

Krmiljenje manipulatorjev na osnovi strojnega vida

Študent: Samo Simončič

Mentor: Primož Potočnik

KAZALO:

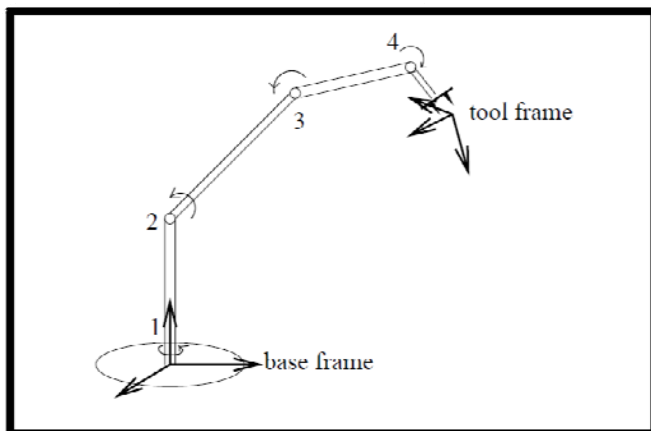
1	UVOD.....	3
2	KINEMATIKA MANIPULATORJA.....	3
3	KALIBRACIJA KAMERE	5
4	DEFINICIJA PROBLEMA.....	7
5	OSNOVNI PRINCIP	8
5.1	REZULTATI EKSPERIMENTA	8
6	PROSTI PRINCIP	9
6.1	PRAVILA UČENJA	12
6.2	POVPREČENJE IZHODNE SPREMENLJIVKE.....	13
6.3	TRENIRANJE NEVRONSKE MREŽE	14
6.4	REZULTATI EKSPERIMENTA	14
7	HIBRIDNI PRINCIP	15
7.1	REZULTATI EKSPERIMENTA	16
8	ZAKLJUČEK.....	17
9	LITERATURA	18

1 UVOD

V zadnjih letih se je strojni vid zelo uveljavil na področju robotike z namenom avtonomnega delovanja manipulatorjev in drugih robotskih gradnikov. V seminarju bom na kratko predstavil nekaj primerov krmiljenja manipulatorjev na osnovi kinematike, kjer se prepletajo znanja strojnega vida, robotike in nevronske mreže. S pomočjo vizualnih informacij, katere pridobimo z različnimi senzorji, določimo dejansko in željeno lego (orientacija in pozicija) manipulatorja. V robotiki se za vizualne informacije pogosto uporabljajo kamere oziroma sistemi kamer na podlagi katerih določimo lego prijemala manipulatorja in željeno ciljno točko. Na podlagi znanih koordinat lahko določimo inverzno kinematiko manipulatorja, ki določa vrednosti spremenljivk sklepov za izbrano točko v prostoru. V primeru uporabe kamer za povratno zvezo manipulatorja je potrebno narediti kalibracijo kamer. S tem določimo aproksimacijski model kamere, ki preslika koordinate točk na kameri v koordinate točk v delovnem prostoru manipulatorja. Na področju robotike so v zadnjem desetletju zelo popularni algoritmi z uporabo nevronske mreže, ki omogočajo natančno določevanje kinematike manipulatorjev. Koncept določevanja kinematike s pomočjo nevronske mreže je predstavljen v seminarju, kjer sem se osredotočil na uporabo »Kohonen's self-organizing« (KSOM) arhitekture, ki spada v skupino »Self-organizing networks«. V seminarski nalogi se bom omejil le na področje kinematike manipulatorjev, ki določa povezavo med spremenljivkami sklepov in željeno točko v delovnem prostoru.

Poglavje KINEMATIKA MANIPULATORJA opisuje osnovne pojme za določevanje direktne in inverzne kinematike za izbrani manipulator. Poglavje KALIBRACIJA KAMERE se navezuje na določitev modela, ki povezuje točke med koordinatnim sistemom kamere in delovnim prostorom. Jedro seminarja opisuje uporabo nevronske mreže, ki so predstavljene od četrtega poglavja naprej (DEFINICIJA NALOGE). V zadnjem poglavju ZAKLJUČEK so kritično ovrednoteni rezultati, ki sem jih predstavil skozi vsa poglavja.

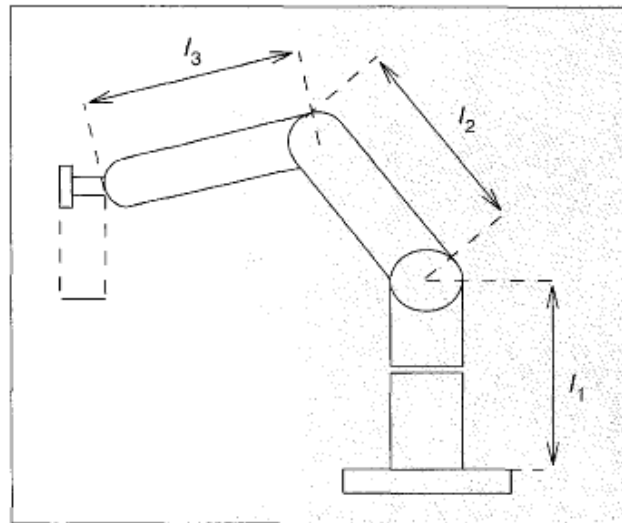
2 KINEMATIKA MANIPULATORJA



i

Za razumevanje osnovnih pojmov na področju robotike bom na kratko opisal osnovne pojme, ki so potrebni za nadaljnje razumevanje. Na sliki 1 je prikazan manipulator s štirimi prostostnimi stopnjami (4 DOF). Vsi štiri sklepi manipulatorja so rotacijski in vsak izmed njih predstavlja eno spremenljivko θ_i ($\theta_1, \theta_2, \theta_3, \theta_4$). Prijemalo (»tool frame«) je namenjeno prijemanju objektov v delovnem prostoru manipulatorja. Točko v delovnem

prostoru lahko opišemo s koordinatnim sistemom delovnega prostora ali s spremenljivkam sklepov manipulatorja. V primeru, da poznamo spremenljivke sklepov θ_i ($\theta_1, \theta_2, \theta_3, \theta_4$) lahko s pomočjo direktne kinematike določimo pozicijo in orientacijo manipulatorja. Če poznamo koordinate točke prijemala (v delovnem prostoru) lahko s pomočjo inverzne kinematike določimo spremenljivke sklepov. Rešitve inverzne kinematike v primerjavi z direktno kinematiko ni enostavno določiti, saj imamo opravka (pri več kot treh prostostnih stopnjah manipulatorja) z več možnim rešitvami spremenljivk sklepov. Za lažje razumevanje inverzne in direktne kinematike si pogledjmo primer določitve kinematike za manipulator s tremi prostostnimi stopnjami, kjer imamo vse tri sklepe rotacijske (slika 2).



Slika 2: Geometrijski model manipulatorja s tremi prostostnimi stopnjami

Analitična določitev direktne kinematike ima obliko:

$$Z = l_2 \sin \theta_2 + l_3 \sin \theta_3 + l_1 \quad (1)$$

$$X = R \cos \theta_1 \quad (2)$$

$$Y = R \sin \theta_1 \quad (3)$$

kjer je $R = l_2 \cos \theta_2 + l_3 \cos \theta_3 + t$ in X, Y, Z so koordinate točke prijemala glede na delovni prostor manipulatorja. Prav tako lahko analitično določimo inverzno kinematiko, ki bo v nadaljevanju potrebna pri kalibraciji kamere. Za željeno točko (X, Y, Z) prijemala lahko določimo spremenljivke sklepov manipulatorja θ_i ($\theta_1, \theta_2, \theta_3$) kot:

$$\theta_1 = \tan^{-1} \left(\frac{Y}{X} \right) \quad (4)$$

$$\theta_2 = \cos^{-1} x \quad (5)$$

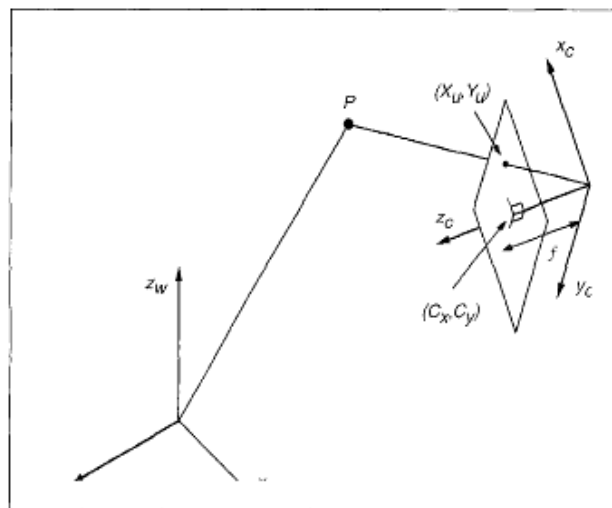
$$\theta_1 = \sin^{-1} \left(\frac{b - l_2 \sin \theta_2}{l_3} \right) \quad (6)$$

kjer je x rešitev $Ax^2 + Bx + C = 0$, $a = \sqrt{X^2 + Y^2} - t$, $b = Z - l_1$, $K = b^2 + l_2^2 - l_3^2 + a^2$, $A = 4a^2 l_2^2 + 4b^2 l_2^2$, $B = -4al_2 K$ in $C = K^2 - 4b^2 l_2^2$.

Če primerjamo rešitvi za direktno in inverzno kinematiko lahko ugotovimo, da so rešitve inverzne kinematike veliko bolj zahtevnejše, čeprav imamo opravka z manipulatorjem, ki ima samo tri prostostne stopnje. V naslednjih dveh poglavjih si bomo ogledali kalibracijo kamere in določitev inverzne kinematike z uporabo nevronske mreže.

3 KALIBRACIJA KAMERE

Kot sem omenil v uvodnih besedah se kalibracija kamere navezuje na določitev modela kamere, ki popisuje pretvorbo med točkami na slikovni ravnini kamere in točkami v delovnem prostoru krmiljenega objekta (v našem primeru imamo krmilni objekt manipulator). Eden izmed najbolj popularnih algoritmov, ki popisuje kalibracijo kamer je Tsai's algoritem, ki ga bom na kratko predstavil v nadaljevanju poglavja. Na sliki 3 je prikazan model za kalibracijo kamere, ki ga je potrebno matematično popisati.



Slika 3: Geometrijski model kamere

Center kamere je označen z (x_c, y_c, z_c) , kjer se z_c os ujema z optično osjo kamere. Slika kamere je paralelna na (x_c, y_c) ravnino in je premaknjena za razdaljo f (gorišče) od izhodišča koordinatnega sistema. Točko P v prostoru opišemo v koordinatah delovnega prostora (x_w, y_w, z_w) . Relacijo med točko P v koordinatnem sistemu delovnega prostora in v slikovnih koordinatah (X_f, Y_f) popišemo s sekvenco transformacij. Prva transformacija je med (x_w, y_w, z_w) in (x_c, y_c, z_c) koordinatam, ki je definirana:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \quad (7)$$

kjer $R = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix}$ je 3x3 rotacijska matrika, ki opisuje orientacijo kamere v koordinatnem sistemu (x_w, y_w, z_w) in $\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}$ je translacija kamere v koordinatnem sistemu (x_w, y_w, z_w) .

Naslednja transformacija je centralno perspektivna projekcija, ki se izvaja med koordinatam kamere (x_c, y_c, z_c) in sliko kamere (X_d, Y_d) . Transformacijo opišemo z:

$$X_u = f \frac{x_c}{z_c} \quad (8)$$

$$Y_u = f \frac{y_c}{z_c} \quad (9)$$

kjer je f goriščna razdalja kamere. V primeru, da imamo zaradi leč prisotno distorzijo moramo še dodatno upoštevati naslednjo transformacijo:

$$X_u = X_u(1 + \alpha \rho^2) \quad (10)$$

$$Y_u = Y_u(1 + \alpha \rho^2) \quad (11)$$

$$\rho = \sqrt{X_d^2 + Y_d^2}$$

kjer je α koeficient radialne distorzije leče. (X_u, Y_u) sta koordinati, ki upoštevata radialno distorzijo po »Cardan's cubic« metodi. Zadnja transformacija je med koordinatam (X_u, Y_u) in slikovnim koordinatam (X_f, Y_f) in je definirana:

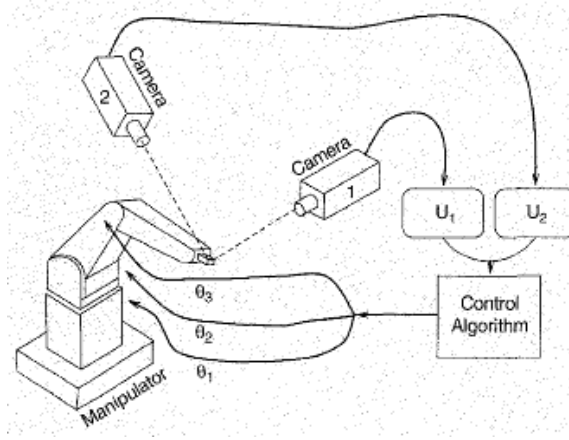
$$X_f = d_x^{-1} X_D s_x + C_x \quad (12)$$

$$Y_f = d_y^{-1} Y_D + C_y \quad (13)$$

kjer sta C_x in C_y koordinati in z_c os seka sliko kamere ter d_x in d_y velikosti slikovnih pik v x in y smeri. Če združimo vse transformacije ugotovimo, da imamo minimalno enajst neznanek, za kar potrebujemo minimalno 11 točk, da določimo neznanke. Če uporabimo več točk kot jih je potrebno (minimalno enajst točk) dobimo s tem bolj natančno kalibracijo.

Kalibracija kamere nam omogoča, da za koordinate slikovnih pik ugotovimo, kje se dane točke nahajajo v delovnem prostoru. Pomembno je omeniti, da za točko v delovnem prostoru s koordinatam (x, y, z) potrebujemo dve kameri, da lahko določimo njeno lego. Ena sama kamera nam popisuje le dve koordinatni dimenziji, kar pomeni, da je nujno potrebno uporabiti dve kameri. Na sliki 4 je prikazan primer takšnega modela z manipulatorjem, ki ima tri prostostne

stopnje. S pomočjo dveh kamer določimo lokacijo objekta v njihovih slikovnih ravninah. Ko imamo točki na sliki obeh kamer lahko s pomočjo kalibracije določimo koordinate objekta glede na delovni prostor manipulatorja ter v nadaljevanju z inverzno kinematiko določimo spremenljivke sklepov danega manipulatorja.



Slika 4: Manipulator z dvema kamerama in ustreznim krmilnikom

4 DEFINICIJA PROBLEMA

Kot je bilo omenjeno v prejšnjem poglavju za določitev lokacije objekta v delovnem prostoru potrebujemo najmanj dve kameri skupaj s poznano kinematiko manipulatorja. V našem primeru si bomo pogledali primer krmiljenja manipulatorja z dvema kamerama, ki je prikazan na sliki 4. S pomočjo dveh kamer določimo koordinate prijema manipulatorja in koordinate željenega objekta, katere označimo z dvodimenzionalnim vektorjem (u_1, u_2) za prvo kamero in (u_3, u_4) za drugo kamero. V veliko primerih zapišemo vse skupaj v štiri dimenzionalni vektor $\vec{u}_{target} = (u_1, u_2, u_3, u_4)$. Pozicija manipulatorja je določena s tremi spremenljivkami sklepov $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$. Za določitev pozicije prijema je potrebno poznati transformacijo $\Theta(\vec{u}_{target})$, ki preslika vektor \vec{u}_{target} v spremenljivke sklepov $\vec{\theta} (\theta_1, \theta_2, \theta_3)$ manipulatorja. Transformacijo $\Theta(\vec{u}_{target})$ za manipulatorje s tremi prostostnimi stopnjami lahko rešimo na več načinov:

- **Osnovni princip (»Model-base approach«):** Določimo model manipulatorja in model kamere ter izračunamo spremenljivke sklepov z uporabo vizualnih informacij. Model manipulatorja za tri prostostne stopnje je razložen v poglavju KINEMATIKA MANIPULATORJA. Model kamere pa v poglavju KALIBRACIJA KAMERE. V našem primeru je potrebno narediti kalibracijo za dve kameri.
- **Prosti princip (»Model-free approach«):** Manipulator prilagaja spremenljivke sklepov $\vec{\theta}$ s tako imenovanim učenjem v realnem času (»on-line«). Princip je zelo fleksibilen in natančen v primerjavi z Osnovnim principom. Uporaba nevronske mreže (»KOHONEN'S SELF-ORGANIZING« arhitekture).

- **Hibridni princip (»Hybrid approach«):** Treniranje Hibridnega modela je sestavljeno iz treniranja Osnovnega in treniranja Prostega modela. Z osnovnim modelom izvajamo učenje brez aktivne uporabe manipulatorja (»off-line«) s tem zmanjšamo čas učenja. Na podlagi Osnovnega modela začnemo izvajati fino učenje na tistih območjih, kjer je potrebno (velika napaka v pozicij manipulatorja...). Fino treniranje se izvaja z aktivno uporabo manipulatorja (»on-line«). Pri finem treniranju uporabljamo Prosti princip, ki omogoča doseganje natančnejšega pozicioniranja manipulatorja kot pri uporabi samo Osnovnega modela.

Vse tri principe bom na kratko opisal v spodnjih poglavjih ter jih kritično ovrednotil.

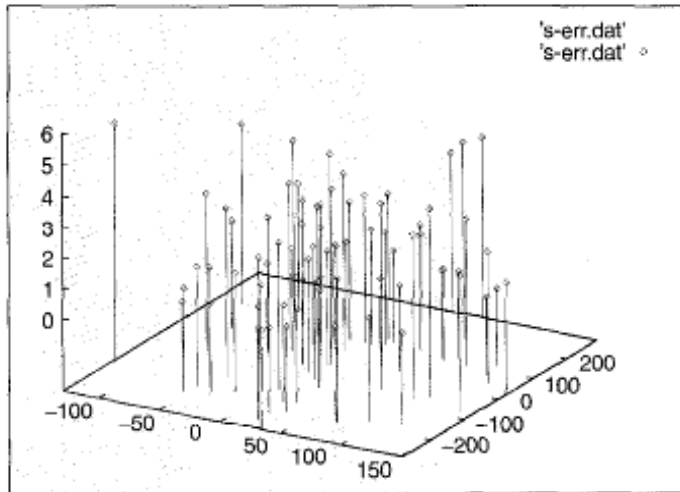
5 OSNOVNI PRINCIP

V primeru, da poznamo model celotnega sistema (model manipulatorja in model kamere) lahko določimo transformacijo med \vec{u}_{target} in $\vec{\theta}$. Iz koordinat dveh kamer \vec{u}_{target} lahko določimo enačbo žarkov, ki določajo željeno točko v prostoru. Na podlagi slike 3 potuje vsak žarek skozi točke (X_d, Y_d, f) in ga definira transformacija (10, 11). Transformacijo koordinatnega sistema kamere glede na koordinatni sistem delovnega prostora definira enačba (7). Iz enačb (10, 11, 7) tako dobimo enačbi za dva žarka (s črtkanimi črtami označena na sliki 4), ki se sekata v željeni točki. Ko poznamo željeno točko v koordinatnem sistemu delovnega prostora lahko določimo inverzno kinematiko manipulatorja. Če uporabimo enačbe (4) (5) (6) lahko določimo spremenljivke kotov manipulatorja $(\theta_1, \theta_2, \theta_3)$. Tako se bo manipulator premaknil k željeni točki oziroma objektu v delovnem prostoru.

Prednost takšnega algoritma je, da na začetku naredimo kalibracijo (določimo model kamere) in med izvajanjem algoritma ni potrebna ponovna kalibracija. Prav tako ni potrebnega nobenega učenja za doseganje željenih točk v prostoru. Slabost takšnega algoritma se najbolj pokaže v primeru slabega modela kamere, kar je ponavadi glavni vzrok za napako v poziciji manipulatorja.

5.1 REZULTATI EKSPERIMENTA

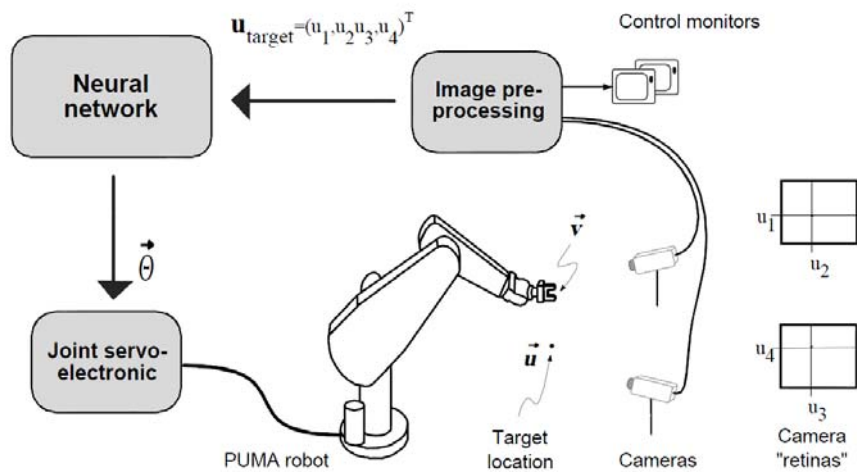
Na sliki 5 je prikazana napaka pozicije manipulatorja pri uporabi Osnovnega principa. Lepo je razvidno, da je največja napaka v okolici meje delovnega prostora. Bolj ko se bližujemo središču delovnega prostora manjša je napaka, kar lahko pojasnimo s tem, da imamo vpliv radialne distorzije četudi smo jo upoštevali v modelu kamere.



Slika 5: Napaka pozicije pri uporabi Osnovnega principa projicirana na XY ravnino

6 PROSTI PRINCIP

Podobno kot v Osnovnem principu imamo tudi v Prostem principu gradnike sistema, ki so: manipulator, dve kameri in krmilnik. Razlika je le v tem, da imamo krmilnik na osnovi nevronske mreže, ki je podrobneje predstavljen v nadaljevanju poglavja. Shema celotnega sistema je prikazana na sliki 6.



Slika 6: Celotna shema sistema za krmiljenje manipulatorja PUMA

Cilj Kohonen's Self-Organizing arhitekture je pravilno določevanje lege prijemala z uporabo le dveh kamer. Nobenega zunanega učitelja ali katerih drugih informacij o geometriji manipulatorja ni na razpolago razen dveh kamer, ki določata vektor \vec{u}_{target} . Tako kot pri

Osnovnem principu tudi pri Prostem principu na podlagi dveh kamer določimo koordinate željene točke v delovnem prostoru, katere zapišemo v skupen vektor $\vec{u}_{target} = (u_1, u_2, u_3, u_4)$. Vektorji \vec{u}_{target} ($\vec{u}_{target} \subseteq U$) določajo vstopne spremenljivke v nevronske mreže, ki predstavljajo krmilnik sistema. Za določevanje lege prijemala mora sistem poznati transformacijo $\vec{\Theta}(\vec{u}_{target})$ med spremenljivkami sklepov in vstopno veličino. Transformacija je odvisna od:

- geometrijskega modela manipulatorja,
- relativno pozicijo manipulatorja glede na pozicijo kamere,
- optičnih lastnosti kamere.

V obeh primerih (Osnovni in Prosti princip) imamo manipulator s tremi prostostnimi stopnjami, kar pomeni, da nimamo opravka s čezmernim številom prostostnih stopenj manipulatorja.

Osnovna ideja uporabe Kohonen's Self-Organizing arhitekture je, da diskretiziramo vstopno množico U v množico N nepovezanih celic F_μ , kjer je $\mu \in \{1, 2, 3, \dots, N\}$. Vsaka celica aproksimira vektor spremenljivk sklepov $\vec{\theta}$. Vsaki celici F_μ pripada en nevron, ki shranjuje tri pomembne spremenljivke:

- težnostni vektor \vec{w}_μ ,
- izhodni vektor oziroma vektor spremenljivk sklepov $\vec{\theta}_\mu$,
- Jacobian-ovo matriko \vec{A}_μ .

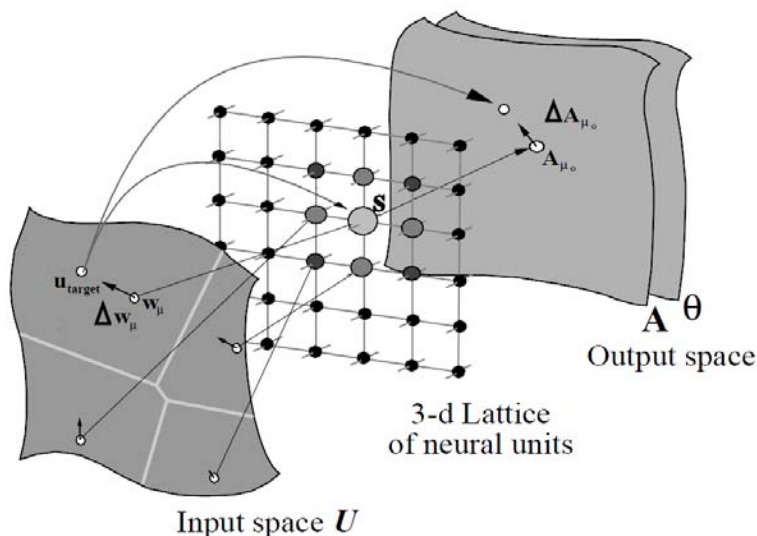
Vse tri spremenljivke skupaj tvorijo linearni Taylor-jev izraz za $\vec{\Theta}(\vec{u}_{target})$ znotraj F_μ celice, kar lahko zapišemo:

$$\vec{\Theta}(\vec{u}_{target}) = \vec{\theta}_\mu + \vec{A}_\mu(\vec{u}_{target} - \vec{w}_\mu). \quad (14)$$

$\vec{\theta}_\mu$ je člen ničtega reda, \vec{A}_μ (3x4 matrika) se imenuje Jacobian-ova matrika, ki označuje člen prvega reda.

Za vsak korak treninga določimo točko v delovnem prostoru in vektor \vec{u}_{target} , katerega dobimo iz dveh kamer. Nevron, ki ima težnostni vektor \vec{w}_μ najbližji vstopnemu vektorju \vec{u}_{target} označimo z μ_0 in ga imenujemo zmagovalni nevron. Postopek je grafično prikazan na sliki 7, kjer je nevron z najbližjim težnostnim vektorjem \vec{w}_μ glede na \vec{u}_{target} označen z μ_0 (zmagovalni nevron). Na podlagi izbranega nevrona in njegovih sosednjih nevronov lahko določimo njegov vektor spremenljivk sklepov $\vec{\theta}_0^{out}$ na podlagi enačbe 14. V nadaljevanju bo razvidno, da zaradi hitrejše konvergence ne spreminjamo samo zmagovalni nevron temveč tudi nevrone v okolici zmagovalnega nevrona.

Gibanje manipulatorja do željene točke v delovnem prostoru je sestavljeno iz dveh gibanj. Prvi korak je grobi premik manipulatorja, kjer se sklepi manipulatorja premaknejo za $\vec{\theta}_0^{out}$. Indeks 0 spremenljivke $\vec{\theta}_0^{out}$ pove, da je bila enkrat uporabljena vizualna informacija (kameri). Premik sklepov manipulatorja za $\vec{\theta}_0^{out}$ prijemalo premakne v bližino željene točke v delovnem prostoru. Novo lokacijo prijemala s pomočjo dveh kamer označimo z \vec{v}_0 . Drugi korak je fini premik manipulatorja, kjer se premaknejo za $\vec{\theta}_1^{out}$, ki je odvisen od razlike med željeno točko \vec{u}_{target} in dejansko točko manipulatorja \vec{v}_0 ($\vec{u}_{target} - \vec{v}_0$). Razliki ($\vec{u}_{target} - \vec{v}_0$) moramo pomnožiti z Jacobian-ovo matriko, da pretvorimo premike v ustrezne spremenljivke sklepov manipulatorja.



Slika 7: Shematični prikaz Kohonen's Self-Organizing arhitekture. Zmagovalni nevron je označen s črko S v izbranem nivoju mreže. Poleg zmagovalnega nevrona imajo vpliv na izhodni prostor tudi nevрони, ki so v okolici le tega (prikazani z različnimi sivinami).

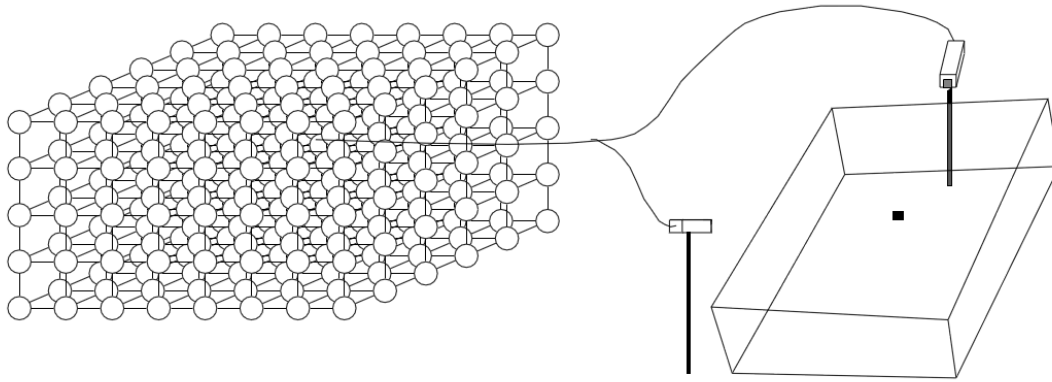
Po premiku sklepov manipulatorja za $\vec{\theta}_1^{out}$ se prijemalo premakne v bližino željene točke. Ponovno bi lahko izvedli fini premik $\vec{\theta}_2^{out}$ in v nadaljevanju $\vec{\theta}_3^{out}$... V praksi zadostujeta grobi premik $\vec{\theta}_0^{out}$ in prvi fini premik $\vec{\theta}_1^{out}$. V primeru, da vzamemo večje število finih premikov zmanjšamo napako med dejansko in željeno pozicijo manipulatorja. Slabost večjega števila finih premikov se kaže v zelo dolgem učenju nevronske mreže, kar je opisano v naslednjem podglavju.

6.1 PRAVILA UČENJA

Kot je bilo že omenjeno pravila učenja ne vplivajo le na zmagovalni nevron temveč tudi na njegove sosednje nevrone, kot je prikazano na sliki 7. V nadaljevanju so razložena pravila učenja za Self-Organizing Feature Map arhitekturo skupaj z uporabo Kohonen's algoritma. Za vstopni vektor \vec{u}_{target} se izbere najbližji težnostni vektor \vec{w}_s kateremu pripada nevron S. Najbližji težnostni vektor \vec{w}_s je izbran na podlagi Euclidean-ove razdalje med vstopnim in težnostnim vektorjem. Euclidean-ovo razdaljo lahko zapišemo:

$$\|w_s - \vec{u}_{target}\| = \min_{\nu\mu} \|w_\mu - \vec{u}_{target}\| \quad (15)$$

Dimenzija oziroma velikost mreže, v kateri so razporejeni nevrone je odvisna od velikosti delovnega prostora manipulatorja. Primer 3D mreže z razporeditvijo nevronov v delovnem prostoru je prikazan na sliki 8.



Slika 8: Primer 3D mreže za Kohonen's algoritem

Za zmagovalni nevron in najbližje sosednje nevrone za vsak korak treninga uporabimo pravila učenja, ki določajo adaptivno spreminjanje vektorjev w_μ , θ_μ in A_μ po naslednjih enačbah:

$$\vec{w}_\mu = \vec{w}_\mu + \varepsilon h_{\mu s} (\vec{u}_{target} - \vec{w}_\mu) \quad (16)$$

$$\vec{\theta}_\mu = \vec{\theta}_\mu + \varepsilon h_{\mu s} \Delta \vec{\theta}_\mu \quad (17)$$

$$\vec{A}_\mu = \vec{A}_\mu + \varepsilon h_{\mu s} \Delta \vec{A}_\mu \quad (18)$$

kjer faktorja učenja $\varepsilon \in [0,1]$ in $\varepsilon' \in [0,1]$ definirata spreminjanje spremenljivk w_μ , θ_μ in A_μ za vsak korak treninga. Spremenljivki $h_{\mu s}$ in $h'_{\mu s}$ sta Gaussian-ovi funkciji, kateri definirata vpliv bližnjih nevronov glede na zmagovalnega.

$$h_{\mu s} = \exp\left(-\frac{\|\gamma-s\|^2}{2\sigma}\right) \quad (19)$$

$$h'_{\mu s} = \exp\left(-\frac{\|\gamma-s\|^2}{2\sigma}\right) \quad (20)$$

kjer spremenljivka σ definira širino Gaussian-ove funkcije in s tem število sosednjih nevronov okoli zmagovalnega nevrona. $\|\gamma - s\|$ je razdalja med nevronom γ in zmagovalnim nevronom s .

Če želimo izračunati enačbi 17 in 18 moram poznati spremembo vektorja sklepov $\Delta\vec{\theta}_\mu$ in spremembo Jacobian-ove matrike $\Delta\vec{A}_\mu$ za vsak korak treninga. Spremenljivko $\Delta\vec{A}_\mu$ določimo s pomočjo pravila korekcije napake za Widrow-Hoff-ov model, ki je definiran:

$$\Delta\vec{A}_\mu = \|\Delta\vec{v}_{01}\|^{-2}(\Delta\vec{\theta}_{01}^{out} - \vec{A}_\mu\Delta\vec{v}_{01})\Delta\vec{v}_{01}^T$$

kjer je $\Delta\vec{v}_{01} = \vec{v}_1 - \vec{v}_0$ in $\Delta\vec{\theta}_{01}^{out} = \vec{\theta}_1^{out} - \vec{\theta}_0^{out}$.

Spremenljivko $\Delta\vec{\theta}_\mu$ določimo:

$$\Delta\vec{\theta}_\mu = \vec{\theta}_1^{out} - \vec{\theta}_\mu^{out} - \vec{A}_\mu(\vec{v}_0 - \vec{w}_\mu)$$

6.2 IZHODNE SPREMENLJIVKE $\vec{\theta}_n^{out}$

Transformacija $\vec{\Theta}(\vec{u}_{target})$ v enačbi (14) se nanaša na lokalno aproksimacijo glede na zmagovalni nevron. Za pozicioniranje prijemale manipulatorja ne uporabljamo le lokalno aproksimacijo za zmagovalni nevron, temveč za vse neurone, ki imajo težnostni vektor \vec{w}_μ dovolj blizu vektorju \vec{u}_{target} (sosednji neuroni). Zato je potrebno povprečiti vse sodelujoče neurone in jim določiti primerni težnostni faktor. Tisti neuroni, ki imajo zelo blizu težnosti vektor \vec{w}_μ glede na \vec{u}_{target} imajo velik težnostni faktor in obratno. Funkcija, ki popisuje porazdelitev težnostnih faktorjev je definirana v enačbi 19 in 20 ter jo imenujemo Gaussian-ova funkcija.

Izhodna spremenljivka $\vec{\theta}_0^{out}$ za grobi premik manipulatorja izračunava nevronska mreža po naslednji enačbi:

$$\vec{\theta}_0^{out} = s^{-1} \sum_k g^{mix}(k)(\vec{\theta}_{\mu k} + \vec{A}_{\mu k}(\vec{u}_{target} - \vec{w}_{\mu k})) \quad (21)$$

kjer je

$$s = \sum_k g^{mix}(k)$$

$$g^{mix}(k) = \exp\left(-\frac{\|y-s\|^2}{2\sigma}\right)$$

Podobno formulacijo lahko zapišemo za fini premik manipulatorja:

$$\vec{\theta}_n^{out} = \vec{\theta}_{n-1}^{out} + s^{-1} \sum_k g^{mix}(k) (\vec{\theta}_{\mu k} + \vec{A}_{\mu k} (\vec{u}_{target} - \vec{v}_{\mu k})) \quad (22)$$

kjer vrednost n določa število povratnih informacij iz kamere. V praksi se največkrat uporablja grobi premik $\vec{\theta}_0^{out}$ skupaj z enim finim premikom $\vec{\theta}_1^{out}$.

6.3 TRENIRANJE NEVRONSKE MREŽE

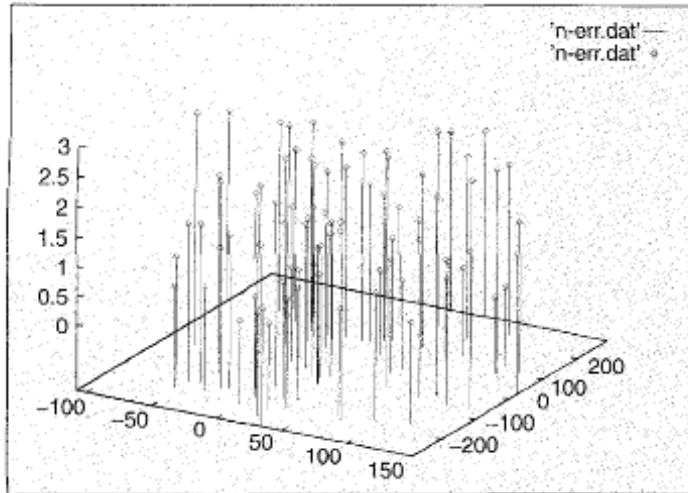
Med fazo učenja naključno izbiramo točke v delovnem prostoru manipulatorja. Za vsako izbrano točko je potrebno izvesti dve nalogi:

- **Določitev točke v prostoru:** Manipulator dobi informacijo iz dveh kamer o položaju točke v prostoru. Položaj točk je določen naključno z enakomerno porazdelitvijo spremenljivk sklepov manipulatorja. Prijemalo manipulatorja se premakne v okolico željene točke \vec{u}_{target} . Vrednosti sklepov manipulatorja so definirane v vektorju $\vec{\theta}_0^{out}$, kar smo definirali kot grobi premik.
- **Obnovitev pozicije iz vizualne informacije:** Po rotaciji sklepov manipulatorja za $\vec{\theta}_0^{out}$ s pomočjo vizualnih informacij določimo dejanski položaj prijemala \vec{v}_0 . Na podlagi dejanskega položaja prijemala določimo po enačbi (22) še fini premik manipulatorja za vrednosti sklepov $\vec{\theta}_1^{out}$. Za zelo natančni položaj manipulatorja glede na željen položaj je potrebno uporabiti večje število finih premikov $\vec{\theta}_2^{out}, \vec{\theta}_3^{out} \dots$

6.4 REZULTATI EKSPERIMENTA

Za treniranje Kohonen-ove mreže je bilo uporabljeno 12 x 7 x 4 nevronov. Željene točke v delovnem prostoru so bile izbrane naključno in na podlagi vsake naključne željene točke je nevronska mreža izračunala grobi premik $\vec{\theta}_0^{out}$ in fini premik $\vec{\theta}_1^{out}$ manipulatorja.

Na sliki 9 je prikazana napaka v položaju manipulatorja glede na izbrano željeno točko. Povprečna napaka v poziciji znaša 1.18 mm. Če primerjamo napako položaja iz slike 5, kjer smo uporabili Osnoven princip je razvidno, da je napaka z uporabo nevronskih mrež očitno zmanjšana. Prav tako je napaka na mejah delovnega prostora zmanjšana v primerjavi z uporabo predhodnega principa. Za treniranje nevronske mreže smo izbrali 2500 različnih željenih točk v prostoru, kar ustreza 2500 korakov treninga.



Slika 9: Napaka pozicije pri uporabi Prostega principa projeciran na XY ravnino

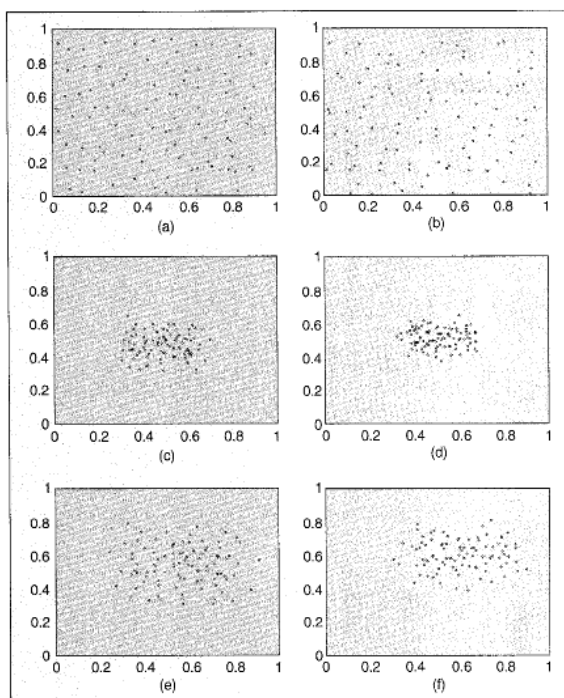
Algoritem z uporabo nevronske mreže omogoča adaptivno prilagajanje parametrov sistema tako, da bo napaka med željeno in dejansko točko v delovnem prostoru čim manjša, četudi pride do neželjenih sprememb v modelu kamere (drifting, osvetljevanje kamere, lega kamer...).

7 HIBRIDNI PRINCIP

Hibridni oziroma mešani princip je sestavljen iz Osnovnega in Prostega principa. Za treniranje uporablja Osnovni princip, ki je opisan v poglavju PROSTI PRINCIP. Treniranje se izvaja »off-line« (ni potrebno gibanje manipulatorja), kjer določimo le vektor \vec{u}_{target} (lokacijo željene točke). Na podlagi \vec{u}_{target} se s pomočjo inverzne kinematike izračuna vektor spremenljivk sklepov manipulatorja θ . Vrednosti za \vec{u}_{target} vzamemo iste, kot v prejšnjih dveh primerih (Prosti in Osnovni princip), saj lahko le tako sklepamo o prednostih in slabostih hibridnega principa. Pri prostem principu dobimo največje napake na robovih delovnega prostora. Zato za točke v območju robov delovnega prostora za trening še dodatno uporabimo Prosti princip, ki je izvršen »on-line« (potrebno gibanje manipulatorja).

7.1 REZULTATI EKSPERIMENTA

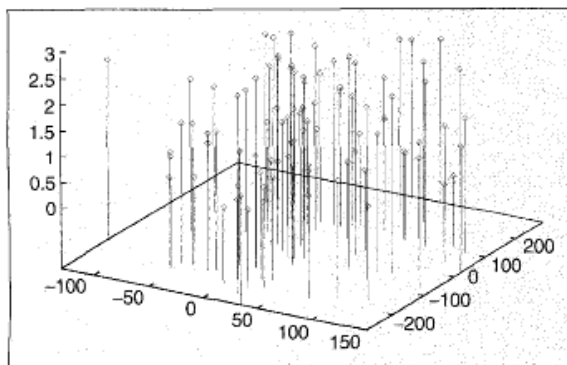
V prvem koraku naključno določimo vrednosti težnostnih vektorjev za vsak posamezni nevron, kar je prikazano na sliki 10a in 10b. Na sliki 10a je prikazana razporeditev težnostnih vektorjev za prvo kamero, na sliki 10b pa za drugo kamero. Obe kameri skupaj določata točke v delovnem prostoru manipulatorja. Med izvajanjem treninga se težnosti vektorji pomikajo proti željenim točkam v prostoru (\vec{u}_{target}). Faktor oziroma koeficient učenja ϵ določa hitrost konvergence težnostnih vektorjev proti ustreznim vektorjem \vec{u}_{target} . V začetni fazi učenja imamo velik faktor ϵ s čimer si zagotovimo dobro konvergenco težnostnih vektorjev. Proti koncu treninga pa faktor ϵ zmanjšujemo. Na sliki 10c in 10d so prikazani težnosti vektorji med treningom ter na sliki 10e in 10f težnosni vektorji po končanem »off-line« treningu.



Slika 10: Položaj težnostnih vektorjev v delovnem prostoru manipulatorja

Po končanem treningu dobimo natančnost pozicije prijemale manipulatorja kot pri Osnovnem principu. Da bi izboljšali natančnost dodatno treniramo »on-line« s Prostim principom. Željene točke za trening izberemo v območju robov delovnega prostora manipulatorja, kajti na tem območju ima Osnovni princip največje napake v poziciji. Izberemo okoli 50 željenih točk na tem območju in dodatno treniramo, s tem izboljšamo napako pozicioniranja na delih, kjer imamo veliko napako pozicije pri Osnovnem principu. Na sliki 11 je prikazana napaka pozicije pri uporabi hibridnega principa. Povprečna napaka pozicije znaša 1.67 mm, kar nam pove, da je napaka manjša od Osnovega principa in večja od Prostega principa. Napako na robovih delovnega prostora z uporabo hibridnega principa očitno izboljšamo v primerjavi z Osnovnim

principom. Največja prednost hibridnega principa se pokaže v času treninga nevronske mreže skupaj z dopustno napako pozicije manipulatorja.



Slika 11: Napaka pozicije pri uporabi hibridnega principa projicirana na XY ravnino

8 ZAKLJUČEK

V seminarju sem na kratko predstavil osnovne tri principe za krmiljenje manipulatorjev s tremi prostostnimi stopnjami. V Tabeli 1 so povzete vse pomembne karakteristike posameznih principov. Vsak princip je ocenjen glede na povprečno napako pozicije v delovnem prostoru («Average Error»), število korakov učenja oziroma treninga («Training Steps») in časa učenja algoritma («Time»).

Table 1. Performance of Different Algorithms			
<i>Scheme</i>	<i>Average Error</i>	<i>Training Steps</i>	<i>Time</i>
Model-based approach	1.67 mm	Not applicable	10 min
Model-free scheme	1.18 mm	750	5 hours
Hybrid approach	1.21 mm	2550	30 min

Tabela 1: Primerjava treh različnih principov za krmiljenje manipulatorja

Pri osnovnem principu je iz Tabele 1 razvidno, da imamo povprečno napako 1.67 mm, kar je največja povprečna napaka za vse tri principe. Na mejah delovnega prostora manipulatorja je bila napaka okoli 6 mm, kar je imelo največji vpliv na skupno povprečno napako. Kot je bilo omenjeno je povečana napaka pozicije ob mejah delovnega prostora nastala zaradi distorzije leče, ki jo povzročajo kamere. Ker ni potrebno izvajati nobenega učenja oziroma treninga je čas učenja le na račun izračunavanja modela kamere in manipulatorja, kar skupaj zneso slabih 10 minut. Čas izračunavanja modela kamere je odvisen od števila kalibracijskih točk.

Z uporabo Prostega principa, ki temelji na Kohonen's self-organizing shemi lahko napako pozicije v povprečju zmanjšamo na 1.18 mm, kar je tudi dopustna napaka manipulatorja. Pri Prostem principu ne smemo pozabiti na čas učenja, ki je odvisen od števila naključno izbranih točk v delovnem prostoru. Če vzamemo 750 naključno izbranih točk potrebujemo približno pet ur za učenje nevronske mreže. Napaka pozicije je približno enaka po celotnem delovnem prostoru manipulatorja.

Zadnji princip, katerega sem predstavil v seminarju je hibridni oziroma mešani princip. Trening hibridnega principa je sestavljen iz Osnovnega principa in Prostega principa. Prednost takšnega principa je v tem, da treniramo Osnovni princip »off-line« in Prosti princip »on-line«, kar nam zmanjša čas treninga na 30 minut. Napaka pozicije je manjša kot pri Osnovnem principu ter čas učenja krajši v primerjavi s časom učenja pri Prostem principu.

9 LITERATURA

[1] P.Liang, Y.L.Chang, and S. Hackwood, »Adaptive self-calibration of vision-based robot system« in IEEE Trans. Syst., Man, Cybern, vol. 19, no. 4, pp.811-824.

[2] M. Kuperstein, »Neural model of adaptive hand-eye coordination for single postures«, Science, vol.239, pp. 1308-1311.

[3] M. Kuperstein, »Adaptive visual-motor coordination in multijoint robots using parallel architecture«, Proc.IEEE Int. Automat. Robotics, Raleigh, NC, pp. 1595-1602.

[4] T.M.Martinetz, H.J.Ritter, and K.J. Schulten. Three-dimensional neural net for learning visual motor coordination of robot arm. IEEE Transactions on Neural Networks.

[5] J.A. Walter and K. J. Schulten. Implementation of self-organizing neural networks for visual-motor control of industrial robot. IEEE Transactions on Neural Networks.

[6] L.Behera and N. Kirubanandan. A hybrid neural control scheme for visual-motor coordination. IEEE Control System Magazine.