v kolikor nov UDP ali ARCNET paket ne prispe v nekem vnaprej definiranem času, program robota zaustavi nato pa gre v čakanje novega paketa. Ko je program v čakanju preverja prispele UDP pakete, v kolikor imajo štartno sekvenco. Ta skrbi za to, da se program začne izvajati s prvim prispelim UDP paketom. Na ta način se poskrbi, da strežnik nebi šele petega UDP paketa prebral in na to začel z delovanjem.

Naslednji varnostni ukrep preverja stanje varnostnega gumba (*Emergency stop*). V kolikor se ta pritisne, se na servo krmilniku že sam po sebi sproži ukaz za ustavitev motorjev ter vklop zavor. Na strežniku pa se to pozna tako, da se program zaustavi. V kolikor želimo nadaljevati z vodenjem, je potrebno program ponovno zagnat.

Zaganjanje ter uporaba strežnika je podrobneje opisana v dodatku A.

4 Vodenje robota PA-10

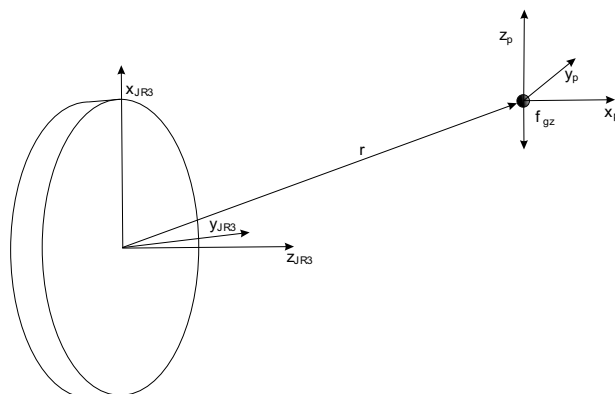
V prejšnjih poglavjih je bilo govora o pošiljanju referenčnih hitrosti in navorov na krmilnik servo motorjev robota PA-10. V poglavju, ki sledi pa bo govora o določanju teh krmilnih veličin v namene vodenja robota po prostoru. V tem poglavju bo natančneje opisano vodenje robota v zunanjih koordinatah. Opisan bo postopek kompenzacije vpliva težnosti prijemala na senzor sile JR3. Na koncu pa bosta opisana dva načina za krmiljenje robota v stiku z okolico, admitančno vodenje ter vodenje preko inverzne dinamike.

4.1 Kompenzacija mase prijemala ter merilnega odmika na senzorju sile JR3

Delovanje senzorja sile JR3 je bilo opisano v prejšnjih poglavjih. Senzor je pritrjen na vrh robota in meri napore ter sile, ki delujejo nanj v svojem koordinatnem sistemu. Homogeno transformacijsko matriko smo definirali v poglavju 2. Sile in napore na vrhu robota lahko preslikamo v koordinatni sistem baze preko matrike homogene transformacije

$$\mathbf{h}_b = \mathbf{T}_7^0 \mathbf{h}_e \quad (4.1)$$

Robot brez prijemala v praksi ni pretirano uporaben saj lahko opravlja le malo količino nalog. Recimo premikanje objektov s porivanjem. V kolikor pa želimo, da bo robot lahko izvajal tudi druge naloge, je potrebno nanj pritrditi



Slika 4.1: Koordinatni sistem senzorja sile ter koordinatni sistem teĹliĹÄŦa prijemala

primerno prijemalo. Ker pa imamo mi na vrhu robota najprej senzor, pomeni, da omenjeno prijemalo pritrdimo na senzor. Ker ima prijemalo svoje maso in teŹiŹŹe, bo na senzor delovala neka sila in navor kot posledica delovanja gravitacije. Kar pomeni, da bomo kljub temu, da senzor ni v kontaktu z okolico, merili neke sile. To pa ni dobrodoŹlo, saj Źelimo meriti le sile, ki se so posledica v kontakta z okolico, prijemalo je namreĹ del robota. Dodatno je potrebno Źe upoŹtevatı merilni odmik, ki se pojavi kot posledica lezenja uporovnih mostiĹkov.

ZapiŹemo izmerjeno silo (\mathbf{F}_s) ter izmerjen navor (\mathbf{M}_s) kot posledico teh prispevkov

$$\begin{aligned}\mathbf{f}_s &= \mathbf{f}_{env} + \mathbf{f}_{grav} + \mathbf{f}_{off} \\ \boldsymbol{\tau}_s &= \boldsymbol{\tau}_{env} + \boldsymbol{\tau}_{grav} + \boldsymbol{\tau}_{off}.\end{aligned}\tag{4.2}$$

Indeks *env* pomeni oznaĹuje silo in navor kot posledico okolice, indeks *grav* oznaĹuje silo in navor kot posledica vpliva gravitacije, indeks *off* pa oznaĹuje merilni odmik. Źelja je, izraĹunati *grav* in *off* tako, da ju lahko kompenziramo (odŹtejemo) in izpostavimo le *env*. Ko vrh robota ni v kontaktu z okolico mora veljati $\mathbf{F}_{env} = 0$ in $\mathbf{M}_{env} = 0$ in sledi:

$$\begin{aligned}\mathbf{f}_s &= \mathbf{f}_{grav} + \mathbf{f}_{off} \\ \boldsymbol{\tau}_s &= \boldsymbol{\tau}_{grav} + \boldsymbol{\tau}_{off}.\end{aligned}\tag{4.3}$$

\mathbf{f}_{grav} je odvisen od mase (m) prijemala ter orientacije vrha robota. Koordinatni sistem, ki ga postavimo v težišče je zaradi poenostavitve enako orientiran kot tisti, ki je v bazi robota. Orientacijo je mogoče zapisati s poznavanjem kinematičnega modela in bo v nadaljevanju označena z \mathbf{R} . Tako je mogoče določiti prispevek

$$\mathbf{f}_{grav} = \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \mathbf{R}\mathbf{f}_{gz}.\tag{4.4}$$

Avtorji v [17] pokažejo, da je mogoče izračunati \mathbf{f}_{grav} in \mathbf{f}_{off} na podlagi dveh meritev ($\mathbf{f}_{meas}^{(1)}$ in $\mathbf{f}_{meas}^{(2)}$). Zapišemo sistem enačb v matrični obliki

$$\begin{bmatrix} \mathbf{f}_{mes}^{(1)} \\ \mathbf{f}_{mes}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{(1)} & \mathbf{I} \\ \mathbf{R}^{(2)} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{gz} \\ \mathbf{f}_{off} \end{bmatrix}\tag{4.5}$$

in izpostavimo \mathbf{f}_{grav} in \mathbf{f}_{off}

$$\begin{bmatrix} \mathbf{f}_{gz} \\ \mathbf{f}_{off} \end{bmatrix} = \left(\begin{bmatrix} \mathbf{R}^{(1)} & \mathbf{I} \\ \mathbf{R}^{(2)} & \mathbf{I} \end{bmatrix}^T \begin{bmatrix} \mathbf{R}^{(1)} & \mathbf{I} \\ \mathbf{R}^{(2)} & \mathbf{I} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{R}^{(1)} & \mathbf{I} \\ \mathbf{R}^{(2)} & \mathbf{I} \end{bmatrix}^T \begin{bmatrix} \mathbf{f}_{meas}^{(1)} \\ \mathbf{f}_{meas}^{(2)} \end{bmatrix}.\tag{4.6}$$

Za navor pa je potrebno poznati tudi translacijski vektor od težišča do koordinatnega sistema senzorja, v nadaljevanju označen z \mathbf{r} . Prispevek navora zaradi vpliva gravitacije je tako

$$\boldsymbol{\tau}_{grav} = \mathbf{r} \times \mathbf{f}_{grav}.\tag{4.7}$$

Dodatno bomo tukaj definirali še $\hat{\mathbf{f}}_{grav}$, ki je matričen zapis vektorja \mathbf{f}_{grav} . Iz treh različnih meritev navorov ($\boldsymbol{\tau}_{meas}^{(1)}$, $\boldsymbol{\tau}_{meas}^{(2)}$ in $\boldsymbol{\tau}_{meas}^{(3)}$) in s tremi različnimi orientacijami ($\mathbf{R}^{(1)}$, $\mathbf{R}_{meas}^{(2)}$ in $\mathbf{R}_{meas}^{(3)}$) se zapiše sistem enačb

$$\begin{bmatrix} \boldsymbol{\tau}_{meas}^{(1)} \\ \boldsymbol{\tau}_{meas}^{(2)} \\ \boldsymbol{\tau}_{meas}^{(3)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}_{grav}^{(1)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(2)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(3)} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \boldsymbol{\tau}_{off} \end{bmatrix}. \quad (4.8)$$

Enačbo obrnemo in izpostavimo \mathbf{r} in $\boldsymbol{\tau}_{grav}$

$$\begin{bmatrix} \mathbf{r} \\ \boldsymbol{\tau}_{off} \end{bmatrix} = \left(\begin{bmatrix} \hat{\mathbf{f}}_{grav}^{(1)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(2)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(3)} & \mathbf{I} \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{f}}_{grav}^{(1)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(2)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(3)} & \mathbf{I} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{f}}_{grav}^{(1)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(2)} & \mathbf{I} \\ \hat{\mathbf{f}}_{grav}^{(3)} & \mathbf{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\tau}_{meas}^{(1)} \\ \boldsymbol{\tau}_{meas}^{(2)} \\ \boldsymbol{\tau}_{meas}^{(3)} \end{bmatrix}. \quad (4.9)$$

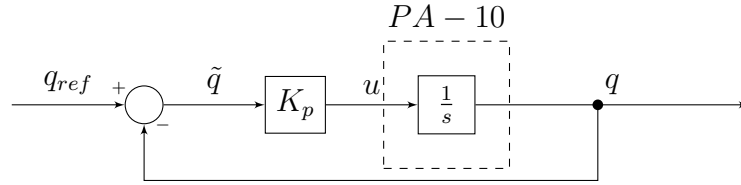
Bistveni trije podatki, ki smo jih s temi izračuni izpostavili so merilni odmik, masa prijemala ter njegovo težišče. Prednost tega postopka je, da imamo orientacijo senzorja v vseh merilnih pozicijah natančno določeno zaradi poznavanja inverzne kinematike. Hkrati pa lahko ta postopek popolnoma avtomatiziramo, kar smo avtorji tega dela tudi naredili. Postopek je opisan v dodatku ???. Masa ter težišče prijemala sta parametra, ki se jih identificira in sta časovno neodvisna. Zaradi efekta lezenja uporovnih lističev pa je merilni odmik časovno odvisen in ga je potrebno pogosteje izmeriti.

V nadaljevanju bo sila ter navor kot posledica interakcije z okolico zapisana v vektorju \mathbf{h}_e kot

$$\mathbf{h}_e = \begin{bmatrix} \mathbf{f}_{env} \\ \boldsymbol{\tau}_{env} \end{bmatrix}. \quad (4.10)$$

4.2 Vodenje preko inverzne kinematike

Vodenje robota preko vhodnih hitrostih v sklepih je relativno enostaven problem saj PI regulator v krmilniku poskrbi, da se vsak sklep vrti s hitrostjo čimbolj podobno referenčni. Iz tega vidika je prednost ta, da ne glede na konfiguracijo ter obremenitev robota, bo regulator vedno poskrbel čim boljše sledenje referenci. Vplivi gravitacije se kompenzirajo s integrirnim členov v regulatorju. V nadaljevanju bo opisano vodenje motorjev robota v zeleno lego oziroma vodenje robota



Slika 4.2: Povratna zanka za krmiljenje kota v sklepu

v notranjih koordinatah. Nato pa se bo iz tega še izpeljalo vodenje v zunanjih koordinatah.

4.2.1 Vodenje v notranjih koordinatah

Krmilnik servo motorjev nam v vsakem danem trenutku vrača trenutni kot v sklepu. Ker ima PI regulator visoka ojačanja ter visoko frekvenco, bomo za namene poenostavitve aproksimirali motor kot integrator $\frac{1}{s}$. Naj bo referenčni kot v sklepih označen kot \mathbf{q}_{ref} in naj bo dejanski kot označen kot \mathbf{q} . Razlika med njima je napaka v kotu

$$\tilde{\mathbf{q}} = \mathbf{q}_{ref} - \mathbf{q}. \quad (4.11)$$

S preprostim P regulatorjem lahko kote v sklepu robota vodimo s povratno zanko tako, kot je prikazano na shemi 4.2. Ojačanje regulatorja bo označeno s \mathbf{K}_p in je diagonalna saj ima vsak sklep svoje ojačanje neodvisno od ostalih. S povečevanjem ojačanja regulatorja \mathbf{K} vplivamo na dinamiko odziva [18].

$$\mathbf{u} = \mathbf{K}_p \tilde{\mathbf{q}}. \quad (4.12)$$

Iz blokovnega diagrama lahko zapišemo prenosno funkcijo sistema $P(s)$ in analiziramo stabilnost:

$$P(s) = \frac{K \frac{1}{s}}{1 + K \frac{1}{s}} = \frac{K}{s + K} \quad (4.13)$$

Pogoj stabilnosti je, da vsi poli sistema ležijo na desni strani s ravnine. V tem primeru je tako pogoj le, da je $K > 0$. Iz tega bi sledilo, da lahko ojačanje K povečujemo v neskončnost. Vendar pa je potrebno tukaj poudariti, da pri previsokih ojačanjih lahko pride do nestabilnosti, ki pa jih z obravnavanjem robota kot integratorja, ne moremo predvideti. Zato je smiselno, da se z ojačanji ne pretirava. Lahko pa v namene stabilnosti vnesemo še signal dušenja $\mathbf{K}_d \dot{\mathbf{q}}$. Matrika \mathbf{K}_d je tako kot \mathbf{K}_p diagonalna. Regulirni signal sedaj zapišemo kot

$$\mathbf{u} = \mathbf{K}_p \tilde{\mathbf{q}} - \mathbf{K}_d \dot{\mathbf{q}}. \quad (4.14)$$

Vodenje robota po notranjih koordinatah je sredstvo za realizacijo vodenja v zunanjih koordinatah saj se samo po sebi redko uporablja. Primer uporabe vodenja v notranjih koordinatah je recimo, če se želi robota nastaviti na neko konfiguracijo.

4.2.2 Vodenje v zunajjih koordinatah

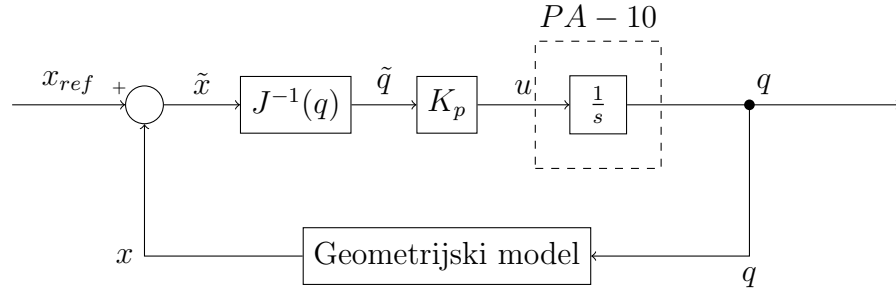
Kinematični model robota in problem inverzne kinematike je bil opisan v 2.1. Predstavljen je bil problem inverzne kinematike za redundantni sistem ter pseudo-inverz Jacobijeve matrike. V nadaljevanju bo opisano vodenje robotov v zunanjih koordinatah v splošnem primeru, ko velja $n = m$.

Naj bo \mathbf{x}_{ref} vektor referenčne lege vrha robota in naj bo \mathbf{x} trenutna lega vrha robota. Napaka je tako

$$\tilde{\mathbf{x}} = \mathbf{x}_{ref} - \mathbf{x} \quad (4.15)$$

Enačba direktne kinematike nam govori o odnosu med hitrostjo vrha robota ter hitrostjo v sklepih robota

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}. \quad (4.16)$$



Slika 4.3: Bločna shema krmiljenja referenčne lege vrha robota

To enačbo je mogoče razumeti tudi kot zvezo med spremembo položaja vrha ter sklepov robota [11]. Zato je mogoče to enačbo uporabiti za zapis relacije med napako vrha ter potrebnim odmikov v sklepih preko preko enačbe za inverzno kinematiko

$$\tilde{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\tilde{\mathbf{x}} \quad (4.17)$$

Z vstavitvijo enačbe 4.16 v 4.12 dobimo P regulator lege vrha robota. Sedaj je mogoče regulirni signal, ki krmili hitrost v sklepih robota tako, da bo vrh sledil referenčni legi

$$\mathbf{u} = \mathbf{K}_p \mathbf{J}^{-1}(\mathbf{q})(\mathbf{x}_{ref} - \mathbf{x}). \quad (4.18)$$

Avtor v [11] 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 \mathbf{K}_p .

4.3 Admitančno vodenje

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 impedanco. Pojem

mehanske impedance izvira iz analogije električne impedance. Predstavlja pa razmerje med silo (F) in hitrostjo (v)

$$Z(s) = \frac{F}{v} = ms + b + \frac{k}{s} \quad (4.19)$$

oziroma razmerje med silo in pozicijo (x)

$$Z(s) = \frac{F}{x} = ms^2 + bs + k \quad (4.20)$$

. V zgornjih enačbah predstavlja m maso, b viskozno dušenje, k togost [16].

Vodenje na podlagi admitance oziroma inverza impedance izhaja iz tega, da je krmilna veličina hitrost, krmiljena veličina pa sila.

4.3.1 Uporaba 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 ?? 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:

$$\tilde{\mathbf{h}} = \mathbf{h}_{ref} - \mathbf{h} \quad (4.21)$$

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}^* .

$$\mathbf{x}_{ref}^* = \mathbf{x}_{ref} + \mathbf{K}_h \tilde{\mathbf{h}}_{JR3} \quad (4.22)$$

Z vstavitvijo enačbe 4.22 v 4.18 je mogoče zapisati celotno regulacijsko enačbo za vodenje robota v zunanjih koordinatah z upoštevanjem delovanja zunanjih sil na vrh robota.

$$\mathbf{u} = \mathbf{K}_p \mathbf{J}^{-1}(\mathbf{q})(\mathbf{x}_{ref} + \mathbf{K}_h \tilde{\mathbf{h}}_{JR3} - \mathbf{x}) \quad (4.23)$$

4.4 Implementacija

Avtorji so za implementacijo na robotskem mehanizmu PA-10 uporabili strežnik opisan v poglavju ?? ter programsko okolje MATLAB[®] SIMULINK[®]. Program za vodenje robota je bil v celoti zgrajen v okolju SIMULINK[®]. 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 4.2.

Avtorji tega dela so pripravili SIMULINK[®] 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 $F = \Delta x k$ 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.

4.5 Rezultati

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

5 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 [8] 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 3.2.

5.1 Teorija

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

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_f\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (5.1)$$

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.

5.1.1 Vodenje v notranjih koordinatah

V poglavju 4.2.1 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 5.1. 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 5.1 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:

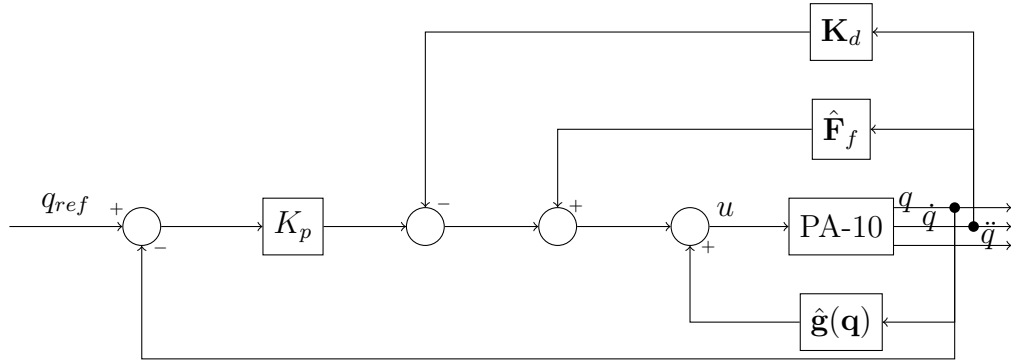
$$\ddot{\mathbf{q}}_{ref} = \mathbf{K}_p(\mathbf{q}_{ref} - \mathbf{q}) - \mathbf{K}_d(\dot{\mathbf{q}}) \quad (5.2)$$

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:

$$\mathbf{u} = \hat{\mathbf{B}}(\mathbf{q})(\mathbf{K}_p(\mathbf{q}_{ref} - \mathbf{q}) - \mathbf{K}_d(\dot{\mathbf{q}})) + \hat{\mathbf{F}}_f\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) \quad (5.3)$$

Shemo takega vodenja prikazuje shema 5.1



Slika 5.1: Bločna shema krmiljenja referenčne lege vrha robota z inverzno dinamiko

5.2 Implementacija

Postopek implementacije je bil v tem primeru podoben tistemu opisanemu v poglavju 4.4. Tudi tokrat so avtorji uporabili strežnik opisan v poglavju?? 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 bit 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.

5.3 Rezultati

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

6 Simulink knjižice

Avtorji tega dela so za visoko nivojsko vodenje uporabljali programski paket MATLAB® SIMULINK®. Ker je želja avtorjev, da se raziskovalno in razvojno delo na robotu PA-10 v prihodnosti nadaljuje, so nastale SIMULINK® 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® shemo.

6.1 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.

6.2 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.

6.3 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 4.2.2.

6.4 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.

7 Zaključek

Literatura

- [1] B. Siciliano, L. Sciavicco, L. Villani in G. Oriolo, *Robotics, Modelling, Planning and Control*. London: Springer, 2009.
- [2] Mitsubishi Heavy Industries, *Portable General Purpose Intelligent Arm - Operating Manual*, rev. 1 izd.
- [3] M. H. I. Ltd., “Mitsubishi clean room robot,” *Seminarska naloga*, 2003.
- [4] J. J. Craig, *Introduction to Robotics*. Z.D.A.: Pearson Education International, 2005.
- [5] N. D. Vuong in M. H. J. Ang, “Dynamic model identification for industrial robots,” *Acta Polytechnica Hungarica*, 2009.
- [6] R. van der Aalst, M. H. J. Ang in H. Nijmeijer, “Dynamic identification of a mitsubishi pa-10 robotic manipulator,” *ICAR*, 1999.
- [7] R. Jamisola, M. J. Ang, T. M. Lim, O. Khatib in S. Y. Lim, “Dynamics identification and control of an industrial robot,” *ICAR*, 1999.
- [8] T. Petrič in M. Munih, “Kompenzacija trenja in gravitacije na robotu mitsubishi pa-10,” *Seminarska naloga*, 2010.
- [9] J. Lenarčič in T. Bajd, *Robotski mehanizmi*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2003.
- [10] A. M. Almassri, W. Z. W. Hasan, S. A. Ahmad, A. J. Ishak, A. M. Ghazali, D. N. Talib in C. Wada, “Pressure sensor: State of the art, design, and

- application for robotic hand,” *Hindawi Publishing Corporation, Jurnal of Sensors*, 2014.
- [11] M. Mihelj, T. Bajd in M. Munih, *Vodenje robotov*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2011.
- [12] S. D. Eppinger in W. P. Seering, “Three dynamic problems in robot force control,” *IEEE Transactions on Robotics and Automation*, str. 751 – 758, 1992.
- [13] Contemporary Controls, *ARCNET Tutorial*.
- [14] I. JR3, “JR3, Installation manual for force - torque sensors with external electronics.” Dosegljivo: <http://www.jr3.com/product-manuals.html>. [Dostopano: 6. 17. 2015].
- [15] I. JR3, “JR3, DPS-based force sensor receivers, software and installation manual.” Dosegljivo: <http://www.jr3.com/product-manuals.html>. [Dostopano: 6. 17. 2015].
- [16] M. Mihelj, *Vodenje robotov*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2011.
- [17] D. Omršen in B. Nemec, “Meritev gibanja kolena z industrijskim robotom - avtomatska kompenzacija gravitacije prijemala,” *Strojniški vestnik*, str. 87 – 97, 2002.
- [18] B. Zupančič, *Teorija regulacij*. Ljubljana: Univerza v Ljubljani, Založba FE in FRI.

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]"  
}
```