SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 444

INTERAKTIVNI INVERZNI KINEMATIČKI MODEL

Kristijan Šimunić

Zagreb, lipanj 2012.

Sadržaj

1. Uvod

Kinematika je dio grafičkog oblikovanja modela koji omogućuje pokretljivost samog modela. Osnova pokretljivosti je kostur koji je povezan s mrežom modela. Pomicanjem pojedine kosti pomičemo pridruženi dio mreže te na taj način upravljamo modelom i stvaramo željene pokrete, odnosno animaciju. Kinematiku možemo podijeliti na dvije grane: unaprijednu i inverznu kinematiku. Unaprijedna kinematika upravlja modelom definiranjem pozicije i orijentacije svake kosti kostura za svaki korak animacije. Za razliku od unaprijedne kinematike, inverzna kinematika upravlja modelom definiranjem položaja kosti koje su na krajevima kostura. Tako na primjeru čovjeka, definiranjem pozicije šake, stopala i glave upravljamo cijelim kosturom modela čovjeka. Problem inverzne kinematike je odrediti poziciju i orijentaciju svake kosti kostura na temelju pozicija kosti s kojima se upravlja modelom. U ovome radu razmatrat ćemo metode rješavanja problema inverzne kinematike i napraviti C++ aplikaciju koja koristi Ogre (eng. „Open-source 3D Graphic Rendering Engine“) gdje ćemo prikazati performanse pojedine metode na modelu robotske ruke odnosno modelu lava „Simbe“. Zatim ćemo međusobno usporediti metode te komentirati rezultate.

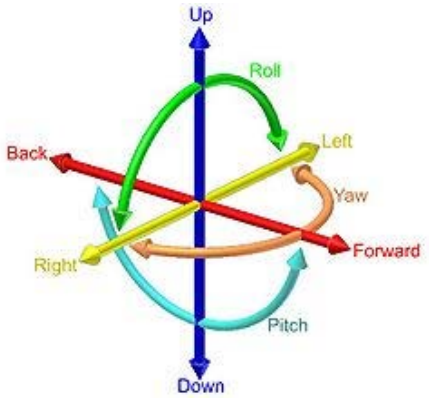
2. Kinematika

Pojam kinematike preuzeli smo iz terminologije korištene u robotici. U robotici njime označavamo granu mehanike koja se bavi proučavanjem gibanja tijela ne uzimajući u obzir sile pod čijim se djelovanjem to gibanje zbiva. Za to je zadužena dinamika koja zglobovima i vezama dodjeljuje masu i na njih djeluje silom, te analizira utjecaje sile na položaj sustava.

Dakle možemo reći da je kinematika grana grafičkog oblikovanja modela, gdje vodimo računa o pokretljivosti, njezinim ograničenjima i povezanosti različitih dijelova modela.

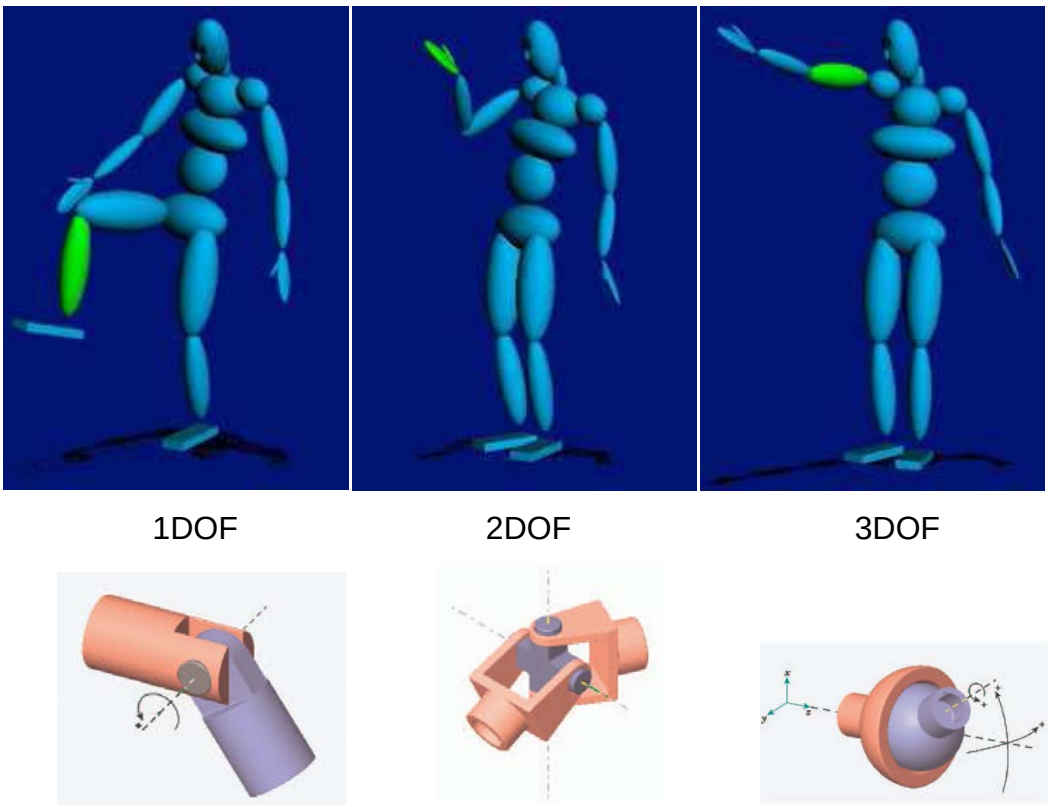
Model kinematičkih lanaca je jedan od jednostavnijih modela pristupa animaciji. Kinematički lanci se prirodno predstavljaju hijerarhijskim odnosom između članova, u našem slučaju kostiju, formiranih u stablastu strukturu. U modelu čovjeka, čvorovi stabla, segmenti su npr. potkoljenica, nadlaktica, ključna kost, itd. Segmenti su kruti elementi, dok grane imaju ulogu zglobova. Svaki čvor može imati više grana. U modelu čovjeka iz čvora trupa „izlaze" grane vrata, ramena, itd.

Za svaki zglob razlikujemo stupanj slobode (DOF – eng. „degrees of freedom“). Postoje rotacijski i translacijski stupnjevi slobode u skladu s mogućnostima kretanja. Po ograničenju stupnja slobode razlikujemo 3 stupnja slobode kod rotacije. To su rotacija oko X, Y i Z osi. Isto tako razlikujemo i 3 stupnja slobode kod translacije po X, Y i Z osi (Slika 2.1).



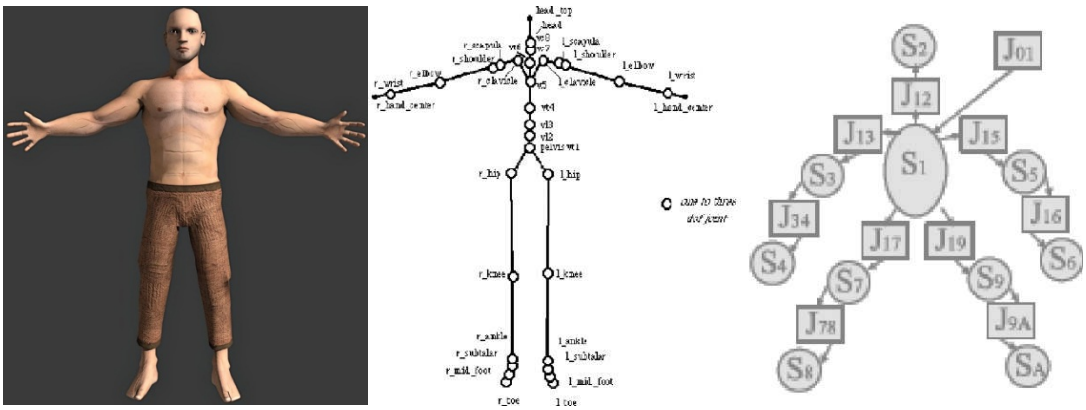
Slika 2.1 Rotacijski i translacijski stupnjevi slobode

Kod kostura čovjeka razlikujemo zglobove s maksimalno tri rotacijska stupnja slobode kao što je prikazano na slici ispod.



Slika 2.2 Rotacijski stupnjevi slobode kod čovjeka

Pri izgradnji hijerarhijskog modela, određeni objekt moramo prikazati pomoću grafa stablaste strukture. To ćemo ostvariti pomoću apstraktne reprezentacije našeg objekta. Na primjeru čovjeka Slika 2.3.



Slika 2.3 a) Model b) Apstraktna reprezentacija modela c) Graf stabla

Apstraktnu reprezentaciju modela radimo tako da označimo bitne točke modela, poput zglobova te ih povežemo nepokretnim dijelovima, čvorovima. Zglobu dodjeljujemo isti stupanj slobode koji ima i zglob u stvarnosti. Tako na primjer, koljeno kao zglob, modeliramo s granom stupnja slobode jedan (1 DOF). Vrat kao zglob, predstavljamo granom stupnja slobode tri (3DOF). Na slici 2.3b vidimo jedan primjer apstraktne reprezentacije modela. Sada radimo graf stabla prema izgledu te apstrakcije.

Izradu grafa provodimo pomoću transformacija iz izvornog čvora. Uzmimo, npr. da nam je segment trupa izvorni čvor. Na Slici 2.3b to je segment S1. Iz tog segmenta izvodimo, npr. segmente bedrene kosti, S7 i S9. Bitno je uočiti da segmenti izvedeni iz roditeljskog segmenta ovise o poziciji i obliku roditelja. Iz segmenta bedrene kosti S7 izveden je segment potkoljenica, S8. Pozicija ovog segmenta ovisit će o poziciji i obliku segmenta S7. Kako segment S7 ovisi o segmentu S1, možemo zaključiti da pozicija segmenta S8 ustvari ovisi i o segmentu S7 i o segmentu S1. Ova se pojava naziva *nizanje transformacija.*

Pojmom nizanja transformacija ostvarili smo pasivni oblik hijerarhijskog modela. U tom obliku možemo odrediti i koje se karakteristike nasljeđuju kroz čvorove: da li će segment-dijete imati iste teksture, boju, oblik, itd. kao segment roditelj. Međutim, postavlja se pitanje kako ćemo ostvariti gibanje određenih segmenata, te kakav će biti njihov utjecaj na segmente-djecu, odnosno segmente-roditelje. Tu dolazimo do razdiobe međusobnih utjecaja kinematičkih segmenata na dvije vrste: unaprijedna kinematika, te inverzna kinematika.

2.1. Kinematika kostura

Format matrice

Matrice 4x4 formata su jako korisne u računalnoj grafici zbog mogućnosti da s njima opišemo položaj i orijentaciju objekta u prostoru. Matrice mogu transformirati geometrijske podatke iz jednog prostora u drugi i zbog toga (između ostalog) se često koriste prilikom animacije nekog objekta. Matrica je uglavnom ovakvog izgleda:

$$M = \begin{bmatrix} a_x & a_y & a_z & 0 \\ b_x & b_y & b_z & 0 \\ c_x & c_y & c_z & 0 \\ d_x & d_y & d_z & 1 \end{bmatrix} \quad (2.1.1)$$

gdje su **a**, **b** i **c** tri bazna vektora koja definiraju orijentaciju objekta (podprostora), a **d** je pozicija. Ovo vrijedi samo ako su vektori **a**, **b** i **c** međusobno okomiti i normirani (duljina im je jednaka 1), drugim riječima ako je matrica 3x3 sastavljena od vektora **a**, **b** i **c** *ortonormalna*.

Transformacije

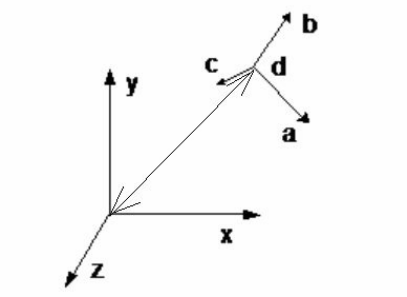
Vektor je transformiran matricom na slijedeći način:

$$v' = v \cdot M \quad (2.1.2)$$

Ako je **v** točka zapisana u lokalnom prostoru sustava i **M** je matrica koja postavlja objekt u globalni prostor sustava, onda je **v'** pozicija točke **v** u globalnom prostoru sustava. Inverzna transformacija, dakle transformacija iz globalnog u lokalni prostor izgleda ovako:

$$v = v' \cdot M^{-1} \quad (2.1.3)$$

gdje je $\mathbf{M^{-1}}$ je *inverz matrice* **M.**



Slika 2.1.1: Globalni i lokalni prostor

Koordinatni sustavi

U 3D grafici i animaciji definiramo jedan fiksni koordinatni sustav kojeg zovemo globalni koordinatni sustav (*world space, global space)* u kojem svi objekti, likovi, efekti, kamere, svijetla i drugi entiteti koegzistiraju.

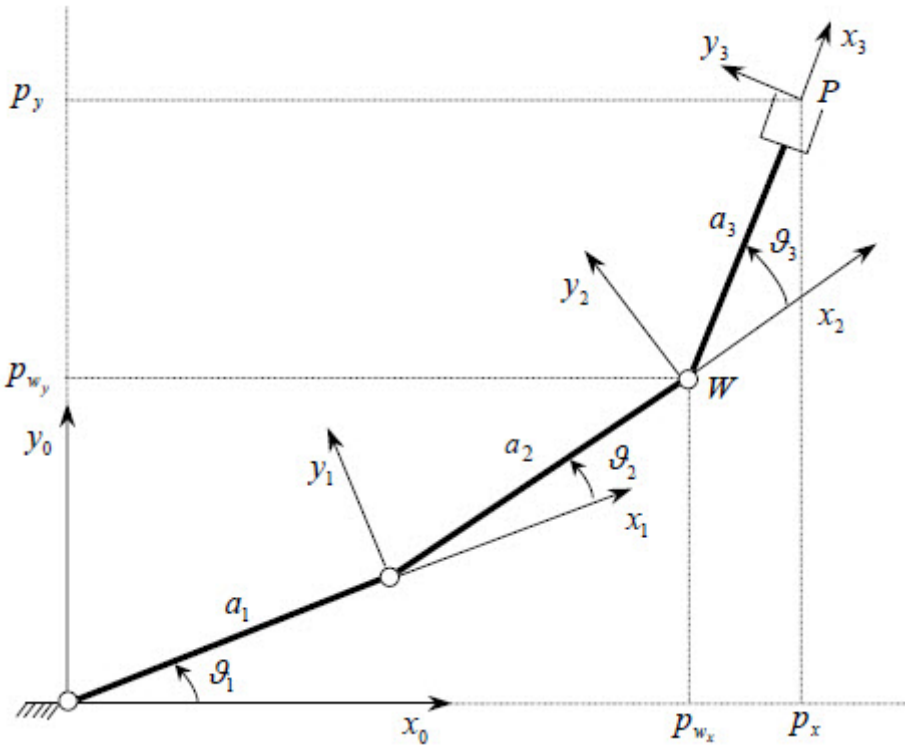
Individualni objekti tipično su definirani u njihovom lokalnom koordinatnom sustavu (*local space)*, te koriste matrice 4x4 kako bi se transformirale u globalni sustav.

U tipičnim interaktivnim grafičkim aplikacijama mnogi različiti objekti trebaju koegzistirati u globalnom koordinatnom sustavu. Neki od tih objekata su jednostavni objekti (stolice). Kako bi manipulirali položajem i orijentacijom stolice, koristimo matrice da bi transformirali točke stolice iz lokalnog sustava stolice u globalni sustav okoline.

Složeniji objekti, kao što su ljudi i životinje, će imati više pokretnih dijelova, pa će stoga trebati više matrica. Kako bi prikazivali takve objekte u globalnom prostoru i na njima izvodili operacije, kao što je detekcija kolizije, potrebne su matrice koje transformiraju iz lokalnog u globalni sustav (*world matrix*) za sve definirane zglobove unutar recimo ljudskog kostura. Pošto je direktno određivanje matrica zglobova (*world matrice zglobova)* komplicirano i neintuitivno, *kinematički lanac (kostur)* je sagrađen hijerarhijski, gdje je svaki zglob definiran relativno u odnosu na nadređeni zglob. Matrice zglobova su definirane u lokalnom sustavu objekta i pretvaraju se u *world matrice* procesom **unaprijedne kinematike**.

2.2. Unaprijedna kinematika

Unaprijedna kinematika za manipulator (manipulator je prikladan termin preuzet iz robotike, a odnosi se na promatrani model koji ima jedan nepomični zglob – bazu, i skup povezanih pomičnih zglobova) zadaje položaje i kutove svih zglobova, te na osnovu njih izračunava položaj krajnje točke manipulatora. Ovo je trivijalan zadatak sa samo jednim rješenjem.



Slika 2.2.1 Unaprijedna kinematika

Kompletno određena poza 3D grafičkog modela specificirana je poznim vektorom (eng. „pose vector“). Pozni vektor čuva pozicije svih zglobova u modelu.

Unaprijedna kinematika, kao princip određivanja položaja jedinica strukture modela, postavlja se kao jednostavno rješenje, ali ne i efikasno. Za ostvarenje pokreta potrebno je mnogo ulaznih podataka, a kao rezultat dobivamo pokrete koji nisu prirodni. Zbog lakšeg modeliranja, te zbog činjenice da ukoliko želimo doći do željene točke s nekim segmentom Sn, moramo postepeno pomicati svaki segment koji je roditelj segmentu Sn, ova je metoda istisnuta od strane inverzne kinematike.

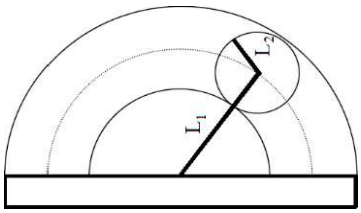
2.3. Inverzna kinematika

Inverzna kinematika radi postupak obrnut unaprijednoj kinematici – za zadanu krajnju točku manipulatora ona pronalazi položaje i kutove svih zglobova takve da manipulator bude u traženoj krajnjoj točki, ako je ista dohvatljiva, ili što je moguće bliže ako nije dohvatljiva. Ovo je mnogo složeniji problem, redovito s beskonačno mnogo rješenja.

Inverzna kinematika je, očito, puno upotrebljivija i zanimljivija od unaprijedne kinematike – za bilo kakvu animacijsku tehniku, puno nam je interesantnije da možemo samo zadati krajnju točku na kojoj mora biti ruka (npr., dohvatiti šalicu) nego da moramo točno navoditi sve položaje zglobova. Isto tako, snimanjem stvarnih pokreta čovjeka 3D senzorima za potrebe analize, dovoljno je snimati samo položaj krajnje točke, a ne položaje svih zglobova.

Za shvaćanje principa rada inverzne kinematike prvo ćemo objasniti pojam dohvatljivog radnog prostora (eng. „reachable workspace”*).* Na slici 2.3.1 vidimo jednostavan primjer, s označenim radnim prostorom. Tanke pune linije označavaju doseg segmenta S2, dok isprekidana predstavlja doseg segmenta S1. Matematički bi radni prostor mogli opisati sljedećom formulom:

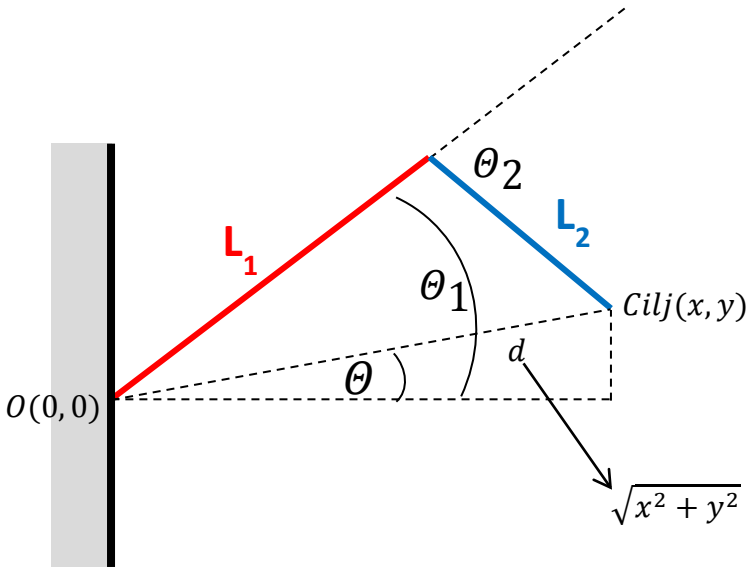
$$L_1 - L_2 \leq radni\ prostor \leq L_1 + L_2 \qquad (2.3.1)$$



Slika 2.3.1 Dohvatljivi radni prostor

Važno je napomenuti da neki zglobovi imaju ograničenja u pokretima. Na Slici 2.3.1 čvor L_1 može se pomicati samo 180 stupnjeva. Ovo je vrlo česta pojava u oblikovanju 3D modela.

Za prikaz principa rada inverzne kinematike koristit ćemo najjednostavniji primjer: dva segmenta povezana međusobno i zajedno na čvrstu točku:



Slika 2.3.2 Princip inverzne kinematike

Segmenti su kruti elementi modela, te imaju fiksnu duljinu. Uz poznate duljine i zadan položaj vrha manipulatora, odredimo nepoznate kutove Θ_1 i Θ_2.

Kut Θ računamo preko x i y koordinata točke $Cilj$:

$$cos\Theta = \frac{x}{\sqrt{x^2+y^2}} \tag{2.3.2}$$

Kosinusovim poučkom

$$c^2 = a^2 + b^2 - 2abcos\alpha \tag{2.3.3}$$

i (2.3.2) računamo kut $\Theta_1 - \Theta$:

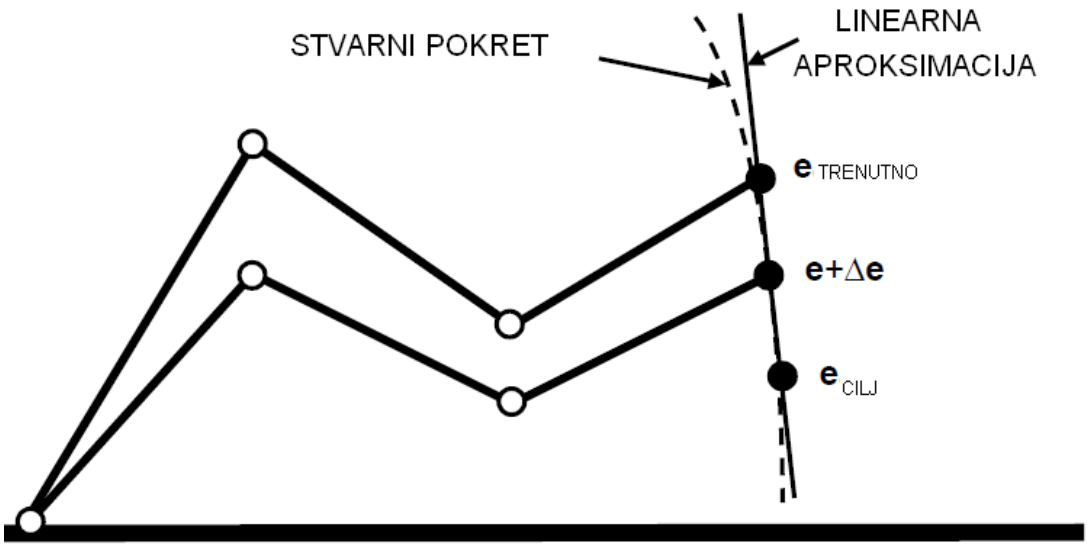
$$\cos(\Theta_1 - \Theta) = \frac{L_1^2 + x^2 + y^2 - L_2^2}{2L_1\sqrt{x^2+y^2}} \tag{2.3.4}$$

Zatim kut $180° - \Theta_2$:

$$\cos(180° - \Theta_2) = \frac{L_1^2 + L_2^2 - (x^2+y^2)}{2L_1L_2} \tag{2.3.5}$$

Ovim postupkom dobivamo samo dva rješenja. Ona su simetrična u odnosu na liniju koja spaja ishodište lanca i točku cilja. U gore spomenutim primjerima, govorimo o isprekidanoj liniji d. Bitno je napomenuti da se slučaj s dva rješenja pojavljuje samo kada upravljamo modelom s dva čvora. U složenijim primjerima nerijetko postoji beskonačno mnogo rješenja.

Glavna problematika inverzne kinematike, zbog nelinearnosti funkcija sinus i kosinus je teška aproksimacija linearnih pokreta. Linearni pokreti su, ipak, česta pojava u stvarnom svijetu.



Slika 2.3.3 Odnos linearne aproksimacije i stvarnog pokreta

U nastavku ćemo se pozabaviti nekim češće korištenim rješenjima ovog problema.

3. Metode rješavanja problema inverzne kinematike

Postoji nekoliko metoda za rješavanje problema inverzne kinematike. Izvorno dolaze iz aplikacija vezanih za robotiku. To su metode CCD (eng. „Cyclic Coordinate Descent“), pseudo-inverzni Jakobijan (eng. „Pseudo-inverse Jacobian“), transponirani Jakobijan (eng. „Jacobian transpose“), DLS metoda (eng. „Damped Least Squares“), quasi-Newton, konjugirani gradijent, neuronske mreže i umjetna inteligencije. Ovaj rad se fokusira na rješavanje problema inverzne kinematike u stvarnom vremenu pomoću prve četiri navedene metode.

U nastavku rada koristit ćemo pojam krajnjeg djelovatelja (eng. „end effector“) koji predstavlja zadnji (najniži u hijerarhiji) segment odnosno kost hijerarhijskog kinemtičkog lanca. Točnije predstavlja zglob koji bi postojao kada bi postojao još jedan segment-dijete.

Koristiti ćemo i pojam konfiguracije koji predstavlja trenutni položaj i orijentaciju svih segmenata kinematičkog lanca.

Mijenjanjem položaja zadnjeg djelovatelja mijenjamo konfiguraciju cijelog kinematičkog lanca.

Želimo koristiti metode koje su robusne i dobro se ponašaju u gotovo svim situacijama. Kao dio robusnosti, želimo da kad krajnji djelovatelj prati ciljnu poziciju, da metode obavljaju dobar posao čak i kada su ciljne pozicije nedohvatljive. Koristeći prve četiri navedene metode razvit ćemo aplikaciju u kojoj će model mijenjati konfiguraciju, pozu u stvarnom vremenu upravljajući modelom pomoću krajnjih djelovatelja.

Mogli biste se zapitati zašto je važno omogućiti nedohvatljivost ciljnih pozicija. Postoji nekoliko razloga: Prvo, može biti teško u potpunosti

eliminirati mogućnost nedohvatljivih pozicija, a još uvijek se može dobiti željeni pokret. Drugo, ako su ciljne pozicije gotovo dohvatljive i mogu se doseći samo s potpuno ispruženim zglobovima, onda je situacija vrlo slična s nedohvatljivim ciljevima. Nažalost, situacije u kojima su ciljne pozicije nedohvatljive teško je robusno riješiti. Mnoge metode, kao što su pseudo-inverzni Jakobijan ili transponirani Jakobijan će jako oscilirati u tim situacijama, međutim DLS metoda će se dobro ponašati u tim situacijama.

Model je modeliran s nizom segmenata povezanih zglobovima. Postoji niz različitih mogućih tipova zglobova. Najčešći tip zgloba je rotacijski zglob opisan samo s jednim kutom. Ostale vrste zglobova su prizmatični (tj. translacijski ili klizni), zglobovi u obliku vijka, itd. Radi jednostavnosti, razmatrat će se samo rotacijski zglobovi, ali algoritmi i teorija svega odnosi se na proizvoljne zglobove. Konfiguracija zgloba je kontinuirana funkcija jednog ili više skalara. Za rotacijske zglobove skalar je kut zgloba. Kompletna konfiguracija modela određena je skalarima $\theta_1, \ldots, \theta_n$ koji opisuju konfiguraciju zglobova. Pretpostavimo da postoji n zglobova i svaka vrijednost θ_j predstavlja kut zgloba (ali, kao što je navedeno, može općenito predstavljati bilo koju vrijednost koja nije kut). Određeni segmenti su identificirani kao krajnji djelovatelji. Ako postoji k krajnjih djelovatelja, njihove pozicije u prostoru su obilježene s $s_1, s_2, \ldots, s_k$. Svaka pozicija krajnjeg djelovatelja je funkcija kutova zglobova.

Pišemo $\vec{s}$ kao vektor stupac $(s_1, s_2, \ldots, s_k)^T$ koji možemo gledati kao vektor stupac sa $m = 3k$ skalara ili k elemenata iz $\mathbb{R}^3$. Modelom se upravlja definiranjem ciljnih pozicija za krajnje djelovatelje. Ciljne pozicije definirane su vektorom $\vec{t} = (t_1, t_2, \ldots, t_k)^T$ gdje je t_i ciljna pozicija za i-ti krajnji djelovatelj. Neka je $e_i = t_i - s_i$ željena promjena pozicija i-tog krajnjeg djelovatelja. Neka je $\vec{e} = \vec{t} - \vec{s}$.

Kutovi zglobova zapisani su kao vektor stupac $\theta = (\theta_1, \theta_2, \ldots, \theta_n)^T$. Pozicije krajnjih djelovatelja su funkcije kutova zglobova što se može izraziti kao

$\vec{s} = \vec{s}(\theta)$ odnosno za svaki $i = 1, \dots, k, s_i = s_i(\theta)$. Problem inverzne kinematike je naći vrijednosti θ_j za koje je

$$t_i = s_i(\theta), \text{ za sve } i. \tag{3.1}$$

Nažalost, ne postoji uvijek rješenje za (3.1), kao što ne mora postojati jedinstveno (najbolje) rješenje. Čak i u situacijama koje se dobro ponašaju, jednadžba rješenja ne mora biti u zatvorenom obliku. Možemo, međutim, koristiti iterativne metode za približno dobro rješenje. Za to, funkcije s_i su linearne aproksimacije koje koriste Jakobijevu matricu. Jakobijeva matrica J je funkcija od θ i definirana je kao

$$J(\theta) = \left(\frac{\partial s_i}{\partial \theta_j}\right)_{i,j}. \tag{3.2}$$

Jakobijan se može gledati kao $k \times n$ matrica čiji elementi su vektori iz $\mathbb{R}^3$ ili kao $m \times n$ matrica čiji elementi su skalari gdje je $m = 3k$.

Osnovna jednadžba unaprijedne kinematike koja opisuje brzinu krajnjeg djelovatelja može biti zapisana kao (s točkom kao oznakom prve derivacije):

$$\dot{\vec{s}} = J(\theta)\dot{\theta} \tag{3.3}$$

Jakobijan vodi do iterativne metode rješavanja jednadžbe (3.1). Pretpostavimo da imamo trenutne vrijednosti θ, $\vec{s}$ i $\vec{t}$. Pomoću njih izračunamo Jakobijan $J = J(\theta)$. Tražimo promjenu $\Delta\theta$ kako bismo promijenili kutove zglobova θ za $\Delta\theta$:

$$\theta := \theta + \Delta\theta \tag{3.4}$$

Preko (3.3) promjena u poziciji krajnjeg djelovatelja uzrokovana promjenom kutova zglobova može se otprilike odrediti kao

$$\Delta\vec{s} \approx J\Delta\theta. \tag{3.5}$$

Ideja je da se vrijednosti $\Delta\theta$ biraju tako da je $\Delta\vec{s}$ otprilike jednako $\vec{e}$, iako je također čest slučaj izabrati vrijednost $\Delta\theta$ tako da pomak $\Delta\vec{s}$ krajnjih djelovatelja djelomično prati brzine ciljnih pozicija. Promjena kutova zglobova

može se koristiti na dva načina: (i) Svaki simulacijski korak obavlja jednu promjenu vrijednosti kuta zgloba koristeći jednadžbu (3.4), tako da krajnji djelovatelji prate ciljne pozicije. (ii) Kutovi zglobova se ažuriraju iterativno sve dok vrijednost $\vec{s}$ nije dovoljno blizu rješenja. Također je moguće koristiti hibrid (i) i (ii), malim brojem ponavljanja ažurirati kutove koristeći (3.4) kako bi se što bolje pratile pozicije krajnjeg djelovatelja.

Ostatak ovog rada diskutira strategije biranja $\Delta\theta$ za ažuriranje kutova zglobova. Na temelju (3.5) jedan pristup je riješiti jednadžbu (3.6) za $\Delta\theta$.

$$\vec{e} = J\Delta\theta \tag{3.6}$$

U većini slučajeva ne postoji jedinstveno rješenje jednadžbe. Jakobijan J ne mora biti kvadratna matrica ili invertibilna, a ako je i invertibilna određivanje $\Delta\theta = J^{-1}\vec{e}$ može davati loše rezultate kada je J blizu singulariteta.

Elemente Jakobijeve matrice obično je lagano izračunati. Ako je j-ti zglob rotacijski zglob s jednim stupnjem slobode tada je kut zgloba skalar θ_j. Neka je $\boldsymbol{p_j}$ pozicija zgloba i neka je $\boldsymbol{v_j}$ jedinični vektor sa smjerom trenutne osi rotacije zgloba. U tom slučaju ako su kutovi mjereni u radijanima sa smjerom rotacije dobivene pravilom desne ruke i ako na i-ti krajnji djelovatelj utječe j-ti zglob tada je odgovarajući element u Jakobijanu

$$\frac{\partial s_i}{\partial \theta_j} = v_j \times (s_i - p_j). \tag{3.7}$$

Ako j-ti zglob ne utječe na i-ti krajnji djelovatelj tada je $\partial s_i / \partial \theta_j = 0$.

Ako je j-ti zglob translacijski, elemente Jakobijeve matrice je još jednostavnije izračunati. Pretpostavimo da j-ti zglob translatira u smjeru vektora v_j tako da „kut“ zgloba mjeri udaljenost u smjeru vektora v_j. Tada, ako j-ti zglob utječe na i-ti krajnji djelovatelj, vrijedi

$$\frac{\partial s_i}{\partial \theta_j} = v_j \tag{3.8}$$

Alternativna metoda za definiranje Jakobijeve matrice je

$$J(\theta) = \left(\frac{\partial t_i}{\partial \theta_j}\right)_{i,j}, \tag{3.9}$$

gdje se parcijalna derivacija računa koristeći formulu ($\partial s_i / \partial \theta_j$) s t_i umjesto s_i. Značenje $\partial t_i / \partial \theta_j$ je da se ciljna pozicija smatra kao da je prikačena na isti segment kao i i-ti krajnji djelovatelj. S ovom formulacijom Jakobijana pokušavamo pomaknuti ciljnu poziciju bliže krajnjem djelovatelj-u umjesto da krajnji djelovatelj pokušavamo približiti ciljnoj poziciji.

Alternativni Jakobijan se može koristiti u bilo kojem opisanom algoritmu. U praksi alternativni Jakobijan reducira oscilacije kada je ciljna pozicija predaleko da bi se mogla doseći s krajnjim djelovateljom. Međutim, negativna strana korištenja alternativnog Jakobijana je što u nekim konfiguracijama dolazi do „čudnog“ ponašanja. To je uglavnom slučaj za rotacijske zglobove kada su segmenti skupljeni te kada svi pokušavaju dosegnuti obližnju ciljnu poziciju.

Približavanje ciljnih pozicija.

Kad su ciljne pozicije previše udaljene svi zglobovi kinematičkog lanca su ispruženi. Jednom kada je kinematički lanac tako ispružen blizu je singulariteta (Jakobijan je jako osjetljiv na male promjene kutova) i model će često trzati neuspješno pokušavajući doseći udaljenu ciljnu poziciju. Ovaj efekt se može reducirati koristeći DLS metodu, ali problem ga je u potpunosti ukloniti.

Jedna metoda kojom bi se smanjio ovaj problem je pomaknuti ciljne pozicije bliže poziciji krajnjeg djelovatelja. Zbog toga mijenjamo definiciju vektora $\vec{e}$. Umjesto $\vec{e} = \vec{t} - \vec{s}$, svaka komponenta e_i vektora $\vec{e}$ ima duljinu smanjenu na određenu maksimalnu vrijednost.

Definiramo

$$e_i = SmanjiVeličinu(t_i - s_i\ , D_{maks}\ )$$

gdje je

$$SmanjiVeličinu(w, d) = \begin{cases} w & ako\ je\ \|w\| \leq d \\ d\dfrac{w}{\|w\|} & inače \end{cases}$$

Ovdje $\|w\|$ predstavlja Euklidsku udaljenost od w. Vrijednost D_{maks} je gornja granica koliko želimo pomaknuti krajnji djelovatelj u pojedinom koraku.

Kod DLS metode, smanjivanje veličine $\overrightarrow{e_i}$ na ovaj način smanjuje oscilacije kada su ciljne pozicije izvan dohvata. Ovo ima prednost dopuštanja manjih vrijednosti konstanti za odbacivanje. Manje konstante za odbacivanje dopuštaju znatno bržu konvergenciju do ciljnih pozicija. Kada krajnji djelovatelj kontinuirano prati ciljnu poziciju u pokretu, udaljenost D_{maks} bi trebala biti nekoliko puta veća nego što se krajnji djelovatelj pomiče u jednom koraku. Postavljanje D_{maks} na približno pola duljine uobičajenog segmenta ima prilično dobar učinak.

3.1. Transponirani Jakobijan

Osnovna ideja je vrlo jednostavna: Koristiti transponiranu matricu J umjesto inverzne matrice J. Definiramo $\Delta\theta$ kao

$$\Delta\theta = \alpha J^T \vec{e}, \tag{3.1.1}$$

za neki odgovarajući skalar α. Transponirana Jakobijeva matrica nije naravno isto što i inverzna matrica, ali je moguće prilagoditi korištenje transponirane matrice u smislu virtualnih snaga. Bez gubitka općenitosti može se pokazati da vrijedi sljedeći teorem [7, 8].

Teorem 1 $Za\ sve\ J\ i\ \vec{e},\ \langle JJ^T\vec{e},\ \vec{e}\rangle \geq 0.$

Dokaz Dokaz je jednostavan: $\langle JJ^T\vec{e},\ \vec{e}\rangle = \langle J^T\vec{e},\ J^T\vec{e}\ \rangle = \|J^T\vec{e}\|^2 \geq 0.$

Aproksimacija (3.5) implicira da za svaki dovoljno mali $\alpha > 0$, ažuriranje kutova jednadžbom (3.4) koristeći $\Delta\theta = \alpha J^T \vec{e}$ promijenit će pozicije krajnjeg djelovatelja za približno $\alpha J J^T \vec{e}$. Prema Teoremu 1, ovo ima efekt smanjivanja veličine vektora pogreške $\vec{e}$ ako je vrijednost α dovoljno mala. Ostaje odlučiti kako odabrati vrijednost α. Jedna mogućnost je pokušati minimizirati vrijednost vektora pogreške $\vec{e}$ nakon ažuriranja. Pretpostavimo da bi promjena pozicije krajnjeg djelovatelja bila točno $\alpha J J^T \vec{e}$, i odaberimo α tako da bi ta vrijednost bila što bliže vrijednosti $\vec{e}$.

Time dobivamo

$$\alpha = \frac{\langle \vec{e}, JJ^T \vec{e} \rangle}{\langle JJ^T \vec{e}, JJ^T \vec{e} \rangle}. \tag{3.1.2}$$

3.2. Pseudo-inverzni Jakobijan

Pseudo-inverzna metoda postavlja vrijednost $\Delta\theta$ jednaku

$$\Delta\theta = J^{\dagger} \vec{e}. \tag{3.2.1}$$

Gdje je $J^{\dagger}$ pseudo-inverz matrice J dimenzija $n \times m$ poznat kao Moore-Penrose [5, 6] inverz matrice J. Definiran je za sve matrice J, čak i za one koje nisu kvadratne ili punog ranga. Pseudoinverz daje najbolja moguća rješenja jednadžbe $J\Delta\theta = \vec{e}$ u smislu najmanjih kvadrata. Konkretno, pseudo-inverz ima sljedeća korisna svojstva. Neka je $\Delta\theta$ definiran jednadžbom (3.1). U prvom slučaju pretpostavimo da je $\vec{e}$ istog ranga kao i J. U tom slučaju, $J\Delta\theta = \vec{e}$. Nadalje $\Delta\theta$ je jedinstveni vektor najmanjih kvadrata koji zadovoljava jednadžbu $J\Delta\theta = \vec{e}$. U drugom slučaju pretpostavimo da $\vec{e}$ nije istog ranga kao i J. U tom slučaju nije moguće $J\Delta\theta = \vec{e}$. Međutim, $\Delta\theta$ ima svojstvo da minimizira razliku vrijednosti $J\Delta\theta - \vec{e}$. Nadalje, $\Delta\theta$ je jedinstveni vektor najmanjih kvadrata koji minimizira $\|J\Delta\theta - \vec{e}\|$ ili ekvivalentno tome minimizira $\|J\Delta\theta - \vec{e}\|^2$.

Pseudo-inverz ima problema u stabilnosti u blizini singulariteta. Kod singulariteta Jakobijeva matrica nema rang punog reda, što odgovara činjenici da postoji promjena pozicije krajnjeg djelovatelja koja nije dohvatljiva. Ako je konfiguracija točno kod singulariteta tada pseudo-inverzna metoda neće pokušati pomaknuti u nemogućem smjeru i pseudo-inverz će se dobro ponašati. Međutim, ako je konfiguracija blizu singulariteta tada će pseudo-inverz voditi do jako velikih promjena u kutovima čak i za male promjene ciljne pozicije. U praksi pogreška kod zaokruživanja znači da će se pravi singulariteti rijetko pojavljivati i umjesto singulariteta potrebno je detektirati vrijednosti oko nule.

Pseudo-inverz također ima svojstvo da -matrica $\left(I - J^{\dagger}J\right)$ vrši projekciju na nulprostor matrice J. Prema tome za sve vektore φ, $J\left(I - J^{\dagger}J\right)\varphi = 0$. Što znači da možemo postaviti $\Delta\theta$ kao

$$\Delta\theta = J^{\dagger}\vec{e} + \left(I - J^{\dagger}J\right)\varphi \tag{3.2.2}$$

za bilo koji vektor φ i opet dobiti vrijednost za $\Delta\theta$ koja minimizira vrijednost $J\Delta\theta - \vec{e}$. Odabirom odgovarajućeg φ moguće je postići dodatne ciljeve uz to što krajnji djelovatelj prati ciljnu poziciju. Na primjer, φ može biti odabran tako da pokušava vratiti zglobove u početni položaj što može pomoći u izbjegavanju singularnih konfiguracija.

Algoritam za pseudo-inverznu metodu može biti izveden na sljedeći način:

Iz jednadžbe (3.6) dobijemo *normalnu jednadžbu*

$$J^{T}J\Delta\theta = J^{T}\vec{e}. \tag{3.2.3}$$

Neka je $\vec{z} = J^{T}\vec{e}$. Riješimo jednadžbu

$$(J^{T}J)\Delta\theta = \vec{z}. \tag{3.2.4}$$

Može se pokazati da je $\vec{z}$ uvijek u rangu od $J^{T}J$. Iz tog razloga jednadžba (3.2.4) uvijek ima rješenje. U principu operacijama nad redcima moguće je

naći rješenje s najmanjom veličinom, međutim u blizini singulariteta algoritam je inherentno numerički nestabilan.

Kada J ima puni rang tada je JJ^T sigurno inverzna. U tom slučaju rješenje s najmanjom veličinom $\Delta\theta$ za jednadžbu (3.2.4) može se izraziti kao

$$\Delta\theta = J^T(JJ^T)^{-1}\vec{e}. \tag{3.2.5}$$

Da bi dokazali ovo primijetimo da ako $\Delta\theta$ zadovoljava (3.2.5) tada je $\Delta\theta$ istog reda kao vektor retka matrice J te vrijedi $J\Delta\theta = \vec{e}$. Jednadžba (3.2.5) se ne može koristiti ako J nije punog ranga.

Pseudo-inverzna metoda je često spominjana u literaturi međutim ima slabe performanse zbog nestabilnosti kod singulariteta. DLS metoda ima puno bolje performanse.

3.3. DLS

DLS metoda (eng. „Damped Least Squares“) također poznata kao Levenberg-Marquardt-ova [5, 6] metoda izbjegava mnoge probleme pseudo-inverzne metode sa singularitetima i može dati numerički stabilnu metodu odabira $\Delta\theta$.

DLS se može u teoriji objasniti na sljedeći način. Umjesto traženja minimum vektora $\Delta\theta$ koji daje najbolje rješenje jednadžbi (3.6) tražimo vrijednost od $\Delta\theta$ koja minimizira vrijednost

$$\|J\Delta\theta - \vec{e}\|^2 + \lambda^2\|\Delta\theta\|^2, \tag{3.3.1}$$

gdje je $\lambda \in \mathbb{R}$ nenegativna konstanta za odbacivanje. To je ekvivalentno minimiziranju vrijednosti

$$\left\| \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta - \begin{pmatrix} \vec{e} \\ 0 \end{pmatrix} \right\|. \tag{3.3.2}$$

Odgovarajuća normalna jednadžba je

$$\begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta = \begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} \vec{e} \\ 0 \end{pmatrix}. \tag{3.3.3}$$

Što se može raspisati kao

$$(J^T J + \lambda^2 I)\Delta\theta = J^T \vec{e}. \tag{3.3.4}$$

Može se pokazati (metodama SVD) da $J^T J + \lambda^2 I$ nije singularno. Radi toga rješenje DLS metode je jednako

$$\Delta\theta = (J^T J + \lambda^2 I)^{-1} J^T \vec{e}. \tag{3.3.5}$$

$J^T J$ je $n \times n$ matrica, gdje je n broj stupnjeva slobode. Lako je pokazati da je $(J^T J + \lambda^2 I)^{-1} J^T = J^T (JJ^T + \lambda^2 I)^{-1}$. Zato je

$$\Delta\theta = J^T (JJ^T + \lambda^2 I)^{-1} \vec{e}. \tag{3.3.6}$$

Prednost jednadžbe (3.3.6) nad (3.3.5) jest da je matrica nad kojom se računa inverz dimenzija $m \times m$, gdje je $m = 3k$ dimenzija prostora ciljne pozicije i m je obično puno manji nego n.

Dodatno, (3.3.6) može biti izračunata bez računanja inverza. Operacijama nad recima može se naći $\vec{f}$ takav da je $(JJ^T + \lambda^2 I)\vec{f} = \vec{e}$. Tada je rješenje $J^T \vec{f}$.

Vrijednost konstante za odbacivanje ovisi o detaljima modela i ciljnoj poziciji. Mora se pažljivo odabrati kako bi jednadžba (3.3.6) bila numerički stabilna. Vrijednost konstante za odbacivanje bi trebala biti dovoljno velika da se rješenja za $\Delta\theta$ dobro ponašaju blizu singulariteta, ali ako je prevelika tada je konvergencija vrlo spora. Postoje mnoge metode za dinamičko određivanje vrijednosti konstante za odbacivanje međutim to prelazi okvire ovoga rada.

3.4. Dekompozicija singularne vrijednosti

Dekompozicija singularne vrijednosti (Eng. „Singular value decomposition“), skraćeno SVD, omogućuje moćnu metodu za analiziranje pseudo-inverzne metode i DLS metode. Neka je J Jakobijeva matrica. Dekompozicija singularne vrijednosti matrice J sastoji se od izražavanja matrice J u obliku

$$J = UDV^T, \tag{3.4.1}$$

gdje su U i V ortogonalne matrice, a D dijagonalna matrica. Ako je J matrica dimenzija $m \times n$ tada U ima dimenzije $m \times m$, D $m \times n$ i V $n \times m$. Jedine vrijednosti elementa matrice D koji nisu nule su $\sigma_i = d_{i,i}$ duž dijagonale.

Pretpostavimo da je $m \leq n$. Bez gubitka općenitosti, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$. Vrijednosti σ_i mogu biti nule. Zapravo, rang matrice J je jednak najvećoj vrijednosti r takvoj da je $\sigma_r \neq 0$. Za svaki $i > t, \sigma_i = 0$. Koristimo $\boldsymbol{u}_i$ i $\boldsymbol{v}_i$ kako bi zapisali i-te stupce matrica U i V. Ortogonalnost matrica U i V implicira da su stupci matrice U (odn., matrice V) ortonormalne osnovice za $\mathbb{R}^m$ (odn., za $\mathbb{R}^n$). Vektori $v_{r+1}, \ldots, v_n$ su ortonormalne osnovice za nulprostor matrice J. Dekompozicija singularne vrijednosti matrice J uvijek postoji i podrazumijeva se da J može biti zapisana u obliku

$$J = \sum_{i=1}^{m} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i{}^T = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i{}^T. \tag{3.4.2}$$

Transponirana matrica D^T matrice D je dijagonalna matrica dimenzija $m \times n$ s vrijednostima $\sigma_i = d_{i,i}$ duž dijagonale. Umnožak DD^T je $m \times m$ matrica s vrijednostima $d_{i,i}^2$ duž dijagonale. Pseudo-inverz, $D^\dagger = \left(d_{i,j}^\dagger\right)$, matrice D je $n \times m$ dijagonalna matrica s vrijednostima

$$d_{i,j}^\dagger = \begin{cases} 1/d_{i,i} & ako\ je\ d_{i,i} \neq 0 \\ 0 & ako\ je\ d_{i,i} = 0 \end{cases}. \tag{3.4.3}$$

Pseudo-inverz matrice J je jednak

$$J^\dagger = VD^\dagger U^T. \tag{3.4.4}$$

Prema tome,

$$J^{\dagger} = \sum_{i=1}^{r} {\sigma_i}^{-1} \boldsymbol{v}_i {\boldsymbol{u}_i}^T . \tag{3.4.5}$$

DLS metoda je lakša za razumjeti s dekompozicijom singularnih vrijednosti. Matrica $JJ^T + \lambda^2 I$ je jednaka

$$JJ^T + \lambda^2 I = (UDV^T)(VD^T U^T) + \lambda^2 I = U(DD^T + \lambda^2 I)U^T. \tag{3.4.6}$$

Matrica $DD^T + \lambda^2 I$ je dijagonalna matrica s vrijednostima $\sigma_i^2 + \lambda^2$. Očito, $DD^T + \lambda^2 I$ nije singularna, i njezin inverz je $m \times m$ dijagonalna matrica s vrijednostima $(\sigma_i^2 + \lambda^2)^{-1}$ koje nisu nula. Onda je

$$J^T(JJ^T + \lambda^2 I)^{-1} = VD^T(DD^T + \lambda^2 I)^{-1}U^T = VEU^T, \tag{3.4.7}$$

gdje je E $n \times m$ dijagonalna matrica s vrijednostima

$$e_{i,i} = \frac{\sigma_i}{\sigma_i^2 + \lambda^2}. \tag{3.4.8}$$

Rješenje DLS metode može se izraziti u obliku

$$J^T(JJ^T + \lambda^2 I)^{-1} = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \boldsymbol{v}_i {\boldsymbol{u}_i}^T . \tag{3.4.9}$$

Ako usporedimo jednadžbe (3.4.5) i (3.4.9) jasna je veza između pseudo-inverzne metode i DLS metode. U oba slučaja J je „okrenuta“ izrazom $\sum_i \tau_i \, \boldsymbol{v}_i {\boldsymbol{u}_i}^T$. Za pseudo-inverznu metodu τ_i je samo ${\sigma_i}^{-1}$ $(0^{-1} = 0)$, dok je za DLS metodu $\tau_i = \sigma_i/(\sigma_i^2 + \lambda^2)$. Pseudo-inverzna metoda je nestabilna kako se σ_i približava nuli. Zapravo kod singularnosti vrijednosti σ_i su jednaki nuli.

Za vrijednosti σ_i koje su velike u usporedbi s λ, DLS metoda nije puno različitija od pseudo-inverzne metode jer za velike vrijednosti σ_i, $\sigma_i/(\sigma_i^2 + \lambda^2) \approx {\sigma_i}^{-1}$. Ali, kad je σ_i jednakog reda veličine kao i λ ili manja, tada vrijednosti ${\sigma_i}^{-1}$ i $\sigma_i/(\sigma_i^2 + \lambda^2)$ divergiraju. Zaista za sve $\lambda > 0, \sigma_i/(\sigma_i^2 + \lambda^2) \to 0$ kad $\sigma_i \to 0$. DLS metoda se ponaša slično pseudo-inverznoj metodi van

singulariteta i efikasno poboljšava performanse pseudo-inverzne metode blizu singulariteta.

3.5. CCD

CCD algoritam (eng. „Cyclic-Coordinate Descent“) je iterativni optimizacijski postupak koji početni problem postavlja na slijedeći način. Pretpostavimo da se kinematički lanac sastoji od n zglobova. Točka $\boldsymbol{P_k} = (x_k, y_k, z_k)$ označava poziciju krajnjeg djelovatelja u njegovom trenutnom stanju, a točka $\boldsymbol{P_c} = (x_c, y_c, z_c)$ označava ciljnu poziciju koju želimo dohvatiti. Rješavanje inverzne kinematike se može svesti na definiranje funkcije pogreške, odnosno kvadrata udaljenosti između te dvije točke:

$$E(\boldsymbol{q}) = \left|\left| \boldsymbol{P_c} - \boldsymbol{P_k} \right|\right|^2 \tag{3.5.1}$$

i zatim primjenjivanja nekog od optimizacijskih algoritama za nalaženje minimuma te funkcije. Stanje $\boldsymbol{q}$ za koje je funkcija pogreške minimalna je stanje u kojem je djelovatelj najbliži ciljnoj poziciji. U slučaju da je cilj dohvatljiv, to je upravo konfiguracija kinematičkog lanca koja taj cilj i dohvaća.

Algoritam se može proširiti i na zadovoljavanje uvjeta orijentacije, a ne samo položaja, tako da se definira i funkcija pogreške orijentacije koja se sumira s pogreškom položaja te se ukupna pogreška pokušava minimizirati. Uvjeti orijentacije nisu proučavani jer se rijetko pojavljuju u praktičnoj upotrebi.

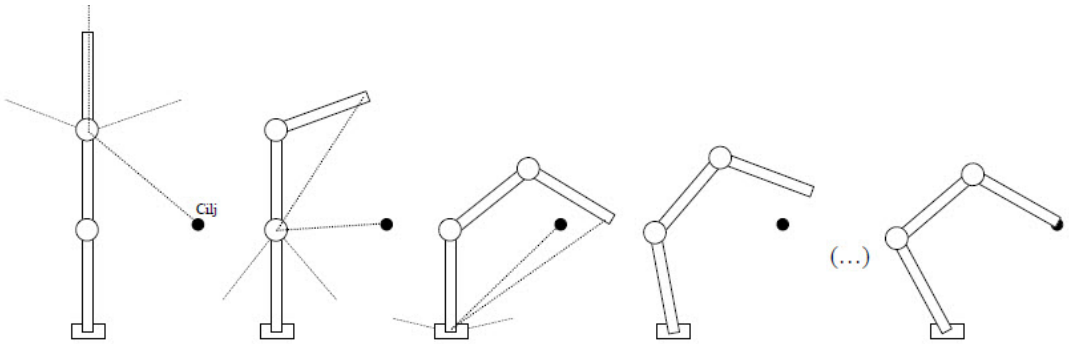
Ono što razlikuje ovaj algoritam od rješavanja Jakobijeve matrice i što algoritmu daje ime, je činjenica da se zglobovi obilaze sekvencijalno, odnosno jedan po jedan. Time se problem optimizacije višedimenzionalne funkcije svodi na sekvencijalno optimiranje jednodimenzionalnih funkcija.

Algoritam u svakoj iteraciji polazi od najudaljenijeg zgloba n u kinematičkom lancu i prelazi redom sve zglobove do korijena, u svakom koraku optimizirajući funkciju pogreške, odnosno zakrećući trenutni zglob u onaj položaj u kojem će djelovatelj biti što bliže ciljnoj poziciji. Svaka

promjena trenutnog zgloba $/i/$ utječe i na sve prethodno posjećene zglobove $/i/+1,\ /i/+2$, itd.

Nakon jedne iteracije, odnosno nakon jednog obilaska svih zglobova u lancu, djelovatelj će biti u poziciji koja je bliža ciljnoj točki, ali za dostizanje optimalnog položaja (dohvatljive točke najbliže ciljnoj) potrebno je više iteracija. Empirija pokazuje da je, ovisno o početnoj konfiguraciji kinematičkog lanca, 10 do 50 iteracija dovoljno za dostizanje ciljne točke s određenom zadanom preciznošću.

Za jednostavan primjer manipulatora s tri zgloba (s ograničenim pokretima), algoritam radi na slijedeći način:



Slika 3.5.1 Prva četiri koraka algoritma i konačno stanje dohvaćanja cilja.

U prvom koraku, zadnji zglob u lancu se postavlja u optimalan položaj. Zglob je ograničen (tanke crtane linije) pa se ne može okrenuti do položaja apsolutno najbližeg cilju, ali dolazi do svog maksimalnog otklona. Zatim se prethodni, tj. srednji zglob postavlja tako da se krajnji djelovatelj što više približi cilju. Nakon njega zakreće se i prvi zglob, odnosno korijen kinematičkog lanca, i to u smjeru cilja, jer je u tom smjeru optimalni položaj. U idućoj iteraciji, ponovo se podešava daljnji zglob, itd.

4. Implementacija

Implementirati ćemo četiri metode rješavanja problema inverzne kinematike.

- CCD
- Transponirani Jakobijan
- Pseudoinverzni Jakobijan
- DLS

Svakoj metodi prosljeđuju se sljedeći parametri:

- Hijerarhijski kinematički lanac
- Ciljna pozicija
- Broj maksimalnih pokušaja
- Maksimalna dozvoljena udaljenost ciljne pozicije i pozicije krajnjeg djelovatelja koja se smatra uspješnim dohvaćanjem
- Ograničenja rotacije zglobova ako su ograničenja uključena

U metodama transponirani Jakobijan, pseudo-inverzni Jakobijan i DLS dodatno se prosljeđuje:

- Alpha, faktor kojim se smanjuju promjene kutova u jednom koraku simulacije

DLS metoda prima još parametre:

- Lambda, potrebna za SVD metodu
- Maksimalnu promjenu kuta u jednom koraku simulacije

Implementaciju svake metode opisati ćemo s kratkim pseudokodom. Radi sličnosti pseudokodova posljednje tri metode, sva tri pseudokoda su ujedinjena s naznačenim razlikama u metodama.

CCD

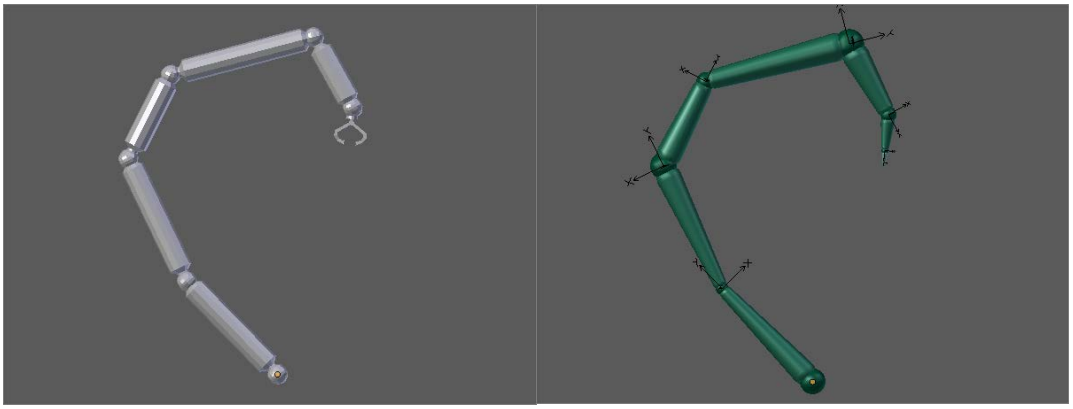
```
cilj = parametri->cilj;
maksBrojPokusaja = parametri->maksBrojPokusaja * BrojKostiULancu();
maksUdaljenost = parametri->maksUdaljenost;
ogranicenjaUkljucena = parametri->ogranicenjaUkljucena;

trenutniBrojPokusaja = 0;
trenutnaKost = RoditeljKrajnjegDjelovatelja();
ponavljaj
{
   trenutniKraj = OdrediPozicijuKrajnjegDjelovatelja();
   pozicijaKosti = OdrediPozicijuKosti(trenutnaKost);
   udaljenost = OdrediUdaljenostCiljaITrenutnogKraja();

   ako je (udaljenost < maksUdaljenost)
   {
      kutRotacije = OdrediKutRotacije(pozicijaKosti, trenutniKraj, cilj);
      osRotacije = OdrediVektorRotacije(pozicijaKosti, trenutniKraj, cilj) ;
      RotirajKost(kutRotacije, osRotacije);

      ako je (ogranicenjaUkljucena == true)
         ProvjeriOgranicenjaZgloba(trenutnaKost, parametri->ogranicenjaKosti);

      trenutnaKost = RoditeljIliRoditeljKrajnjegDjelovatelja(trenutnaKost);
   }

   trenutniKraj = OdrediPozicijuKrajnjegDjelovatelja();
   udaljenost = OdrediUdaljenostCiljaITrenutnogKraja();
   trenutniBrojPokusaja = trenutniBrojPokusaja + 1;

} dok je (trenutniBrojPokusaja < maksBrojPokusaja ili
          udaljenost > maksUdaljenost)

ako je (trenutniBrojPokusaja <= maksBrojPokusaja)
   problemRjesen = true;
inace
   problemRjesen = false;
```

```
OdrediKutRotacije(pozicijaKosti, trenutniKraj, cilj)
{
   trenutniVektor = trenutniKraj – pozicijaKosti;
   ciljniVektor = cilj – pozicijaKosti;
   kosinusKuta = SkalarniUmnozak(trenutniVektor, ciljniVektor);
   vrati IzracunajKut(kosinusKuta);
}
OdrediVektorRotacije(pozicijaKosti, trenutniKraj, cilj)
{
   trenutniVektor = trenutniKraj – pozicijaKosti;
   ciljniVektor = cilj – pozicijaKosti;
   vrati VektorskiUmnozak(trenutniVektor, ciljniVektor);
}
```

Transponirani Jakobijan (1), Pseudo-inverzni Jakobijan (2), DLS (3)

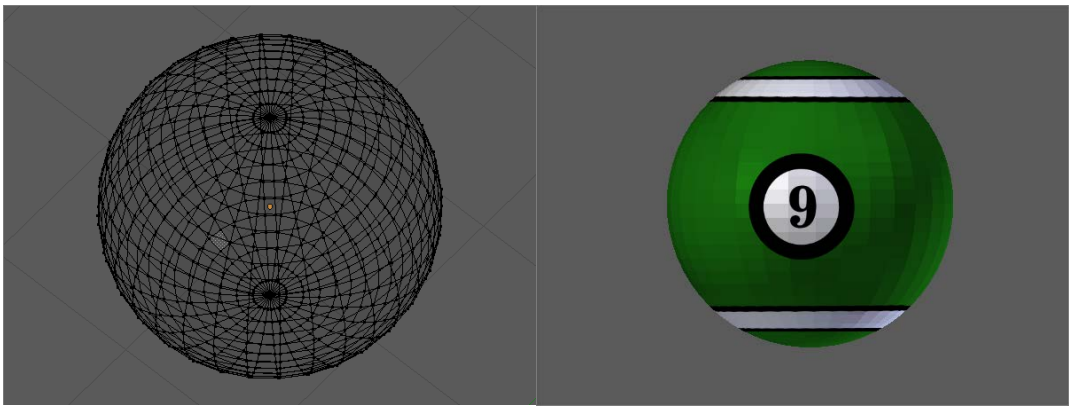
```
cilj = parametri->cilj;
maksBrojPokusaja = parametri->maksBrojPokusaja;
maksUdaljenost = parametri->maksUdaljenost;
ogranicenjaUkljucena = parametri->ogranicenjaUkljucena;
alpha = parametri->alpha;
lambda = parametri->lambda;
maksPromjenaKuta =  parametri-> maksPromjenaKuta;

trenutniBrojPokusaja = 0;
ponavljaj
{
   trenutniKraj = OdrediPozicijuKrajnjegDjelovatelja();
   udaljenost = OdrediUdaljenostCiljaITrenutnogKraja();
   snaga = cilj – trenutniKraj;
   ako je (udaljenost < maksUdaljenost)
   {
      za svaku (kost u parametri->kostiLanca)
      {
         osRotacije[kost] = OdrediVektorRotacije(kost, trenutniKraj, cilj);
         element = IzracunajElementJakobijana(kost, osRotacije, cilj);
         jakobijan.DodajElement(element);
      }

      (1) transpose = IzracunajTransponiranuMatricuJakobijana(jakobijan);
      (1) kutRotacije = transpose * snaga; //Δθ
      (2) pseudoinverz = IzracunajPseudoInverznuMatricuJakobijana(jakobijan);
      (2) kutRotacije = pseudoinverz * snaga; //Δθ
      (3) kutRotacije = OdrediKuteveRotacijeSVDMetodom(jakobijan, lambda);
      (3) kutRotacije = ProvjeriMaksPromjene(kutRotacije, maksPromjenaKuta);

      za svaku (kost u parametri->kostiLanca)
      {
         RotirajKost(kutRotacije[kost] * alpha, osRotacije[kost]);
         ako je (ogranicenjaUkljucena == true)
         ProvjeriOgranicenjaZgloba(trenutnaKost, parametri->ogranicenjaKosti);
      }
   }
   trenutniKraj = OdrediPozicijuKrajnjegDjelovatelja();
   udaljenost = OdrediUdaljenostCiljaITrenutnogKraja();

   trenutniBrojPokusaja = trenutniBrojPokusaja + 1;
} dok je (trenutniBrojPokusaja < maksBrojPokusaja ili
          udaljenost > maksUdaljenost)

ako je (trenutniBrojPokusaja <= maksBrojPokusaja)
       problemRjesen = true;
inace
       problemRjesen = false;
```

Napraviti ćemo model s kojim ćemo testirati implementaciju gore navedenih metoda rješavanja problema inverzne kinematike. Najjednostavniji model koji će poslužiti svrsi je model robotske ruke. U programskom alatu Blender modelirati ćemo robotsku ruku koja je zapravo jedan hijerarhijski lanac sa sedam segmenata. Prvih pet segmenata su dijelovi robotske ruke. Šesti segment je hvataljka, a zadnji segment predstavlja krajnji djelovatelj.



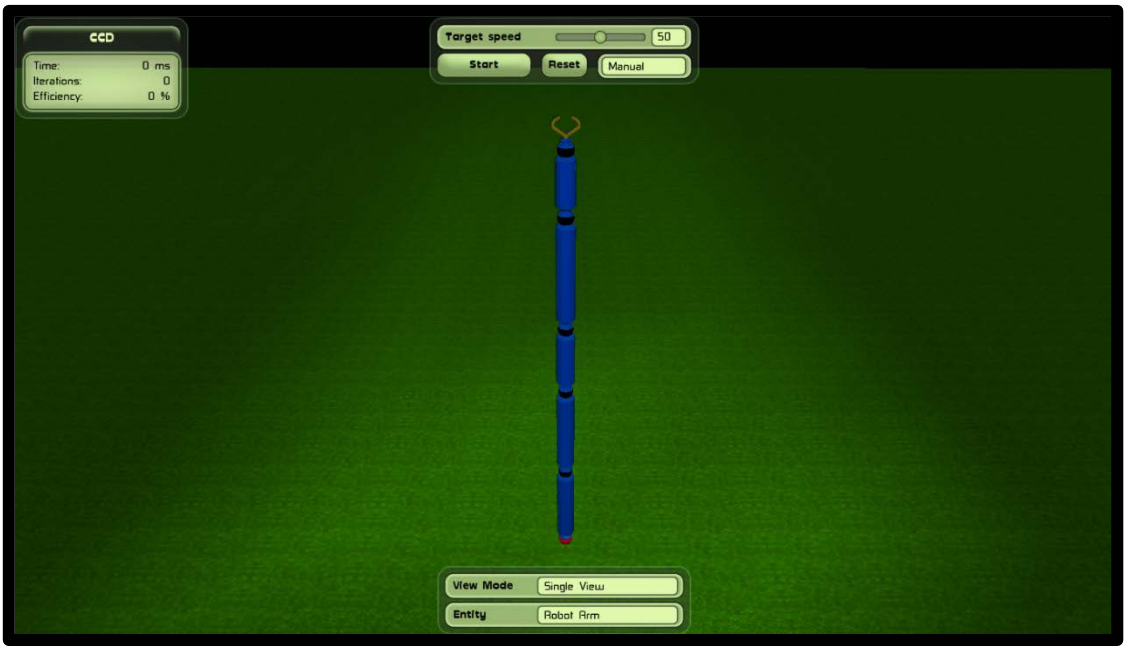
Slika 4.1 a) mreža robotske ruke b) kostur robotske ruke

Želimo da robotska ruka prati neki predmet, tj. da ga pokušava dohvatiti. Napraviti ćemo neku jednostavnu kuglu koja će predstavljati ciljnu poziciju hvataljke robotske ruke.



Slika 4.2 a) mreža kugle b) model kugle s teksturom

Najprije ćemo se upoznati sa sučeljem i mogućnostima aplikacije, zatim ćemo napraviti usporedbu rezultata simulacije te komentirati rezultate.



Slika 4.3 Početni zaslon nakon pokretanja aplikacije

U aplikaciju su učitana dva modela za testiranje metoda rješavanja problema inverzne kinematike koje je moguće odabrati u padajućem izborniku “Entity“ na dnu ekrana Slika (4.4a).

Slika 4.4 a) Odabir aktivnog modela b) Odabir način rada

Podržana su tri načina rada Slika (4.4b). Prvi način rada je „Single View“ u kojem je učitana samo jedna instanca odabranog modela, zatim „Compare Two“ i „Compare Four“ za usporedbu dvije, odnosno četiri instance modela istovremeno. Za svaku instancu moguće je pratiti statistiku Slika (4.5a) koja mjeri prosječno vrijeme rješavanja problema, zatim prosječan broj potrebnih iteracija te efikasnost koja pokazuje koliki je postotak uspješnog rješavanja problema. Statistika se vodi za zadnjih 1000 simulacijskih koraka (1 simulacijski korak = 1 „frame“), dok se na zaslonu ažurira svake sekunde.

Slika 4.5 a) Statistika s prikazom parametara b) Mijenjanje parametara simulacije

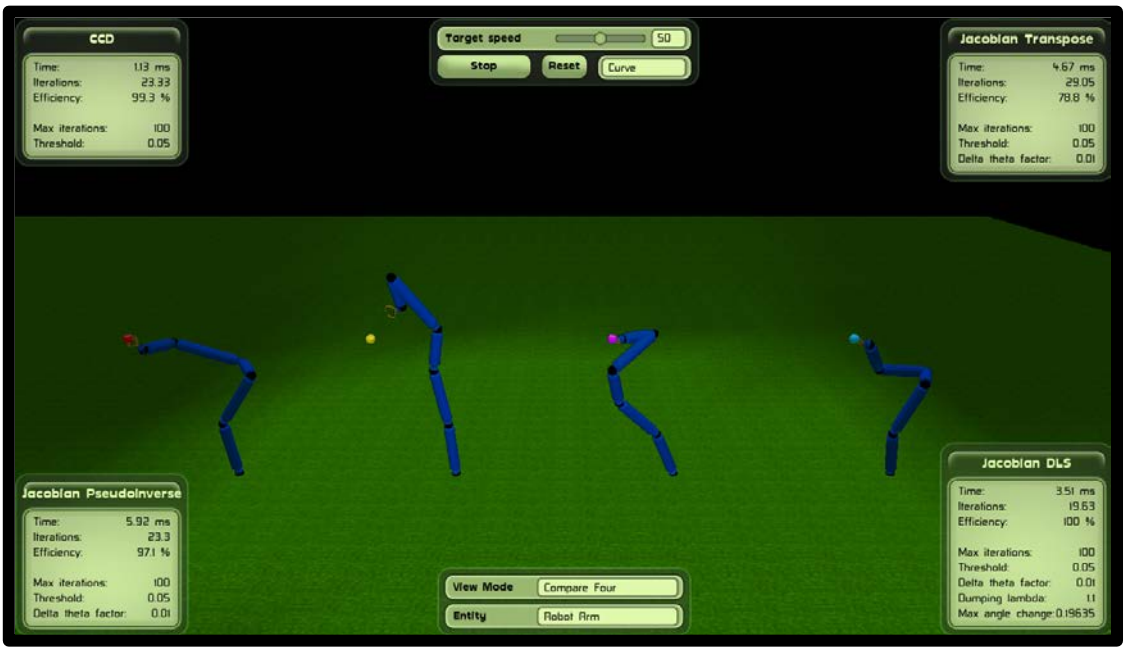
Klikom na ime trenutne metode (Slika 4.5a) pojavljuje se odnosno nestaje popis svih parametara koji utječu na tu metodu. Klikom na parametre pojavljuje se novi izbornik (Slika 4.5b) u kojem je moguće mijenjati iste i na taj način utjecati na performanse rješavanja problema inverzne kinematike. Metodu rješavanja i njezine parametre moguće je birati za svaku prikazanu instancu modela i na taj način dobiti željenu usporedbu metoda.

Slika 4.6 Usporedba četiri metode rješavanja problema inverzne kinematike

Kako bi bilo jednostavnije usporediti metode, ugrađene su pomoćne kontrole za animaciju. Animacija se može odvijati u tri načina rada. Ručno mijenjajući položaj modela kugle, zatim automatski po krivulji ili slučajnim odabirom. U slučaju druge dvije opcije moguće je pokrenuti i zaustaviti animaciju klikom na „Start/Stop“. U svakom trenutku moguće je vratiti scenu u prvobitni položaj klikom na „Reset“ te mijenjati brzinu promjene položaja kugle (Slika 4.7).

Slika 4.7 Kontrole za simulaciju

Za usporedbu ćemo koristiti treći način rada u kojem ćemo vršiti usporedbu metoda na četiri instance modela robotske ruke. Svakoj instanci dodijeliti ćemo drugu metodu rješavanja problema inverzne kinematike te za početak ostaviti zadane parametre bez ograničenja rotacije zglobova. Pokrenut ćemo animaciju u drugom načinu rada, dakle kugla će mijenjati položaj po krivulji. U drugom koraku odabrati ćemo simulaciju u kojoj se položaj kugle određuje slučajnim odabirom. Ponoviti ćemo prvi i drugi korak, ali ovaj put s ograničenjima rotacije zglobova na $\pm 45°$ po x i z osi. Rezultati mjerenja nalaze se u tablici 4.1.



Slika 4.9 Usporedba performansi metoda rješavanja inverzno kinematičkog problema

Parametri:

Broj iteracija: 100

Najmanja udaljenost za uspješno rješavanje problema: 0.05

Faktor smanjivanja promjene kute po simulacijskom koraku: 0.01

Odbacujuća konstanta (lambda): 1.1

Maksimalna promjena kuta: $\pi/16$

Usporedba	1.			2.			3.			4.		
Elementi mjerenja	t	N	η	t	N	η	t	N	η	t	N	η
CCD	1.1	20	99	5.0	210	75	5.0	83	92	16.7	232	74
Transponirani Jakobijan	5.0	28	78	15.4	92	15	45.6	93	8	47.8	99	3
Pseudoinverzni Jakobijan	7.0	25	96	21.8	79	47	23.8	40	81	51.9	88	29
DLS	3.5	18	100	19	100	0	27.8	55	69	50	99	2

Tablica 4.1 Rezultati usporedbi

CCD metoda najbrže konvergira prema rješenju što se najbolje vidi u drugoj i četvrtoj usporedbi. Naime u svakom koraku animacije ciljna pozicija je određena slučajnim odabirom što znači da često dolazi do velikih pomaka ciljne pozicije. Da bi model dohvatio ciljnu poziciju potrebne su velike promjene u konfiguraciji modela. Dok ostale metode imaju poteškoća s time, CCD metoda ima 75% uspješnih pokušaja te zauzima 3 puta manje resursa odnosno izvršava se 3 puta brže. Nedostatak CCD metode je što pri animaciji dolazi do velikih skokova u konfiguraciji, što rezultira neprirodnim pokretima. Transponirani Jakobijan daje najslabije rezultate. Izvršava se u većini slučajeva najdulje, a daje najmanji udio uspješnih rješenja problema. Model vrlo često „trza“ dok pokušava dosegnuti ciljnu poziciju koja nije u blizini, dakle kad je Jakobijan blizu singulariteta. Pseudo-inverzni Jakobijan daje bolje rezultate od transponiranog Jakobijan, ali i dalje lošije rezultate od CCD ili DLS metode. Pseudo-invernzna metoda ima veću toleranciju na udaljenije ciljne pozicije, ali model svejedno trza kada je ciljna pozicija nedohvatljiva ili gotovo nedohvatljiva. DLS metoda daje najbolje rezultate, pokreti su prirodni i uglađeni, međutim poprilično se dugo izvršava. Još jedan nedostatak DLS metode je slaba konvergencija prema rješenju problema

(druga i četvrta usporedba) što ovisi o odabiru konstante za odbacivanje. Njezina optimalna vrijednost ovisi od modela do modela i najbolje ju je interaktivno odrediti.

Slika 4.9 Model lava u opuštenoj pozi

Slika 4.10 Model lava dohvaća kuglu

5. Zaključak

Transponirani Jakobijan se najlošije ponaša u usporedbi s ostalim metodama. Iza njega slijedi pseudo-inverzni Jakobijan. Problem jedne i druge metode je manji postotak uspješno riješenih problema u usporedbi s druge dvije metode. Model često „trza“ pokušavajući dohvatiti udaljenu ciljnu poziciju i nikada ne dolazi u konfiguraciju u kojoj su svi zglobovi ispruženi. Međutim, metode su relativno jednostavne za implementirati i daju dovoljno dobre rezultate za modele u kojima nije česta konfiguracija s ispruženim kutovima zglobova. CCD metoda je najjednostavnija za implementirati, ujedno koristi najmanje resursa, najbrže se izvršava i vrlo brzo konvergira prema rješenju problema. Jedini nedostatak CCD metode su ponekad neprirodni pokreti radi skokova između konfiguracija. DLS metoda daje najbolje rezultate, ako su potrebni prirodni i uglađeni pokreti modela. Nedostatak DLS metode je nešto slabija konvergencija prema rješenju problema. Implementacija je nešto zahtjevnija od ostalih metoda i potrebno je odrediti konstantu za odbacivanje za svaki model kako bi ista davala što bolje rezultate.

Za svaku metodu korisno je približiti ciljnu poziciju trenutnoj poziciji krajnjeg djelovatelja za svaki korak animacije kako bi se izbjegle velike promjene u kutovima i time poboljšale performanse nalaženja rješenja problema inverzne kinematike.

Literatura

[1] Direct Kinematics, http://www-clmc.usc.edu/~cs545/Lecture_VIII.pdf, 14.6.2010

[2] Forward Kinematics: The Denavit-Hartenberg Convention, http://www.cs.duke.edu/brd/Teaching/Bio/asmb/current/Papers/chap3-forward-kinematics.pdf, 14.6.2010.

[3] Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods, http://math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf, 15.6.2010.

[4] Mike Tabaczynski - „Jacobian Solutions to the Inverse Kinematics Problem", http://www.eecs.tufts.edu/~mtabac0a/IK/Proiect.pdf, 15.6.2010.

[5] C. W. Wampler, *Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods*, IEEE Transac- tions on Systems, Man, and Cybernetics, 16 (1986), str. 93-101.

[6] Y. Nakamura and H. Hanafusa, I*nverse kinematics solutions with singularity robustness for robot manipulator control*, Journal of Dynamic Systems, Measurement, and Control, 108 (1986), str. 163-171.

[7] A. Balestrino, G. De Maria, and L. Sciavicco, *Robust control of robotic manipulators, in Proceedings of the 9th IFAC World Congress*, Vol. 5, 1984, str. 2435-2440.

[8] W. A. Wolovich and H. Elliot, *A computational technique for inverse kinematics, in Proc. 23rd IEEE Conference on Decision and Control*, 1984, str. 1359-1363.

[9] Zeljka Mihajlovic - „Unaprijedna i inverzna kinematika", http://www.zemris.fer.hr/predmeti/ra/predavanja/4 kinemat.pdf, 15.6.2010.

[10] Blender Essential Training, http://www.lynda.com/home/DisplayCourse.aspx?lpk2=740, 10.6.2010.

Sažetak

INTERAKTIVNI INVERZNI KINEMATIČKI MODEL

Glavni zadatak ovoga rada je upoznavanje s inverznom kinemtikom te korištenje iste u ostvarenju interaktivnog modela. Rad se sastoji od pet poglavlja kroz koja se čitatelju pokušava objasniti ostvarenje glavnog zadatka.

Drugo poglavlje se bavi općenito kinematikom i kinematičkim strukturama, posebice hijerarhijskim kinematičkim lancima. Kinematika se dijeli na dvije grane: unaprijedna i inverzna kinematike. Obje grane su detaljno obrađene. Objašnjene su razlike između te dvije grane, koje su prednosti inverzne kinematike nad unaprijednom kinematikom i zašto ju koristimo.

U trećem poglavlju se upoznajemo s metodama rješavanja problema inverzne kinematike. Njihovim prednostima, nedostacima te matematičkom pozadinom.

U četvrtom poglavlju čitatelja se upoznaje s rezultatima ovoga rada. Opisani su postupci primjene metoda za rješavanje problema inverzne kinematike na upravljanje modela za ostvarenje interaktivnog inverzno kinematičkog modela.

Ključne riječi: inverzna kinematika, CCD, DLS, transponirani Jakobijan, pseudo-inverzni Jakobijan, model

Abstract

INTERACTIVE INVERSE KINEMATIC MODEL

The main task of this paper is to introduce the inverse kinematic and learn how to use it to achieve an interactive model. The paper consists of five sections through which it's explained the main achievement of the task.

The second chapter deals with general kinematics and kinematic structures, especially the hierarchical kinematic chains. Kinematics is divided into two branches: forward and inverse kinematics. Both branches are analyzed in detail and it's explained what are advantages of inverse kinematics and why we use it.

The third chapter is introduced to the methods for solving inverse kinematics problem, their advantages, disadvantages and mathematical background.

In the fourth chapter, the reader becomes familiar with the results of this work. The methods for solving inverse kinematic problem are applied to interactively control the inverse kinematic model.

Keywords: Inverse kinematic, CCD, DLS, Jacobian transpose, pseudo-inverse Jacobian, model