

4. Direktna i inverzna kinematika

- zadavanje pokreta objekta sačinjenih od kinematičkih lanaca
- KINEMATIKA
 - grana mehanike - pokreti objekata neovisni o silama koje uzrokuju pokrete
 - kinem. lanac - položaj objekta je funkcija kutova između segmenata objekta
 - animacija - interpolacija između zadanih ključnih položaja
 - hijerarhijski modeli (model čovjeka, životinja, robot)
 - <https://schteppe.github.io/p2.js/demos/ragdoll.html> <https://openprocessing.org/sketch/375472/>
<https://codepen.io/BagIdea/full/LYRPZew> <http://www.visualiser.fr/rag/default.htm>
- DINAMIKA
 - djelovanje sila u ostvarivanju pokreta (gravitacija, inercija ...)
 - fizikalno temeljeni algoritmi
- objekti kao što su ljudi ili životinje imaju krutu unutrašnju strukturu i mišićnu strukturu izvana, pa se na taj način i grade
- NPR:
 - <http://pablotheflamingo.com/> <https://codepen.io/gordonnl/pen/LvJiG>
 - <http://schteppe.github.io/ammo.js-demos/demos/RagdollDemo/three.html>

4.1 Hijerarhijski kinematički modeli

Struktura

- organizacija elemenata u hijerarhijsku strukturu

Artikuliranje

- promjene u orijentaciji elemenata strukture

Robotika

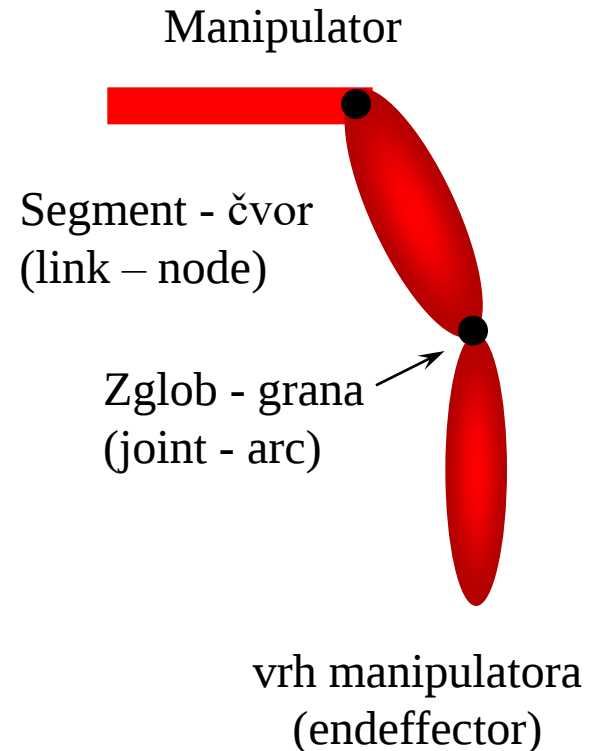
- većina termina preuzeta iz robotike

Manipulator

- zglob – ne uzimamo dimenziju u obzir
- segment – kruti element
- niz segmenata zglobovima povezanih u lanac
- ponekad se zglob i segment promatraju kao cjelina, odnosno jedan zglobni element

http://creature.kestrelmoon.com/demo_raptor.html - neki zglobovi su „u zraku”

<http://cs.wellesley.edu/~cs307/threejs/demos/BasicModeling/Leg1.shtml>



Vrste zglobova

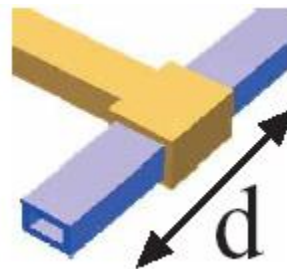
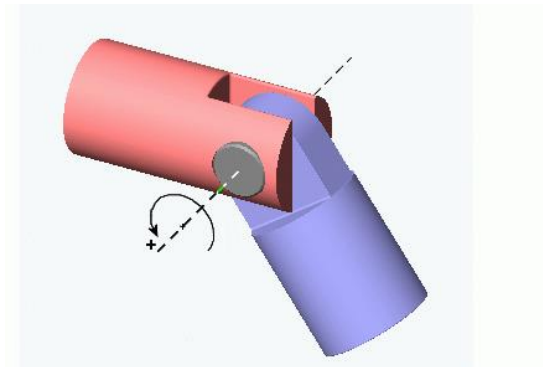
Stupanj slobode u zglobu (DOF)

- jedan rotacijski stupanj slobode u zglobu (engl. Revolute Joint)
- jedan translacijski stupanj slobode u zglobu (engl. Prismatic Joint)

Složeni zglobovi

- povezivanje n 1DOF zglobova sa segmentima veličine nula

http://www.realtimerendering.com/udacity/?load=unit4/unit4-robot_arm_extended_exercise.js

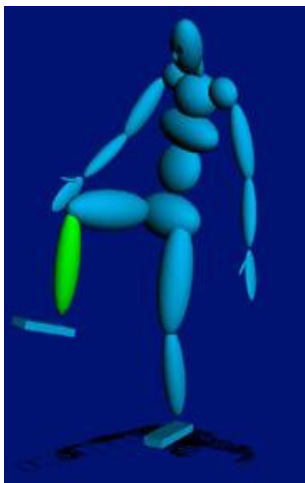


Rotacijski stupanj slobode u zglobu (uz ograničenje kutova)

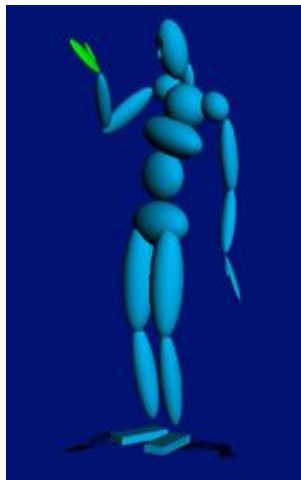
- dimenzije zgloba obično zanemarujemo

<http://www.kineman.com/new/webapp/index.php>

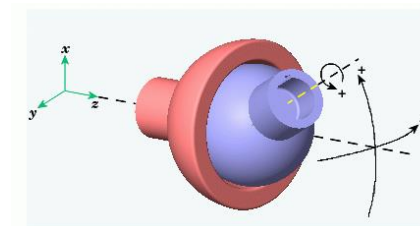
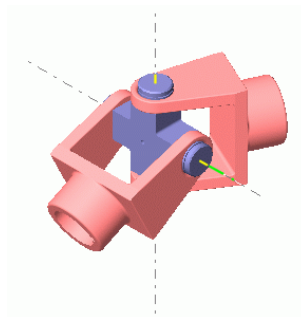
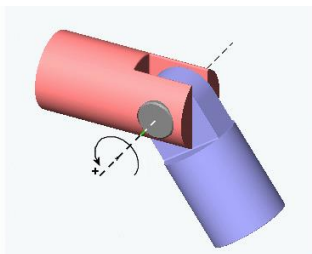
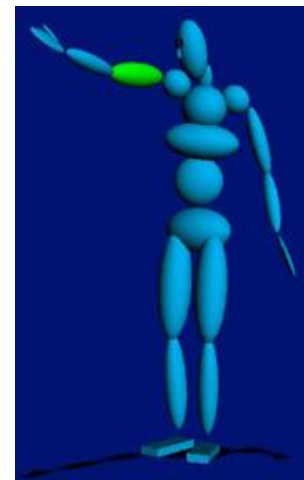
1DOF



2DOF



3DOF

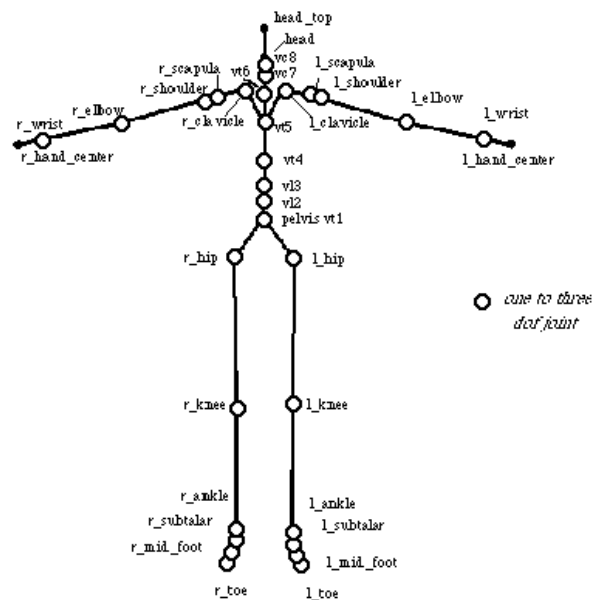


Izgradnja hijerarhijskih modela

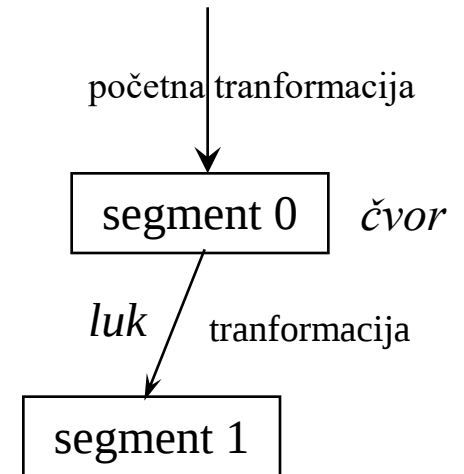
- objekt
- apstraktna reprezentacija objekta ovisno o modelu
<http://3dflashlo.free.fr/webgl2013/index.html>
- stablo (čvor – segment, luk-zglob)



objekt

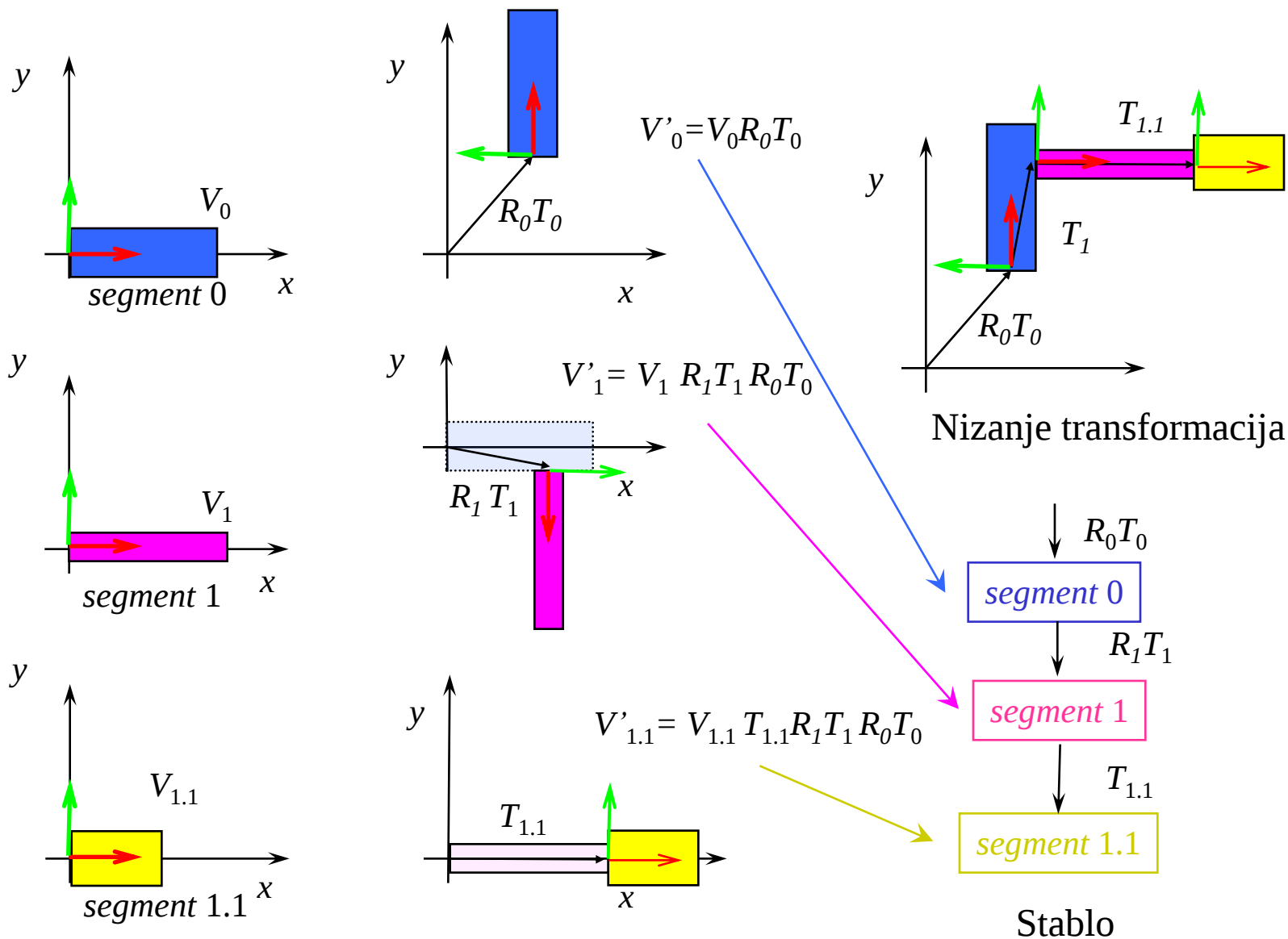


apstraktna reprezentacija

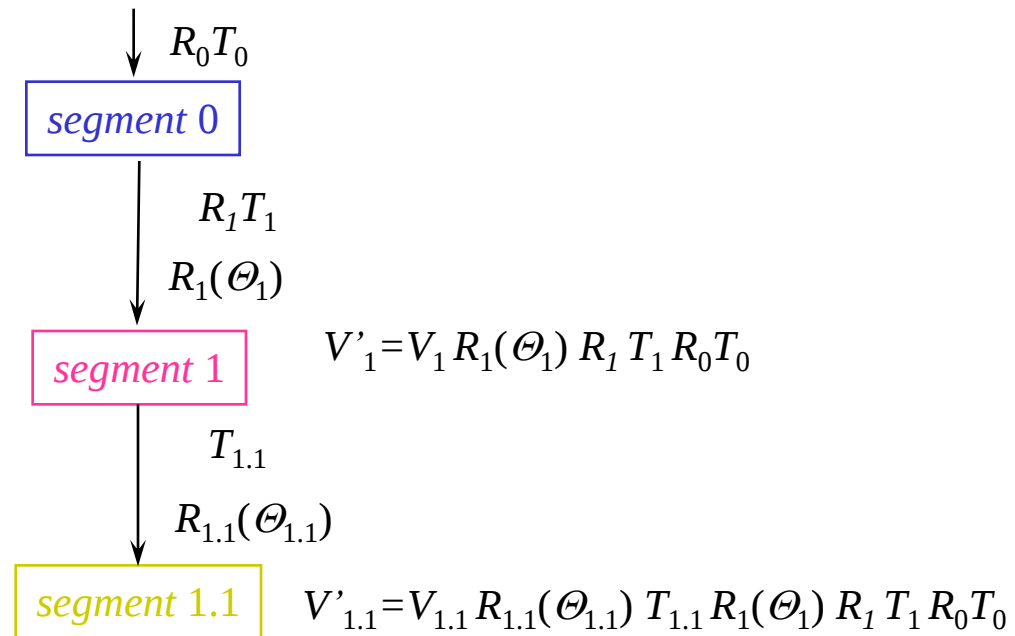
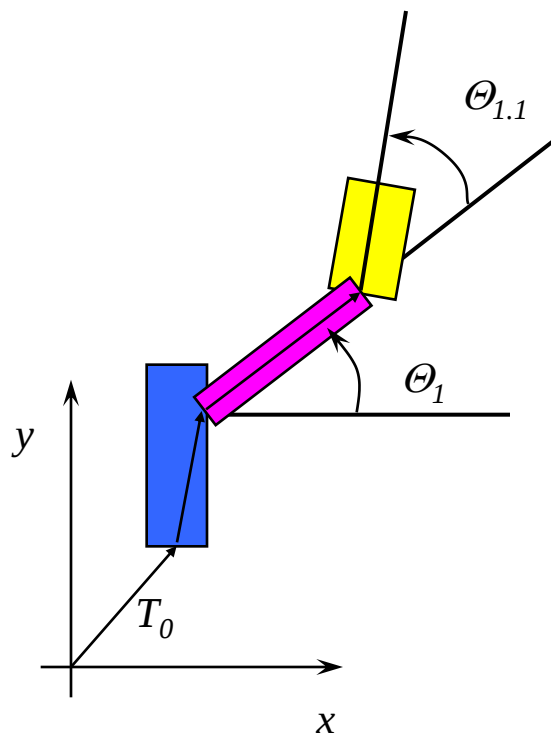


stablo

lokalni koordinatni sustavi: <http://jyunming-chen.github.io/tutsplus/tutsplus18.html>



- nakon definiranja statičkog položaja objekta definiramo pomake u zglobu:
 - rotacija se obavlja prije konstantne (lokalne) transformacije u luku
 - grananje stabla – više jednostrukih lanaca
 - nasljeđivanje transformacija s roditelja na djecu
 - jednostavni pokreti: <http://schteppe.github.io/p2.js/demos/ragdoll.html>

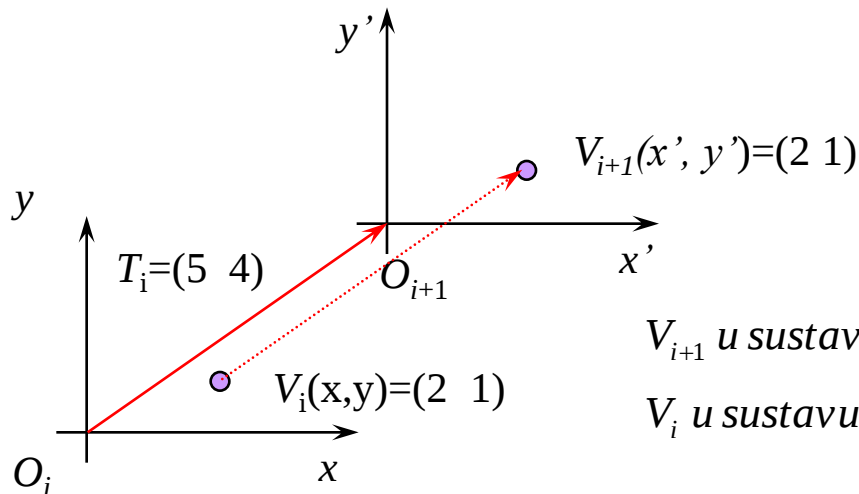


Stablo uz rotacije u zglobovima

- hijerarhijski model
 - ishodišta lokalnih koordinatnih sustava određuju se na osnovi osi rotacije u pojedinom zglobu (os je obično poravnata s x , y ili z osi) [DOG 5.blend](#)
 - zglob s 2 i više stupnja slobode ostvaruje se kao niz segmenata duljine 0 s 1 stupnjem slobode (ovisno o fizičkoj izvedbi)
 - mogu se koristiti višestruko referenciranje za pridjeljivanje svojstava materija pojedinim segmenata (boja, materijal, tekstura, vidljivost, oblik) ili se atributu nasljeđuju
 - struktura čvorova i lukova može biti načinjena odvojeno, tako da se pokreti ostvaruju nezavisno o izgrađenom modelu (isti pokreti se izvode na različitim modelima, kloniranje pokreta) <https://www.mixamo.com/#/?page=2&type=Character>
- kada izgradimo hijerarhijski model, načinili smo pasivni strukturni model izgrađeni model je kao lutka bez lutkara (<http://swissarmyman.com/>)
 - potrebno je definirati gibanje koje će “prirodno” izgledati, vrlo težak problem <http://www.visualiser.fr/Babylon/character/default.htm>
 - tehnikama niže i više razine definiramo gibanje
 - kontrola pojedinačnog pokreta zadavanjem kutova u zglobovima
 - kontrola pokreta zadavanje putanja koju prati kraj manipulatora / MoCap
 - zadavanje naredbi oblika: sjedni, šeci, čučni ...

4.2 Kinematika

- DIREKTNA
 - na osnovi stabla i zadanih parametara u zglobovima određuje se položaj
 - transformacijom $\mathbf{T}_i$ određen je položaj objekta V_{i+1} u sustavu O_i roditelja
 - (stog s trenutnom transformacijskom matricom, rekurzivni pozivi)
- INVERZNA
 - vrh manipulatora treba pratiti zadanu putanju, kutovi u zglobovima se računaju
 - inverznom transformacijom $\mathbf{T}_i^{-1}$ određen je položaj V_i u sustavu O_{i+1}



$$V_{i+1} \text{ u sustavu } O_i \quad V_{i+1}(x, y) = V_i(x, y) \mathbf{T}_i = \begin{pmatrix} 7 & 5 \end{pmatrix}$$

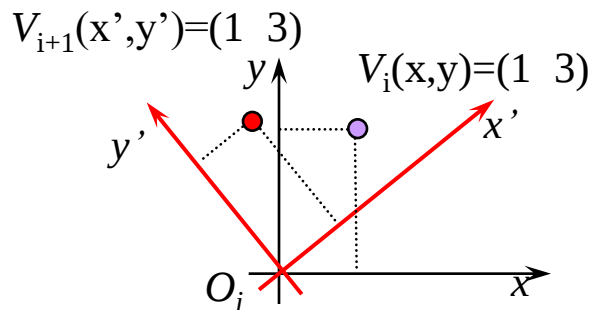
$$V_i \text{ u sustavu } O_{i+1} \quad V_i(x', y') = V_{i+1}(x', y') \mathbf{T}_i^{-1} = \begin{pmatrix} -3 & -3 \end{pmatrix}$$

- točka V_i u sustavu O_i ima koordinate npr. $V_i = (1 \ 3)$

$$V_{i+1} \text{ u sustavu } O_i \quad V_{i+1}(x, y) = V_i(x, y) \mathbf{R}_i = (1 \ 3 \ 1) \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$V_i \text{ u sustavu } O_{i+1} \quad V'_i(x', y') = V_{i+1}(x', y') \mathbf{R}_i^{-1}$$

- rotacija točke matricom $\mathbf{R}$, rotira vektor $\mathbf{R}^{-1}$



$$\mathbf{V}_r = \mathbf{V}_i \mathbf{R}$$

$$\vec{x}' = \mathbf{R}^{-1} \vec{x}$$

- rotiramo jedinične vektore koji razapinju koordinatni sustav

$$\begin{aligned} \vec{x}' &= \mathbf{R}^{-1} \vec{x} \\ \vec{y}' &= \mathbf{R}^{-1} \vec{y} \end{aligned} \quad \begin{bmatrix} \vec{x}' & \vec{y}' \end{bmatrix} = \mathbf{R}^{-1} \begin{bmatrix} \vec{x} & \vec{y} \end{bmatrix} = \mathbf{R}^{-1} \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

- ako znamo novi koordinatni sustav – znamo matricu rotacije
 - koordinatne osi x' i y' određuju matricu rotacije (osi x, y daju jediničnu matricu)
 - lako odredimo za transformirani sustav gdje je točka u prethodnom sustavu
 - npr. $V_{i+1} = (1, 3)$ onda je $V_{i+1} \mathbf{R}^{-1}$ u sustavu O_{i+1}

Denavit-Hartenberg-ova notacija

- robotika – 4 parametra
- relacija između koordinatnih sustava roditelja i djeteta

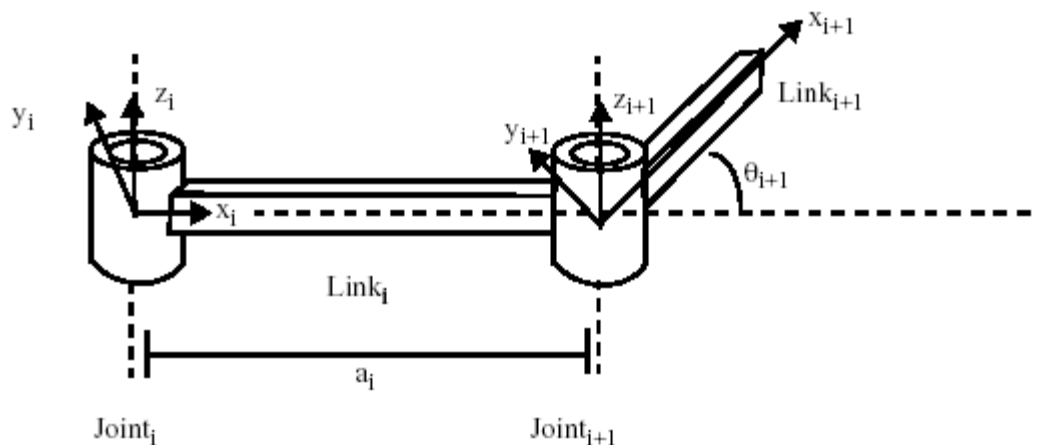
z - os oko koje se obavlja **rotacija**

x - os određuje segment (engl. link)

y - os je $z \times x$

duljina segmenta a_i – udaljenost okomice između z_i do z_{i+1}

kut zgloba θ_{i+1} – kut između x_i i x_{i+1} oko osi z_{i+1}



Denavit-Hartenberg-ova notacija

duljina segmenta

a_{i-1} – duljina okomice između z_{i-1} do z_i

pomak segmenata

d_i – duljina duž z_i osi između sjecišta
- z_i sa a_{i-1} i z_i sa a_i

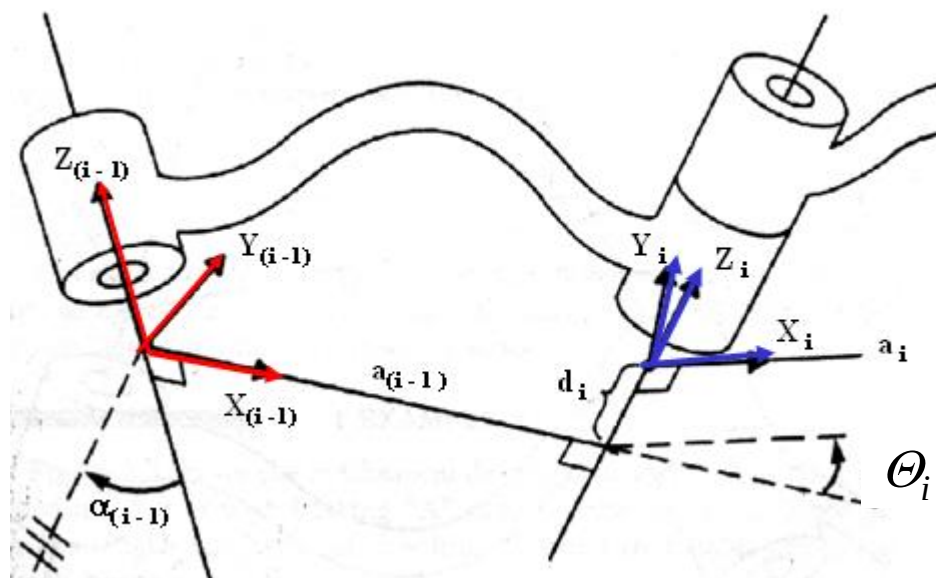
kut uvijanja

α_{i-1} – kut između z_{i-1} i z_i oko osi x_{i-1}

kut zgloba

θ_i – kut između x_{i-1} i x_i oko osi z_i

indeksi!



- transformacija točke iz sustava O_i u sustav O_{i-1} :

<http://folk.ntnu.no/laht/DH/>

$V_{i-1} = T \cdot V_i$ koristi se konvencija „matrica · točka”

$$T = T_x(a_{i-1})R_x(\alpha_{i-1})T_z(d_i)R_z(\theta_i)$$

$$T = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & 0 \\ 0 & \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\Theta_i) & -\sin(\Theta_i) & 0 & 0 \\ \sin(\Theta_i) & \cos(\Theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

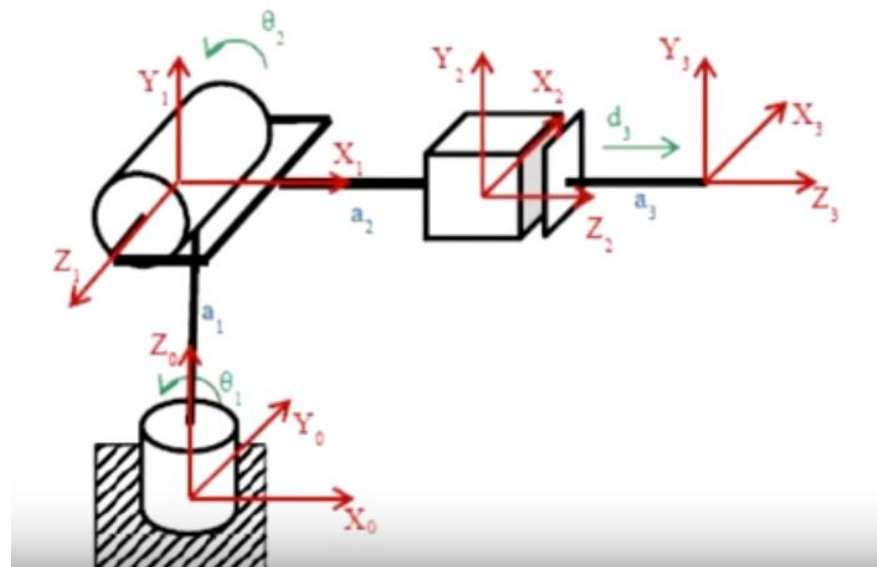
$$T = \begin{bmatrix} \cos(\Theta_i) & -\sin(\Theta_i) & 0 & a_{i-1} \\ \cos(\alpha_{i-1})\sin(\Theta_i) & \cos(\alpha_{i-1})\cos(\Theta_i) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1})\sin(\Theta_i) & \sin(\alpha_{i-1})\cos(\Theta_i) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Složeniji primjer kinematičkog lanca:

- $\Theta_1, \Theta_2, d_3 \dots$ ulazne varijable, kutovi u zglobovima (translacija u zglobovima)
- $x_3, y_3, z_3 \dots$ pozicija vrha manipulatora
- Matrice koja povezuje koordinatne sustave

A_1^0 koordinate u sustavu 1
gledano iz sustava 0

$$F(\Theta) = A_1^0 \cdot A_2^1 \cdot A_3^2 = A_3^0$$



A_1^0	<i>Rotacija oko z_0</i>	<i>Translacija</i>	x_1 y_1 z_1 ↓ ↓ ↓	<i>dinamički dio</i>	<i>statički dio</i>	<i>ukupna matrica</i>
$A_1^0 = \begin{bmatrix} c \Theta_1 & -s \Theta_1 & 0 & 0 \\ s \Theta_1 & c \Theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c \Theta_1 & -s \Theta_1 & 0 & 0 \\ s \Theta_1 & c \Theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c \Theta_1 & 0 & s \Theta_1 & 0 \\ s \Theta_1 & 0 & -c \Theta_1 & 0 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

- kratice - c cos, s sin.

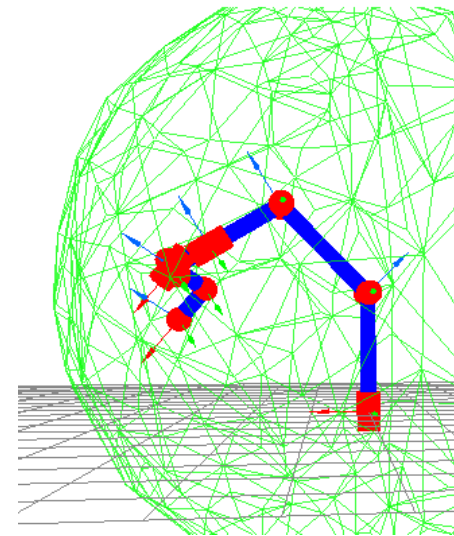
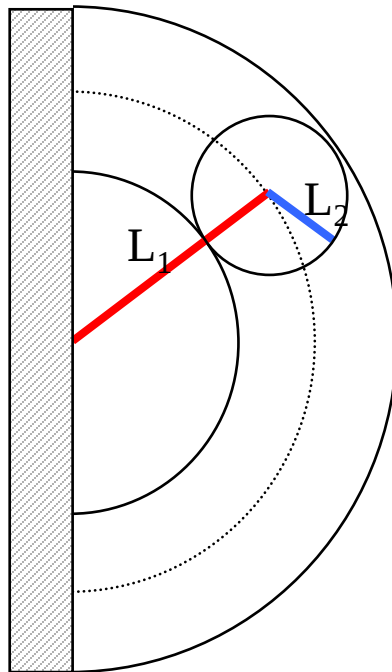
Inverzna kinematika

- željena pozicija i orijentacija vrha manipulatora (npr. glodalica) je zadana, a treba odrediti kutove u zglobovima da bi se to ostvarilo
- rješenja
 - niti jedno <http://kiroxx.de/html/RobotsInLove.html>
 - jedno
 - puno rješenja
- postavljaju se ograničenja u zglobovima (predimenzioniran sustav, poddimenzioniran sustav)
- zadan je početni položaj vrha manipulatora te ciljni položaj koji je potrebno dohvatiti, a zatim je potrebno odrediti promjene kutova u zglobovima da se to ostvari
- na vrhu kinematičkog lanca pratimo promjenu linearne $\mathbf{v}$ i kutne brzine $\boldsymbol{\omega}$

Radni prostor

- dohvatljiv radni prostor (engl. reachable workspace)
- spretno dohvatljiv radni prostor (engl. dexterous workspace)
 - prostor dohvatljiv uz razne orijentacije

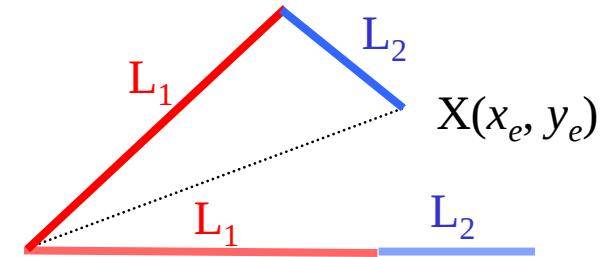
$$(L_1 - L_2) \leq \text{radni prostor} \leq (L_1 + L_2)$$



NPR: Određivanje kutova Θ_1 i Θ_2 iz zadanog položaja vrha manipulatora $\mathbf{X}(x_e, y_e)$

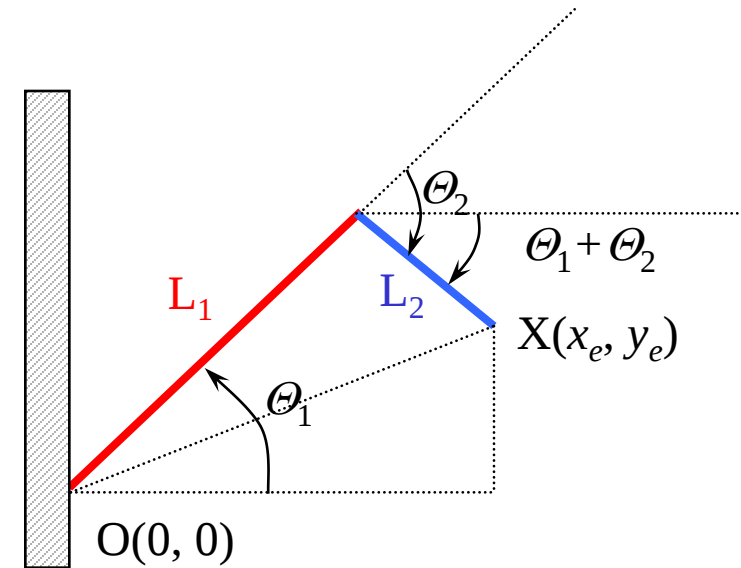
- direktna kinematika

$$\mathbf{X} = \mathbf{F}(\Theta_1, \Theta_2) \quad \begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} L_1 \cos \Theta_1 + L_2 \cos(\Theta_1 + \Theta_2) \\ L_1 \sin \Theta_1 + L_2 \sin(\Theta_1 + \Theta_2) \end{bmatrix}$$



- inverzna kinematika

$$\Phi = \begin{bmatrix} \Theta_1 \\ \Theta_2 \end{bmatrix} = \mathbf{G}(x_e, y_e) = \begin{bmatrix} \mathbf{G}_x(x_e, y_e) \\ \mathbf{G}_y(x_e, y_e) \end{bmatrix}$$



- u općem slučaju složenog sustava imat ćemo puno rješenja
- u zglobovima je potrebno još postaviti ograničenja, odnosno minimalni i maksimalni kut čime je zadan raspon pojedinog zgloba
- u realnom slučaju možemo imati i ograničenja vezana uz okolni prostor, npr. da kroz neku cijev trebamo dohvatiti predmet
- u animaciji treba kontinuirano pratiti cilj – iterativno numeričko rješenje
- matematički alat
 - upotreba Jakobijeve matrice, rotacija sadrži funkcije $\sin$ i $\cos$, a to znači da je problem nelinearan. Jakobijevom maticom lineariziramo problem promatranjem prve derivacije (prvi član Taylor-ovog reda).
 - geometrijska Jakobijeva matrica
 - transponirana Jakobijeva matrica
 - CCD algoritam <http://jyunming-chen.github.io/tutsplus/tutsplus18-3.html>

Jednostavni primjer kinematičkog lanca:

- Θ_i ... ulazne varijable, kutovi u zglobovima (translacija u zglobovima)
- x_e, y_e ... pozicija vrha manipulaora (orijentacija u vrhu manipulatora)

$$\mathbf{X} = \begin{bmatrix} x_e \\ y_e \end{bmatrix}, \Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \end{bmatrix} \mathbf{F} = \begin{bmatrix} f_1(\Theta_1, \Theta_2) \\ f_2(\Theta_1, \Theta_2) \end{bmatrix} \quad \begin{array}{l} x_e = f_1(\Theta_1, \Theta_2) \\ y_e = f_2(\Theta_1, \Theta_2) \end{array}$$

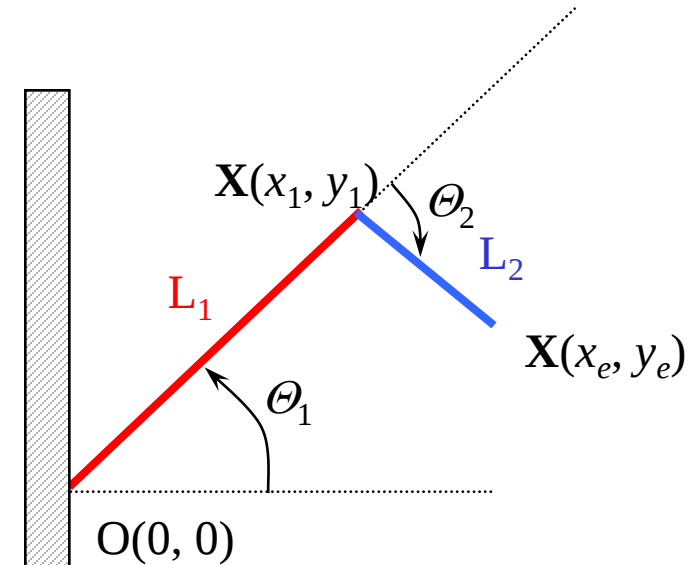
Direktna kinematika:

$$\mathbf{X} = \mathbf{F}(\Theta)$$

Inverzna kinematika

$$\Theta = \mathbf{F}^{-1}(\mathbf{X})$$

- funkcije f_i su nelinearne funkcije ($\sin, \cos$)
 - analitički definiran $\mathbf{F}^{-1}$
 - numeričko rješavanje – Jakobijeva matrica



- u općem slučaju mogu biti zadane i neke među-točke lanca npr: $\mathbf{X}(x_1, y_1)$

Matematičke osnove – rješavanje nelinearnih jednadžbi

- $f(x)$ nelinearna funkcija

$$\frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta f}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \quad f(x + \Delta x) \approx f(x) + \Delta x \frac{df}{dx} \quad (1)$$

Newton–Raphson – tražimo x_n gdje je $f(x)=0$

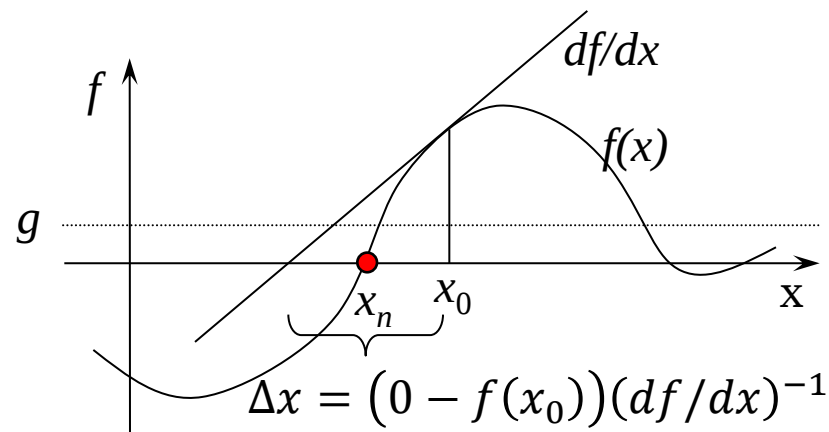
- za neki početni x_0 tražimo Δx :

$$x_{i+1} = x_i + \Delta x \text{ tako da } f(x_n)=0$$

$$(1) \Rightarrow \Delta x = -f(x_i) \left(\frac{df}{dx} \right)^{-1}$$

$$\Delta x = \beta (g - f(x_i)) \left(\frac{df}{dx} \right)^{-1}$$

$0 < \beta < 1$, kako odrediti β ?



Gradijentni spust – određivanje minimuma funkcije za funkcije više varijabli

Jakobijeva matrica

$$x_1 = f_1(\Theta_1, \Theta_2, \dots, \Theta_m)$$

$$x_2 = f_2(\Theta_1, \Theta_2, \dots, \Theta_m)$$

...

$$x_n = f_n(\Theta_1, \Theta_2, \dots, \Theta_m)$$

$$\mathbf{X} = \mathbf{F}(\Theta), \quad \frac{\partial \mathbf{F}}{\partial \Theta} = \frac{\partial (f_1, f_2, \dots, f_n)}{\partial (\Theta_1, \Theta_2, \dots, \Theta_m)}$$

$$\partial x_1 = \frac{\partial f_1}{\partial \Theta_1} \partial \Theta_1 + \frac{\partial f_1}{\partial \Theta_2} \partial \Theta_2 + \dots + \frac{\partial f_1}{\partial \Theta_m} \partial \Theta_m$$

$$\frac{\Delta \mathbf{X}}{\Delta \boldsymbol{\theta}} \approx \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}}$$

...

$$\partial x_i = \frac{\partial f_i}{\partial \Theta_1} \partial \Theta_1 + \frac{\partial f_i}{\partial \Theta_2} \partial \Theta_2 + \dots + \frac{\partial f_i}{\partial \Theta_m} \partial \Theta_m$$

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}}$$

$$\Delta \mathbf{X} \approx \mathbf{J} \Delta \boldsymbol{\theta}$$

...

- opis promjene izlaznih varijabli obzirom na promjenu ulaznih varijabli
- malo promijenimo ulaznu varijablu i promatramo što će se dogoditi s izlaznom varijablom

- Θ_i ... kutovi u zglobovima (pomak ako su prizmatični)
- ω_i ... kutne brzine u zglobovima
- $\mathbf{V}$... vektor linearnih i rotacionih brzina na vrhu manipulatora
- $\mathbf{J}(\theta)$... Jakobijeva matrica – Jakobijan

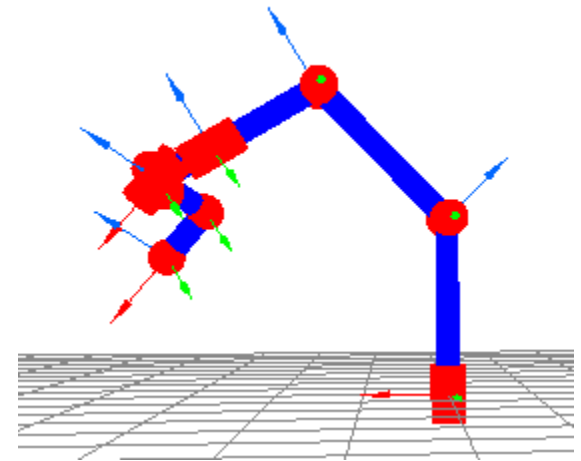
<https://markaren.github.io/WebGL-demos/FK.html>

$$\mathbf{V} = [v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^T$$

$$\mathbf{J}(\theta) = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \dots & \frac{\partial f_1}{\partial \theta_n} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \dots & \frac{\partial f_2}{\partial \theta_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_m}{\partial \theta_1} & \frac{\partial f_m}{\partial \theta_2} & \dots & \frac{\partial f_m}{\partial \theta_n} \end{bmatrix}$$

$$\Delta \mathbf{X} = \mathbf{J}(\theta) \Delta \theta$$

$$\mathbf{V} = \mathbf{J}(\theta) \dot{\theta}$$



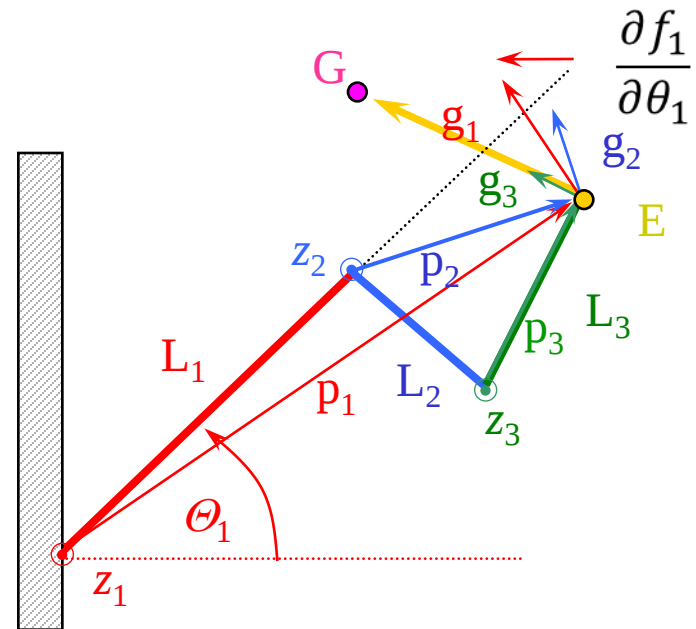
Geometrijski Jakobijan

- [illegible]

$$\Delta \mathbf{X} \approx \mathbf{J} \Delta \boldsymbol{\theta}$$

vrijedi samo lokano pa se u malim koracima treba približavati rješenju (iterativno)

$$f(x + \Delta x) \approx f(x) + \Delta x \frac{df(x)}{dx}$$



- pojedini članovi geometrijskog Jakobijana (engl. Geometric Jacobian) određuju doprinos pojedinih zglobova promatranoj promjeni na vrhu manipulatora npr. v_y .

$$z_1 = (0 \ 0 \ 1) \quad z_2 = (0 \ 0 \ 1) \quad z_3 = (0 \ 0 \ 1)$$

$$\begin{bmatrix} (G-E)_x \\ (G-E)_y \\ (G-E)_z \end{bmatrix} = \begin{bmatrix} (z_1 \times p_1)_x & (z_2 \times p_2)_x & (z_3 \times p_3)_x \\ (z_1 \times p_1)_y & (z_2 \times p_2)_y & (z_3 \times p_3)_y \\ (z_1 \times p_1)_z & (z_2 \times p_2)_z & (z_3 \times p_3)_z \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad \mathbf{V} = \mathbf{J}(\theta)\dot{\theta}$$

Algoritam:

G ciljna točka, **E** točka u kojoj se nalazimo, odredimo (**G-E**)

- Jakobijevu matricu potrebno je invertirati kako bi došli do rješenja, no ona nije nužno kvadratna. Tada se određuje pseudo-inverzna matrica tako da se jednačica pred-multiplicira s transponiranom Jakobijevom matricom

- odredimo pomak u kutu $\Delta\theta_n$ i $\theta_{n+1} = \theta_n + \beta \Delta\theta_n$

$$\mathbf{V} = \mathbf{J}\dot{\theta}$$

- robot:

$$\mathbf{J}^T \mathbf{V} = \mathbf{J}^T \mathbf{J} \dot{\theta}$$

- <https://markaren.github.io/WebGL-demos/Crane.html>
- PINV – pseudo inverse, DLS – prigušeni minimalni kvadrati
- <http://lo-th.github.io/fullik>

$$(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{V} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{J} \dot{\theta}$$

- <http://robot.glumb.de/>

$$(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{V} = \dot{\theta}$$

Pseudo inverzna Jakobijeva matrica $\mathbf{J}^+$

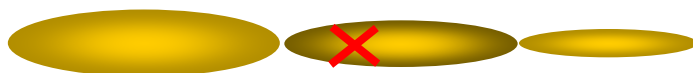
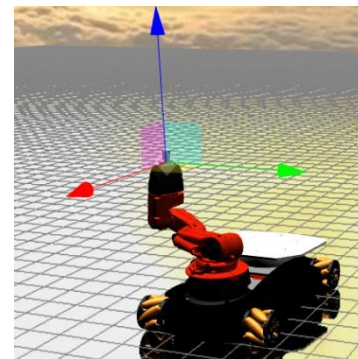
- U općem slučaju Jakobijeva matrica je veličine $[6 \times n]$, gdje je n broj segmenata kinematičkog lanca
- pseudo-inverzna matrica je:

$$(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{V} = \dot{\boldsymbol{\theta}}$$

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

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^+ \mathbf{V}$$

- jedinstvena,
- nestabilna u blizini singulariteta,
- vremenski zahtjevna za izračunavanje
- [Npr](#)



primjer singulariteta

CILJ

- generalno gledano imamo optimizacijski problem zadovoljavanja postavljenih uvjeta i možemo primijeniti razne optimizacijske algoritme – odabir β (selektivni) - umjesto invertiranja možemo minimizirati

Transponirana Jakobijeva matrica $\mathbf{J}^T$

- umjesto pseudo-inverzne matrice može se koristiti transponirana matrica
 - korištenje transponirane matrice nije matematički egzaktno, teži ka rješenju, no sporije konvergira
 - računanje nije vremenski zahtjevno
 - nema problema sa singularitetima (kod invertiranja matrice)
 - nedostatak prema $\mathbf{J}^+$ je što promjene u zglobovima nisu ujednačene, obrtni moment je jači dalje od kraja, pa su promjene disproporcionalne
 - javljaju se problemi pri skaliranju

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^T \mathbf{V}$$

- zapravo računamo projekcije vektora pomaka ($\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3$) na vektor ($\mathbf{G}-\mathbf{E}$)
- nije potrebno pohranjivati matricu $\mathbf{J}^T$ već kako računamo odmah primijenimo

CCD (engl. Cyclic Coordinate Descent)

- jednostavnija ideja, rješavaju se 1DOF problemi duž lanca
- iterativan postupak
- za svaki zglob se traži promjena koja će minimizirati udaljenost do cilja
 - 1DOF problemi su jednostavni i imaju rješenje
- nedostatak je neujednačenost promjena u zglobovima
- gibanje često nije glatko, može se zaglaviti, (slično lancu čiji vrh vučemo po podu

[Bajo](#)

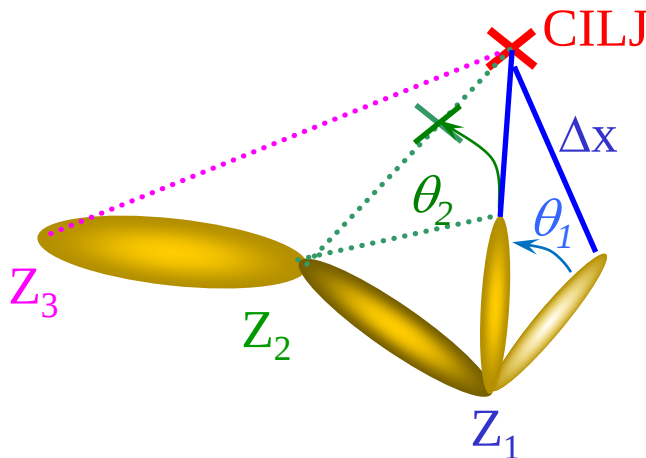
[CCD](#)

[Primjer](#)

[Npr](#)

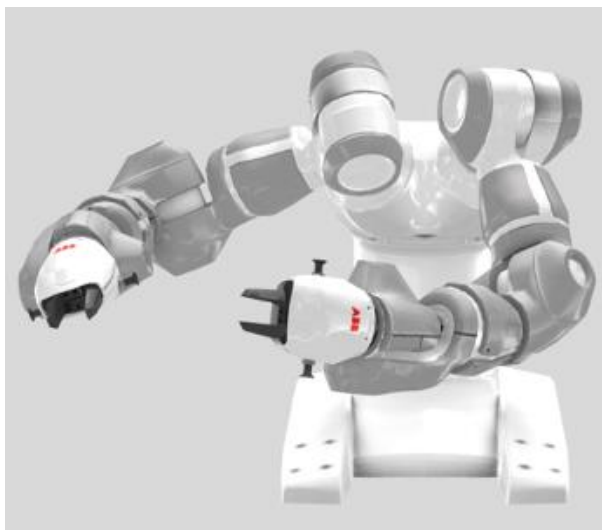
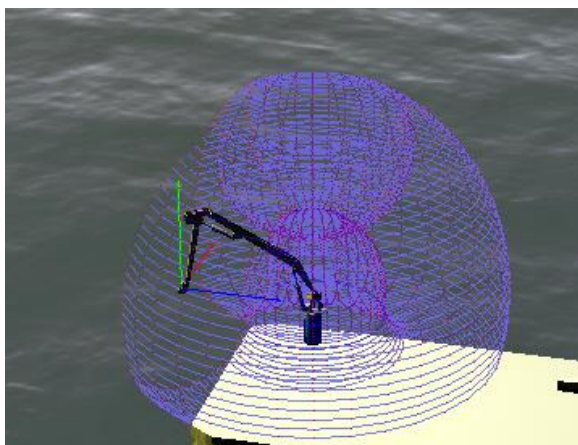
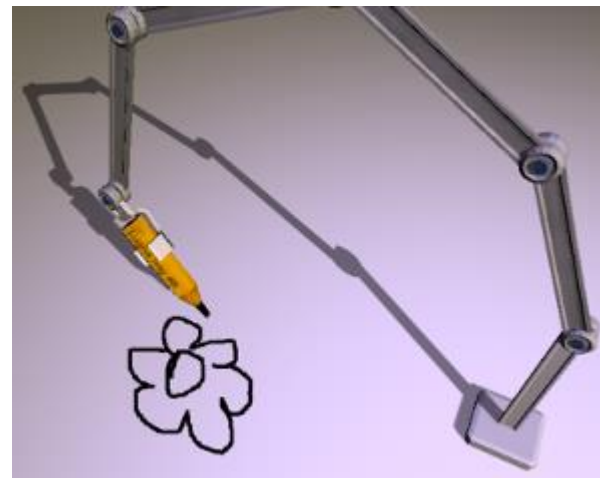
primjer pomaka koji minimiziraju udaljenost do cilja (iterativno ponavljamo):

1. korak – zadnji segment rotiramo oko zgloba Z_1 za θ_1 tako da se poklopi sa spojnicom Z_1 -CILJ (vrh se približio CILJU)
2. korak – rotaciju zadnja dva segmenta radimo oko zgloba Z_2 za kut θ_2 tako da vrh lanca dođe na spojnici Z_2 -CILJ
3. korak - rotiramo cijeli lanac oko zgloba Z_3



Primjeri primjene

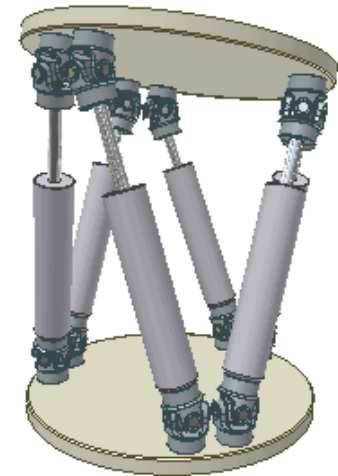
- simulacija rada robota
glatki kontinuirani pokreti
 - jednostavan primjer
 - http://kbladin.se/dmt_projects/inverse_kinematics.php
 - složen primjer
 - <https://markaren.github.io/WebGL-demos/Offshore.html>
 - <http://www.youtube.com/watch?v=b2bExqhhWRI>
 - <https://www.bostondynamics.com/spot>
- /



Primjeri primjene

- proračuni i prikaz kod mehaničkih konstrukcija
 - određivanje radnog prostora
 - simulacija rada
- heksapod (Stewart-ova platforma)
 - paralelni manipulatori <http://motionplatform.eu/10/>
 - kokpit kod simulatora leta
 - CNC uređaji za obradu ploha
 - roboti

From Computer Desktop Encyclopedia
Reproduced with permission.
© 1996 Evans & Sutherland Computer Corporation



Primjeri primjene

- animiranje likova (skeletal animation)
 - model kostura služi kao osnova za dodavanje kože, vrhovi koji definiraju kožu vežu se uz pojedine elemente kosti s pripadnim težinama i sve transformacije za kost primjenjuju se i na kožu (problem nastaje u području preklapanja)
 - stapanje animacija stajanje-hod-trčanja (blending)

<http://lo-th.github.io/Avatar.lab/> ANIMATION

x <https://jsantell.github.io/THREE.IK/examples/#single-effector>

x <http://www.realitymeltdown.com/WebGL3/skeletal-anim-inspector.html> start Weight

http://lo-th.github.io/root/sea_dev/index_babe_phy.html (vezanje s kožom) Bones bip

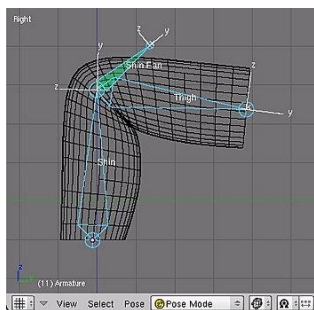
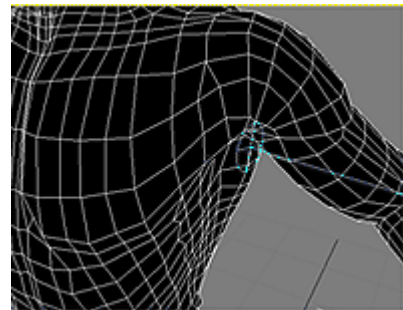
- kombinacija animacije kadrovima i proceduralne

• x http://www.visualiser.fr/demo.php?id=Away3D_Ragdoll

• [Medo](#)

- model koji sadrži i mišićnu strukturu

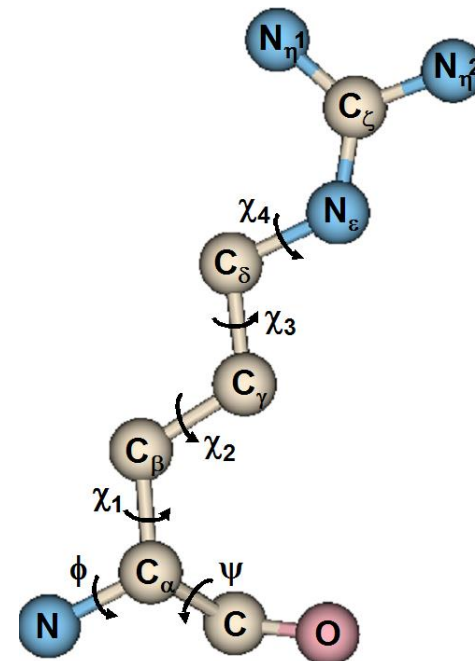
• <https://codemyui.com/multi-layer-image-slider/>
(tekstura) <http://lo-th.github.io/root/blending2/index.html>
[cgSkinLeg.avi](#)



Primjeri primjene

– direktna kinematika proteina

- aminokiseline → proteini (bjelančevine)
- dihedralni kut – kut između dvije ravnine koje određuju 4 atoma
- različiti 3D oblici (konformacije) istog proteina uz varijaciju dihedralnih kutova
određeni rotacijski DOF na peptidnim vezama,
(proračun minimalne energije),



https://threejs.org/examples/#webgl_loader_pdb

.pdb zapis dio iz Hemoglobina

ATOM	4384	CD2 HIS D 146	-4.470	20.439	23.452	1.00	22.44	C
ATOM	4385	CE1 HIS D 146	-3.622	22.457	23.406	1.00	28.57	C
ATOM	4386	NE2 HIS D 146	-3.890	21.436	24.195	1.00	34.26	N
ATOM	4387	OXT HIS D 146	-2.678	20.431	18.984	1.00	24.06	O
TER	4388	HIS D 146						
HETATM	4389	FE HEM A 1	15.703	15.811	15.254	1.00	18.90	FE
HETATM	4390	CHA HEM A 1	17.214	18.452	16.603	1.00	13.89	C
HETATM	4391	CHB HEM A 1	17.793	15.899	12.545	1.00	17.48	C
HETATM	4392	CHC HEM A 1	13.730	13.474	13.636	1.00	15.77	C
HETATM	4393	CHD HEM A 1	12.980	16.329	17.470	1.00	14.95	C
HETATM	4394	NA HEM A 1	17.206	16.929	14.684	1.00	18.55	N