

KINEMATIKA nastavak

JAKOBIJAN

Svako kretanje u zglobovima robota ima za posledicu kretanje hvataljke. Međutim, kretanje hvataljke po određenoj trajektoriji zahtijeva koordinisano kretanje svih zglobova mehaničke strukture robota. Da bi mogli da realizujemo zahtijevano kretanje hvataljke robota potrebno je da se uspostavi veza između kretanja u zglobovima mehaničke konfiguracije i kretanja hvataljke, tj brzine u zglobovima $\dot{\mathbf{q}}$, i linearne i ugaone brzine hvataljke ($\dot{\mathbf{p}}$ i $\boldsymbol{\omega}$, respektivno) pa možemo pisati

$$\dot{\mathbf{p}} = J_L \dot{\mathbf{q}}$$

$$\boldsymbol{\omega} = J_U \dot{\mathbf{q}}$$

Ove dvije jednačine možemo objediniti u jedinstven izraz čime se dobija relacija

$$\dot{\mathbf{s}} = \mathbf{v} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}$$

gdje je $\dot{\mathbf{S}}$ ukupni vektor brzine hvataljke izražen u spoljašnjim koordinatama (objedinjeni su vektori linearne i ugaone brzine hvataljke), $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]^T$ je vektor brzina u zglobovima mehanizma. Matrica $\mathbf{J}(\mathbf{q})$ kojom se povezuje kretanje u zglobovima mehanizma sa kretanjem hvataljke se naziva Jakobijevom matricom ili Jakobijanom. Prema tome, ***jakobijeva matrica povezuje brzine kretanja zglobova robotskog mehanizma sa linearnom i ugaonom brzinom kretanja hvataljke izražene u spoljašnjim (dekartovim pravouglim) koordinatama.***

Da bi bolje razumjeli prirodu matrice \mathbf{J} napišimo izraz u razvijenoj formi

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & J_{66} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & \dots & J_{1n} \\ J_{21} & J_{22} & J_{23} & \dots & J_{2n} \\ J_{31} & J_{32} & J_{33} & \dots & J_{3n} \\ J_{41} & J_{42} & J_{43} & \dots & J_{4n} \\ J_{51} & J_{52} & J_{53} & \dots & J_{5n} \\ J_{61} & J_{62} & J_{63} & \dots & J_{6n} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} & \dots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} & \dots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} & \dots & \frac{\partial z}{\partial q_n} \\ \frac{\partial \phi}{\partial q_1} & \frac{\partial \phi}{\partial q_2} & \frac{\partial \phi}{\partial q_3} & \dots & \frac{\partial \phi}{\partial q_n} \\ \frac{\partial \theta}{\partial q_1} & \frac{\partial \theta}{\partial q_2} & \frac{\partial \theta}{\partial q_3} & \dots & \frac{\partial \theta}{\partial q_n} \\ \frac{\partial \psi}{\partial q_1} & \frac{\partial \psi}{\partial q_2} & \frac{\partial \psi}{\partial q_3} & \dots & \frac{\partial \psi}{\partial q_n} \end{bmatrix}$$

Numeričko sračunavanje Jakobijana

- Treba primijetiti da su elementi matrice J funkcije uglova u zglobovima i da se sa promjenom položaja mehaničke konfiguracije robota mijenjaju. Prema tome, tokom realizacije određenog pokreta matrica J nije konstantna i vrijednosti njenih elemenata treba sračunavati za svaki novi položaj mehaničke konfiguracije.

U ovom dijelu će biti prikazan način numeričkog sračunavanja elemenata Jakobijana kako bi se ovaj zadatak mogao efikasno realizovati primjenom računara. Kao što smo već uočili i linearnom i ugaonom kretanju hvataljke doprinosi kretanje u svakom od zglobova mehanizma. Ako napišemo Jakobijevu matricu u sledećem obliku

$$J = \begin{bmatrix} J_{L1} & J_{L2} & \cdots & J_{Ln} \\ J_{U1} & J_{U2} & \cdots & J_{Un} \end{bmatrix}$$

tako da svaki elemenat J_{Li} i J_{Ui} predstavlja vektor kolonu dimenzije (3x1), lako se može uočiti da prvi dio kolone (J_{Li}) odgovara doprinosu i-tog zgloba linearnom, a drugi deo (J_{Ui}) ugaonom kretanju hvataljke.

U razvijenoj formi prethodna relacija postaje

$$\begin{bmatrix} v_H \\ \omega_H \end{bmatrix} = \begin{bmatrix} J_{L1}^1 \\ J_{L1}^2 \\ J_{L1}^3 \\ J_{U1}^1 \\ J_{U1}^2 \\ J_{U1}^3 \end{bmatrix} \begin{bmatrix} J_{L2}^1 \\ J_{L2}^2 \\ J_{L2}^3 \\ J_{U2}^1 \\ J_{U2}^2 \\ J_{U2}^3 \end{bmatrix} \cdots \begin{bmatrix} J_{Ln}^1 \\ J_{Ln}^2 \\ J_{Ln}^3 \\ J_{Un}^1 \\ J_{Un}^2 \\ J_{Un}^3 \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

Odavde se linearna brzina vrha robota može izraziti na sledeći način

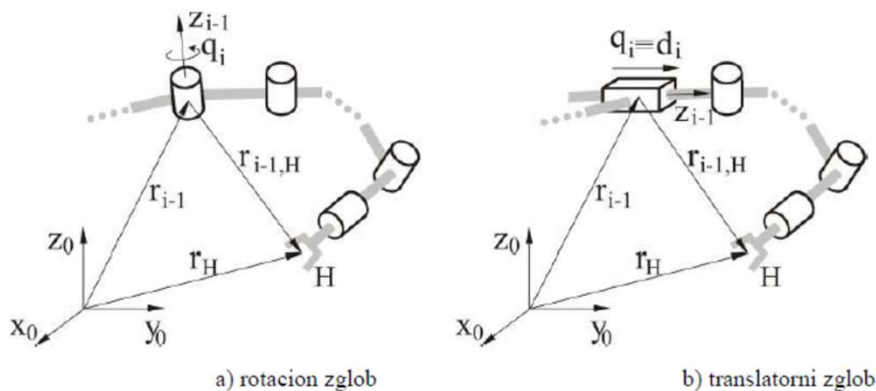
$$v_H = J_{L1} \cdot \dot{q}_1 + J_{L2} \cdot \dot{q}_2 + \dots + J_{Ln} \cdot \dot{q}_n$$

a na isti način i ugaona brzina hvataljke

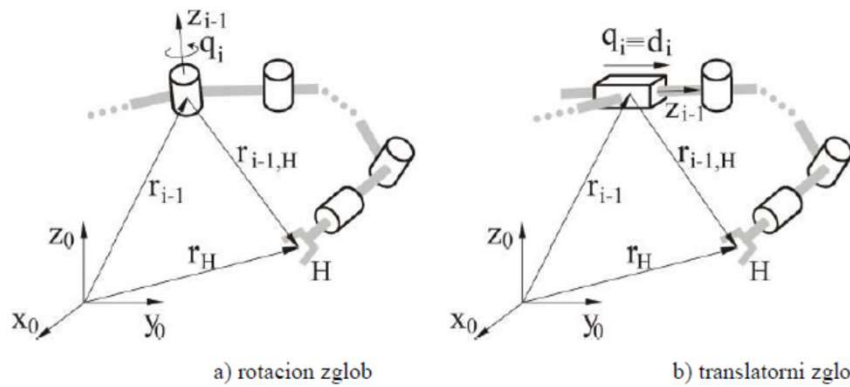
$$\omega_H = J_{U1} \cdot \dot{q}_1 + J_{U2} \cdot \dot{q}_2 + \dots + J_{Un} \cdot \dot{q}_n$$

Razmotrimo na koji način možemo da odredimo elemente matrice J_{Li} i J_{Ui} .

Na Sl. 4.3.a. je prikazan slučaj kada je i-ti zglob robota (rotacioni) i njegov položaj u odnosu na hvataljku, dok je na Sl. 4.3.b. i-ti zglob robota translatorni. Neka su sa z_{i-1} obilježene ose zglobova koje su postavljene u postupku određivanja DH parametara. Razmotrimo najprije kako kretanje nekog od zglobova mehanizma može da doprinese promjeni linearne brzine hvataljke.



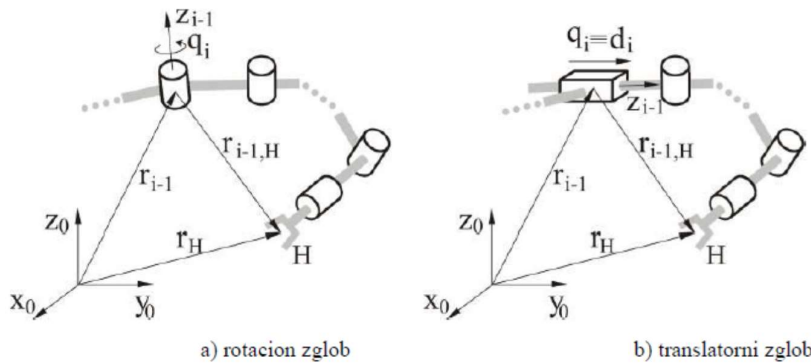
Sl. 4.3. Infinitesimalni pomeraji hvataljke zbog pomeranja rotacionog i linearnog (translatorsnog) zgloba



Sl. 4.3. Infinitesimalni pomjeraji hvataljke zbog pomaranja rotacionog i linearnog (translatornog) zgloba

U slučaju da je i -ti zglob translatoran njegovo pomjeranje izaziva samo translatorno pomjeranje hvataljke u pravcu z_{i-1} ose, intenziteta \dot{d}_i (pripadajuća z osa je tada z_{i-1} , a DH parametar d_i je promjenljiva), tj.

$$J_{Li} \cdot \dot{q}_i = z_{i-1} \cdot \dot{d}_i$$



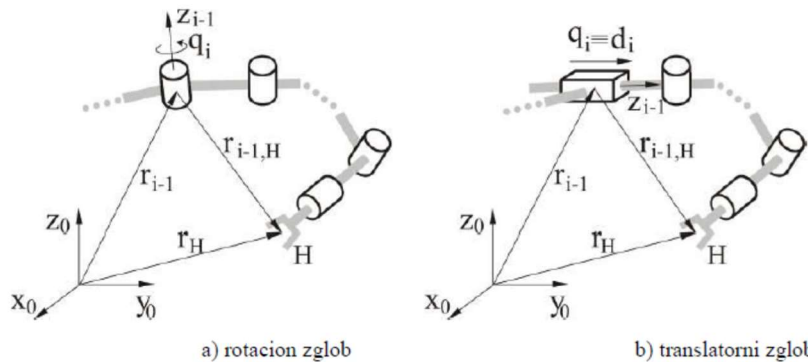
Sl. 4.3. Infinitesimalni pomjeraji hvataljke zbog pomaranja rotacionog i linearnog (translatornog) zgloba

Ako je i -ti zglob rotacioni njegovim pokretanjem će dio mehanizma koji se nalazi dalje od zgloba koji se pokreće (uključujući i hvataljku) dobiti ugaonu brzinu

$$\omega_i = z_{i-1} \cdot \dot{q}_i$$

koja će linearnu brzinu hvataljke robota, koja je od ose obrtanja udaljena za $r_{i-1,H}$, promijeniti za

$$J_{Li} \cdot \dot{q}_i = \omega_i \times r_{i-1,H} = (z_{i-1} \times r_{i-1,H}) \cdot \dot{q}_i$$



Sl. 4.3. Infinitesimalni pomeraži hvataljke zbog pomeranja rotacionog i linearnog (translatornog) zgloba

Na sličan način ćemo odrediti kako kretanja u translatorsnom i rotacionom zglobu doprinose promjeni ugaone brzine hvataljke. Ako je i -ti zglob translatorsni on ne doprinosi promjeni ugaone brzine hvataljke, pa je

$$J_{U_i} \cdot \dot{q}_i = 0$$

Ako je i -ti zglob rotacioni, ugaona brzina koja se generiše kretanjem zgloba, je data sa

$$J_{U_i} \cdot \dot{q}_i = z_{i-1} \cdot \dot{q}_i$$

Prema tome, prethodnim jednačinama je moguće odrediti elemente jakobijana J_{L_i} i J_{U_i} za slučaj translatorsnih i rotacionih zglobova pa, prema tome, možemo rezimirati

- Za translatorsni zglob
$$\begin{bmatrix} J_{L_i} \\ J_{U_i} \end{bmatrix} = \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix}$$

- Za rotacioni zglob
$$\begin{bmatrix} J_{L_i} \\ J_{U_i} \end{bmatrix} = \begin{bmatrix} z_{i-1} \times r_{i-1,H} \\ z_{i-1} \end{bmatrix}$$

Pri sračunavanju Jakobijana jedine veličine koje treba odrediti su vektori z_{i-1} i $r_{i-1,H}$ koji su funkcije samo pomjeraja u zglobovima robota. Vektor z_{i-1} predstavlja treću kolonu u matrici rotacije R^{i-1}_0 i moguće ga je veoma jednostavno odrediti na sledeći način

$$z_{i-1} = R_0^1(q_1)R_1^2(q_2)\dots R_{i-2}^{i-1}(q_{i-1})z_0$$

gde je $z_0 = [0 \ 0 \ 1]^T$ što nam omogućava da se iz matrice R^{i-1}_0 odaberemo samo treću kolonu.

Da se podsjetimo

$$R_0^1 = \begin{bmatrix} \bar{i}_1 \cdot \bar{i}_0 & \bar{j}_1 \cdot \bar{i}_0 & \bar{k}_1 \cdot \bar{i}_0 \\ \bar{i}_1 \cdot \bar{j}_0 & \bar{j}_1 \cdot \bar{j}_0 & \bar{k}_1 \cdot \bar{j}_0 \\ \bar{i}_1 \cdot \bar{k}_0 & \bar{j}_1 \cdot \bar{k}_0 & \bar{k}_1 \cdot \bar{k}_0 \end{bmatrix}$$

Matrica R_0^1 se naziva matricom rotacije. Iz gornje matrice se vidi da kolone matrice R_0^1 predstavljaju projekcije ortova koordinatnog sistema $O_1-x_1y_1z_1$ na sistem $O_0-x_0y_0z_0$

Vektor položaja od i-1-vog zgloba do hvataljke $r_{i-1,H}$ je moguće odrediti (Sl. 4.3.a i b) kao razliku vektora p_H od koordinatnog početka do hvataljke i vektora p_{i-1} od koordinatnog početka do i-1-vog zgloba. Vektor p_H je moguće odrediti kao prva tri elementa četvrte kolone ukupne homogene matrice transformacije H_0^H koja je data sa

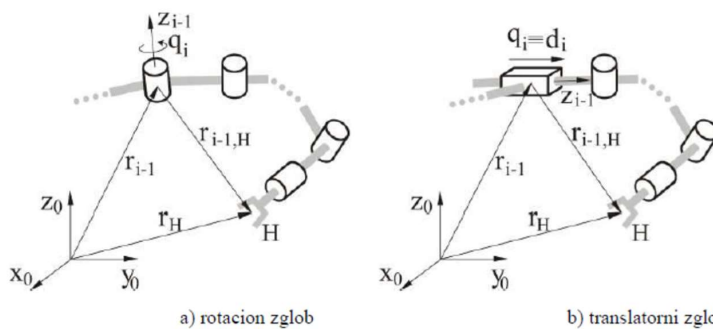
$$H_0^H = H_0^1(q_1)H_1^2(q_2) \cdots H_{n-1}^n(q_n)$$

,pa slijedi

$$p_H = H_0^H p_0$$

gdje je p_0 dato kao

$$p_0 = [0 \ 0 \ 0 \ 1]^T$$



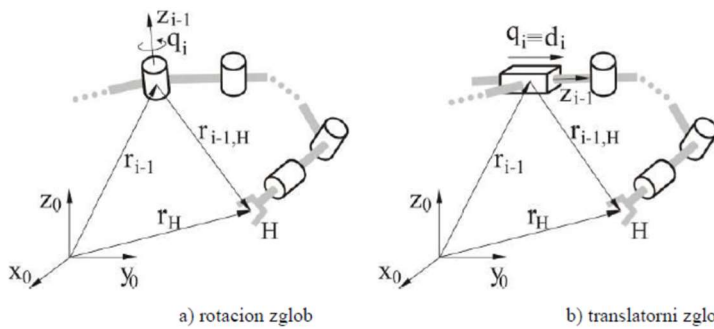
Sl. 4.3. Infinitesimalni pomjeraji hvataljke zbog pomjeranja rotacionog i linearnog (translacionog) zgloba

Sličnim postupkom je moguće odrediti i vektor p_{i-1}

$$P_{i-1} = H_0^1(q_1)H_1^2(q_2)\dots H_{i-2}^{i-1}(q_{i-1})P_0$$

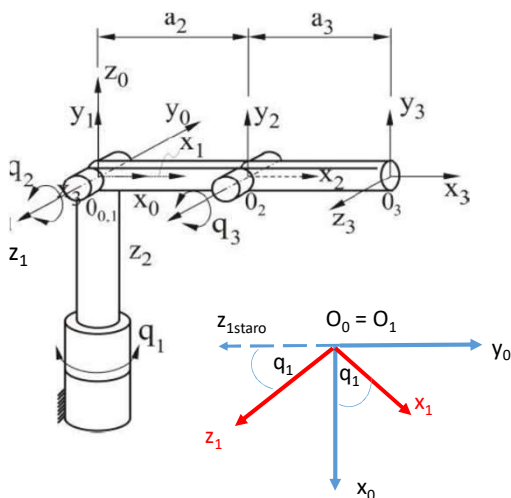
i konačno dobijamo:

$$r_{i-1,H} = P_H - P_{i-1}$$



Sl. 4.3. Infinitesimalni pomeraji hvataljke zbog pomeranja rotacionog i linearnog (translatorsnog) zgloba

Prema tome, ovi izrazi omogućavaju jednostavno sračunavanje elemenata Jakobijana numeričkim putem pod uslovom da su poznate matrice transformacije koje odgovaraju datoj (trenutnoj) pozi mehaničke strukture robota



PRIMJER: Numeričkim načinom odrediti elemete Jakobijana za trosegmentnu antropomorfnu konfiguraciju na slici prema postavljenim koordinatnim sistemima

Odredimo najprije projekcije osa zglobova

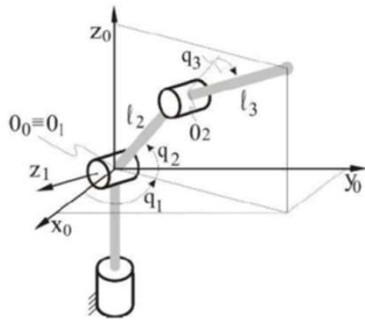
$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad z_1 = \begin{bmatrix} S1 \\ -C1 \\ 0 \end{bmatrix}, \quad z_2 = \begin{bmatrix} S1 \\ -C1 \\ 0 \end{bmatrix}$$

Obzirom da su sva tri zgloba rotaciona Jakobijan ima oblik

$$J_{11} = \begin{bmatrix} z_0 \times (O_3 - O_0) & z_1 \times (O_3 - O_1) & z_2 \times (O_3 - O_2) \end{bmatrix}$$

na osnovu čega slijedi

$$O_0 = O_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T, \quad O_2 = \begin{bmatrix} a_2 C1C2 \\ a_2 S1C2 \\ a_2 S2 \end{bmatrix}, \quad O_3 = \begin{bmatrix} a_2 C1C2 + a_3 C1C23 \\ a_2 S1C2 + a_3 S1C23 \\ a_2 S2 + a_3 S23 \end{bmatrix}$$



Sada možemo da odredimo elemete Jakobijana

Sl. 4.2. Kinematska šema trosegmentnog antropomorfnog mehanizma

$$z_0 \times (O_3 - O_0) = \begin{bmatrix} -a_2 S1C2 - a_3 S1C23 \\ a_2 S1C2 + a_3 C1C23 \\ 0 \end{bmatrix}$$

$$z_1 \times (O_3 - O_1) = \begin{bmatrix} -C1(a_2 S2 + a_3 S1C23) \\ -S1(a_2 S2 + a_3 S23) \\ a_2 C2 + a_3 C23 \end{bmatrix}$$

$$z_2 \times (O_3 - O_2) = \begin{bmatrix} a_3 C1S23 \\ a_3 S1S23 \\ a_3 C23 \end{bmatrix}$$

odakle konačno slijedi

$$J = \begin{bmatrix} -a_2 S1C2 - a_3 S1C23 & -C1(a_2 S2 + a_3 C23) & -a_3 C1S23 \\ a_2 S1C2 + a_3 C1C23 & -S1(a_2 S2 + a_3 S23) & a_3 S1S23 \\ 0 & a_2 C2 + a_3 C23 & a_3 C23 \end{bmatrix}$$

što je isti rezultat dobijen analitičkim načinom. Osim toga, primijetimo da je Jakobijan dobijen u formi matrice dimenzija (3x3) obzirom da se radi samo o minimalnoj konfiguraciji sa tri stepena slobode.

NUMERIČKO RJEŠAVANJE INVERZNOG KINEMATSKOG PROBLEMA

Jakobijeva matrica koju smo odredili u prethodnom dijelu predstavlja osnovu za numeričko rješavanje inverznog kinematskog problema. Postupak rješavanja je potpuno opšti i ne zavisi od tipa robota, ali za razliku od analitičkog rješenja, ovdje se dobija samo jedno rješenje koje odgovara zadatom vektoru spoljašnjih koordinata \mathbf{s} i to ono koje je najbliže zadatom rješenju \mathbf{q}^{POC} od kojeg je procedura započeta. Naime, iz izraza

$$\dot{\mathbf{s}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

inverzijom jakobijeve matrice dobijamo

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^{-1} \dot{\mathbf{s}}$$

Ukoliko je poznata početna poza robota \mathbf{q}^{POC} koja je bliska očekivanom rješenju (tj. ako su poznati uglovi u zglobovima robota koji dovode hvataljku u okolinu očekivanog položaja i orijentacije) tada se pozicije u zglobovima mogu pronaći integracijom brzina u zglobovima nad posmatranim vremenskim intervalom

$$\mathbf{q}(t) = \mathbf{q}^{POC}(t) + \int_t^{t+\Delta t} \dot{\mathbf{q}}(\tau) d\tau$$

Ako integraciju realizujemo u diskretnom vremenu, najjednostavnijom Ojlerovom metodom integracije, gornji izraz postaje

$$\mathbf{q}(t + \Delta t) = \mathbf{q}^{POC}(t) + \dot{\mathbf{q}}(t) \Delta t$$

Ukoliko izrazimo generalisane brzine preko spoljašnjih izraz postaje

$$\mathbf{q}(t + \Delta t) = \mathbf{q}^{POC}(t) + \mathbf{J}^{-1}(\mathbf{q}(t)) \cdot \dot{\mathbf{s}}(t) \cdot \Delta t$$

ili sa modifikovanom notacijom

$$\mathbf{q}_{(k+1)} = \mathbf{q}_k + \mathbf{J}^{-1}(\mathbf{q}_k) \Delta \mathbf{s}_k$$

Neka je zadat vektor spoljašnjih koordinata \mathbf{s}^0 i neka je potrebno da se odredi vektor unutrašnjih koordinata koje odgovaraju položaju hvataljke \mathbf{s}^0

Pretpostavimo najprije približno početno rješenje $\mathbf{q}^{(k)}$, ($\mathbf{q}^{(k)} \approx \mathbf{f}^{-1}(\mathbf{s}^0)$). Kada se na osnovu pretpostavljenog rješenja $\mathbf{q}^{(k)}$ sračuna pozicija hvataljke pojavice se odstupanje između tačne i približne pozicije

$$\Delta \mathbf{s}^{(k)} = \mathbf{s}^0 - \mathbf{f}(\mathbf{q}^{(k)}) \approx \mathbf{J}(\mathbf{q}^{(k)}) \cdot [\mathbf{q}^{(k+1)} - \mathbf{q}^{(k)}]$$

gde je $\mathbf{q}^{(k+1)}$ tačnije rješenje inverznog kinematskog problema od početno pretpostavljenog $\mathbf{q}^{(k)}$ za \mathbf{s}^0 odakle slijedi

$$\mathbf{q}^{(k+1)} \approx \mathbf{q}^{(k)} + \mathbf{J}^{-1}(\mathbf{q}^{(k)}) \cdot \Delta \mathbf{s}^{(k)}$$

što predstavlja osnovnu relaciju za iterativni postupak rješavanja inverznog kinematskog problema

Postupak rješavanja se može definisati sledećim koracima:

- korak 1:** za zadato \mathbf{s}^0 pretpostavimo rješenje inverznog problema $\mathbf{q}^{(k)}$;
- korak 2:** odrede se vrijednosti spoljašnjih koordinata za pretpostavljeno $\mathbf{q}^{(k)}$;
- korak 3:** izračunamo grešku $\Delta \mathbf{s}^{(k)}$;
- korak 4:** izračunamo Jakobijan \mathbf{J} u pretpostavljenom rješenju $\mathbf{q}^{(k)}$;
- korak 5:** izračunamo inverzni Jakobijan \mathbf{J}^{-1} ;
- korak 6:** na osnovu izraza

$$\mathbf{q}^{(k+1)} \approx \mathbf{q}^{(k)} + \mathbf{J}^{-1}(\mathbf{q}^{(k)}) \cdot \Delta \mathbf{s}^{(k)}$$

odredimo naredno rješenje koje je bliže tačnom rješenju nego što je bilo prethodno;

- korak 7:** opisanu proceduru ponavljamo sve dok greška ne bude manja od unaprijed zadate vrednosti ε kojom se definiše željena tačnost rješenja, tj. dok ne bude ispunjen uslov,

$$|\Delta \mathbf{s}^{(k)}| < \varepsilon$$

Prikazani postupak je klasična Njutnova metoda za rješavanje sistema nelinearnih jednačina kojim se dobija samo jedno rješenje i to ono koje je najbliže početnom pretpostavljenom rješenju $\mathbf{q}^{(0)}$.

PSEUDO INVERZNI JAKOBIJAN

Prethodno opisani postupak numeričkog rješavanja inverznog kinematskog problema je baziran na relaciji

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^{-1} \dot{\mathbf{s}}$$

tj. na pretpostavci da je moguće odrediti inverzni jakobijan $\mathbf{J}^{-1}(\mathbf{q})$, tj. da je matrica \mathbf{J} kvadratna i da nema singularitet. Ako sa m označimo dimenziju vektora spoljašnjih koordinata \mathbf{s} , a sa n broj stepeni slobode mehaničke konfiguracije robota možemo razlikovati tri slučaja:

a) $m = n$

U ovom slučaju je Jakobijeva matrica kvadratna te se može odrediti njena inverzna matrica ako je njena determinanta različita od nule. Položaji robota u kojima je $\det(\mathbf{J}(\mathbf{q})) = 0$ predstavljaju singularne tačke u kojima se ne može odrediti inverzni Jakobijan.

b) $m > n$

U ovom slučaju broj stepeni slobode zadatka je veći od broja stepeni slobode robota pa se može dogoditi da ne postoji \dot{q} koje je rješenje sistema

$$\dot{q} = J(q)^{-1} \dot{s}$$

Drugim riječima broj jednačina je manji od broja nepoznatih u sistemu. Stoga se pribjegava traženju rješenja koje zadovoljava sistem uz najmanju grešku. Obilježimo sa \dot{q}^* vektor koordinata zglobova koji zadovoljava uslov da je greška definisana izrazom

$$G = \text{Greska}^2 = (J\dot{q}^* - \dot{s})^T (J\dot{q}^* - \dot{s})$$

minimalna. Razvojem gornjeg izraza dobijamo

$$G = (J\dot{q}^*)^T (J\dot{q}^*) - 2(\dot{s})^T J\dot{q}^* + \dot{s}^T \dot{s}$$

Da bi minimizirali grešku diferencirajmo gornji izraz i izjednačimo ga sa nulom tj.

$$\frac{\partial G}{\partial \dot{q}} = \frac{\partial}{\partial \dot{q}} (\dot{q}^{*T} (J^T J) \dot{q}^*) - \frac{\partial}{\partial \dot{q}} (2\dot{s}^T J\dot{q}^*)$$

Sređivanjem izraza dobijamo $(J^T J) \dot{q}^* = J^T \dot{s}$ ili $\dot{q}^* = (J^T J)^{-1} J^T \dot{s}$

pod uslovom da $(J^T J)^{-1}$ postoji. Matrica $(J^T J)^{-1} J^T$ se naziva lijevim pseudo inverznim Jakobijanom. Ako ovu matricu obilježimo sa J^{PS} možemo pisati

$$\dot{q} = J^{PS} \dot{s}$$

Čime su određene koordinate zglobova tako da je greška pozicioniranja minimalna.

c) $m < n$

U ovom slučaju je broj stepeni slobode koji zahtijeva zadatak manji od broja stepeni slobode koji posjeduje robotski manipulator (manipulator je redundantan), tj broj jednačina je veći od broja nepoznatih u sistemu pa postoji beskonačno mnogo rješenja od kojih treba odabrati jedno. Jedan od načina da se vrši izbor rješenja je da Euklidova norma bude minimalna. Ovo rješenje možemo dobiti metodom Lagranževih množitelja. Ako je $\lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m]$ vektor Lagranževih množitelja možemo postaviti novu funkciju cilja

$$\Psi(\dot{q}) = (\dot{q})^T (\dot{q}) + \lambda^T (J\dot{q} - \dot{s})$$

koja je u minimumu kada je $\partial\Psi/\partial q = 0$, pa se može pisati

$$2\dot{q} + J^T \lambda = 0 \quad \text{odakle slijedi} \quad \dot{q} = -J^T \lambda / 2$$

Zamjenom ovog izraza u polazni $\dot{s} = J(q)\dot{q}$ slijedi $\dot{s} = J(-J^T \lambda / 2)$

pa se posle sređivanja dobija: $\lambda = -2(JJ^T)^{-1} \dot{s}$

pod uslovom da $(JJ^T)^{-1}$ postoji. Zamenjujući ovaj izraz u $\dot{q} = -J^T \lambda / 2$

i nakon jednostavnih transformacija dobijamo

$$\dot{q} = J^T (JJ^T)^{-1} \dot{s}$$

Matrica $J^T (JJ^T)^{-1}$ se naziva desnim pseudoinverznim Jakobijanom koji ćemo kao i u prethodnom slučaju obilježiti sa J^{PS} , pa se konačno može pisati

$$\dot{q} = J^{PS} \dot{s}$$

odakle možemo odrediti tražene koordinate zglobova.

Na opisani način se može prevazići problem kada Jakobijan nije kvadratni i kada nije moguće naći inverznu matricu.

SINGULARNE KONFIGURACIJE

Kao što smo već vidjeli, linearna veza između brzine hvataljke izražene u spoljašnjim koordinatama i brzina u zglobovima

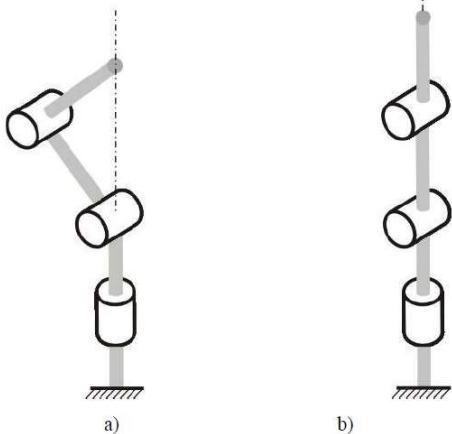
$$\dot{q} = J(q)^{-1} \dot{s}$$

je uspostavljena Jakobijanom. Ukoliko takvu vezu, pri određenim položajima (konfiguracijama) mehaničke strukture robota, nije moguće jednoznačno uspostaviti govorimo o pojavi singularnih konfiguracija. Matematičkim rečnikom rečeno, singularne su one konfiguracije mehaničke strukture pri kojima matrica $J(q)$ nema puni rang.

Singularni položaji se odlikuju sledećim osobinama:

- a) U singularnom položaju robot gubi jedan ili više SS.

Kao primjer ovakvog slučaja navedimo bilo koji robot čija hvataljka je dostigla granicu radnog prostora (spoljnu ili unutrašnju). Ovaj slučaj odgovara situaciji sa Sl. 4.4.b (hvataljka je na spoljašnjoj granici radnog prostora) ili kada je treći segment savijen potpuno naniže $q_3 = \pi$.

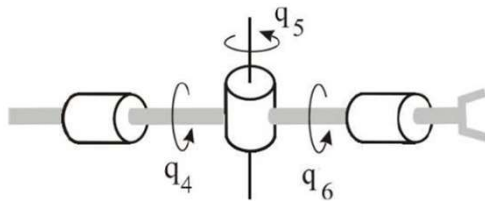


Sl. 4.4. Singularni položaji antropomorfne minimalne konfiguracije

- b) U singularnom položaju može postojati beskonačno mnogo rešenja inverznog kinematskog problema.

Da bi ilustrovali ovaj slučaj posmatrajmo minimalnu konfiguraciju (samo prva tri stepena slobode) antropomorfne konfiguracije u položaju koji je prikazan na Sl. 4.4. a. Obzirom da se vrh minimalne konfiguracije nalazi na osi obrtanja prvog zgloba njen položaj ne zavisi od trenutne vrednosti ugla q_1 i ovaj položaj je ostvaren za bilo koju vrednost ugla q_1 . Ovo važi za bilo koje vrednosti uglova q_2 i q_3 ako je ispunjeno da se vrh minimalne konfiguracije nalazi na vertikalnoj osi obrtanja, pa i za slučaj kada je $q_2 = q_3 = 0$ i kada je mehanička struktura robota potpuno ispužena naviše (Sl. 4.4.b)

Drugi primjer kojim ovakvu situaciju možemo ilustrovati je sferni (Ojlerov) zglob hvataljke robota u položaju koji je prikazan na Sl. 4.5. Uočimo da, ukoliko su ugaone brzine u četvrtom i šestom zglobu jednake po intenzitetu a suprotne po smjeru hvataljka će mirovati. Osim toga, ovdje pripada bilo koji položaj robota sa Sl. 4.4.a osim potpuno ispružene konfiguracije (Sl. 4.4.b).



Sl. 4.5. Singulani položaj sfernog zgloba

$$\dot{q}_4 = -\dot{q}_6$$

Kao što je već pomenuto uslov za pojavu singulariteta je kada Jakobijan gubi puni rang. Međutim, sračunavanje uslova singulariteta se može pojednostaviti ako se mehanička struktura robota raspregne na minimalnu konfiguraciju i zglob hvataljke. U tom slučaju Jakobijan robota sa šest stepeni slobode se u blok-formi može prikazati na sledeći način

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

gdje su, u slučaju sfernog zgloba,

matrice J_{12} i J_{22} date sa $J_{12} = [z_3 \times r_{3,H} \quad z_4 \times r_{4,H} \quad z_5 \times r_{5,H}]$ $J_{22} = [z_3 \quad z_4 \quad z_5]$

Obzirom da se položaj koordinatnog sistem hvataljke može slobodno birati, kao i da rezultat ne zavisi od izbora koordinatnog sistema u odnosu na koji je izražen smjestimo ga u centar zgloba (mjesto gde se sve tri ose sijeku) pošto je tada ispunjeno $r_{i-1,H} = 0$ $i=4,5,6$. U tom slučaju matrica J_{12} postaje

$$J_{12} = [0 \quad 0 \quad 0]$$

pa matrica J postaje blok donje-trougaona matrica, što znatno pojednostavljuje sračunavanje determinante

$$\det(J) = \det(J_{11}) \det(J_{22})$$

odakle se uslovi singularnosti za minimalnu konfiguraciju mogu napisati kao

$$\det(J_{11}) = 0$$

dok za zglob hvataljke važi sledeći uslov singularnosti

$$\det(J_{22}) = 0$$

čime je postignuto raspregnuto razmatranje uslova singularnosti.

PLANIRANJE KRETANJA

DEFINISANJE PROBLEMA PLANIRANJA KRETANJA

U ovom dijelu bavićemo se problemom planiranja kretanja robota. Kretanje je osnovna operacija kojom roboti obavljaju postavljene zadatke, bez obzira da li pri tome prenose različite predmete sa jednog mjesta na drugo, na primjer u zadacima opsluživanja alatnih mašina, ili na svom vrhu imaju pričvršćen alat kojim se zadatak neposredno izvršava, kao na primjer kod farbanja, ili lučnog zavarivanja. Mi ćemo u ovom poglavlju razmotriti različite načine za definisanje kretanja koje robot treba da izvrši. Cilj svake od metoda za planiranje kretanja jeste da se omogući korisniku, ili programeru robota, da na jednostavan, intuitivan način definiše osnovne parametre kretanja robota, a da upravljački softver robota na osnovu zadatih parametara generiše sve informacije o kretanju potrebne upravljačkom sistemu robota u toku izvršavanja kretanja.

Definišimo najprije osnovne pojmove vezane za planiranje kretanja robota:

- **Zadatak robota** koji često sadrži verbalne opise šta bi robot trebao da uradi, se uobičajeno sastoji se od više podzadataka, koji mogu dalje da se raščlanjuju, sve dok se ne dođe do niza kretanja i akcija koje robot treba da obavi. Na primjer, jedan zadatak mogao bi biti: sastavi proizvod od datih komponenti, ili, premjesti obradak iz mašine na paletu, itd.
- **Putanja** je skup tačaka kroz koje robot treba da prođe da bi izvršio zadato kretanje. Primijetimo da putanja nosi čisto geometrijsku informaciju. Primjeri za putanje su: prava linija, kružnica, helikoida, itd.
- **Trajektorija** je putanja kojoj je pridružena informacija o vremenu. Drugim riječima, trajektorija definiše u kom trenutku će se robot naći na kojoj poziciji sa putanje. Za razliku od putanje, trajektorija sadrži informacije i o brzinama i ubrzanjima tokom kretanja, a ne samo pozicije kroz koje će robot proći.

Na sadašnjem nivou razvoja upravljačkih sistema robota, uglavnom još uvek nije moguće direktno zadavati zadatke robotu. Obzirom da se svaki zadatak raščlanjuje dalje na skup putanja kroz koje robot treba da prođe, a zatim se putanjama dodjeljuje i vremenska zavisnost da bi se dobile trajektorije, to i dalje predstavlja previše složen problem za upravljački softver robota, a naročito kada se ima u vidu velika raznolikost zadataka koje roboti mogu da obavljaju. Zbog toga raščlanivanje zadatka na putanje obično obavlja korisnik, odnosno programer robota, a planiranje kretanja, odnosno generisanje putanja i trajektorija se obavlja uz pomoć upravljačkog softvera robota. Pri tome, programer zadaje samo osnovne parametre kretanja, na primjer, ukupno vrijeme ili brzina kretanja robota od početne do krajnje pozicije, itd.

Metode za planiranje kretanja možemo podijeliti u nekoliko kategorija, i na osnovu nekoliko kriterijuma, koji nijesu međusobno isključivi. Prva podjela može biti na osnovu karaktera zadatka koji robot obavlja. Postoje primjene robota kod kojih je isključivo važno da robot iz tačke A stigne u tačku B, pri čemu oblik putanje između njih nije od značaja.

Ovakav tip kretanja javlja se, na primjer, u zadacima transporta materijala, gde je najvažnije da materijal bude uzet na jednom, a ostavljen na drugom mjestu, a kako se robot kreće između, nije od naročitog interesa. Ovakav režim rada naziva se **"kretanje od tačke do tačke", (point to point motion),** ili **"režim uzmi pa ostavi" (pick and place operation).** Mi ćemo se na dalje služiti skraćenicom termina na engleskom, **PTP,** za obilježavanje ovakvog načina kretanja.

Drugu grupu primjena robota karakteriše činjenica da je važna cijela putanja po kojoj se robot kreće. Primjer za to je robotsko farbanje, gde putanja robota direktno određuje kako se nanosi boja na površinu koja se farba, ili robotsko lučno zavarivanje, gde putanja vrha robota treba da prati sastav koji se vari. Ovakve režime rada nazivamo **"kretanje po kontinualnim putanjama" (continuous path motion),** i na dalje ćemo ga označavati sa **CP,** skraćenicom termina na engleskom jeziku.

Jasno je da je planiranje **PTP** kretanja jednostavnije u odnosu na **CP** kretanje. Kod **PTP** dovoljno je zadati početnu i krajnju poziciju, kao parametre, i eventualno brzinu, ili vrijeme izvršavanja, a upravljači softver robota će onda sam generisati sve ostale potrebne informacije. Kod **CP** kretanja, potrebno je definisati putanju (koja može biti oblika jednostavnih matematičkih primitiva ali može biti i složena), a zatim treba definisati i vremenske informacije o kretanju duž putanje.

Druga podjela se može učiniti na osnovu prostora u kome se zadaje kretanje. Tako razlikujemo planiranje kretanja u unutrašnjim koordinatama od planiranja kretanja u spoljašnjim koordinatama.

Obzirom na to da je vrh robota relevantan za obavljanje zadatka, planiranje kretanja najprirodnije se obavlja u prostoru spoljašnjih koordinata, koje su vezane za kretanje vrha. Sa druge strane, kretanje robota odvija se u prostoru unutrašnjih koordinata, odnosno zglobova, pa je ponekad od interesa da se planiranje kretanja obavlja u ovom prostoru.

Mi smo već vidjeli da se primenom funkcija direktne ili inverzne kinematike, kretanje iz jednog prostora može preslikati u drugi prostor, naravno, uz sve teškoće, koje rješavanje inverzne kinematike nosi sa sobom. Treba reći da iako ove dvije podele nisu isključive, najčešće se planiranje **CP** kretanja obavlja u prostoru spoljašnjih koordinata, dok se planiranje **PTP** kretanja obavlja u prostoru unutrašnjih koordinata. Naravno, primenljive su i druge dvije kombinacije, kada je to potrebno ili poželjno.

Uočimo da rezultat planiranja kretanja treba da bude vremenska zavisnost pozicije i orijentacije vrha robota $\mathbf{r}(\mathbf{t})$, ili vremenska zavisnost pozicija zglobova $\mathbf{q}(\mathbf{t})$, koje treba da posluže kao referentne vrijednosti za upravljački sistem robota. Ovim vremenskim zavisnostima pozicija, po potrebi, mogu se pridružiti i brzine i ubrzanja.

Proučićemo nekoliko metoda za generisanje trajektorija, koje su primenljive za slučaj da u radnom prostoru robota nema prepreka, odnosno, da je robot slobodan da se kreće unutar celog radnog prostora. Ovaj pristup može se relativno lako proširiti i na slučaj kada unutar radnog prostora postoje statičke prepreke, čije su osobine, odnosno, veličina i položaj, unaprijed poznati i nepromjenljivi.

Slučaj kada se u radnom prostoru nalaze dinamičke prepreke, zahtijeva potpuno drugačiji pristup planiranju kretanja, a takođe i sofisticirani senzorski sistem koji će prepreke detektovati, i odrediti njihove karakteristike, kao i složenije algoritme koji mogu on-line donositi odluke o promjeni trajektorija. Problem planiranja kretanja u prisustvu dinamičkih prepreka izlazi van domena ovog kursa, i mi se njime nećemo baviti.

PLANIRANJE PTP KRETANJA

Obzirom na činjenicu da su kod PTP kretanja važne samo početna i krajnja tačka putanje, a da kretanje između njih može biti proizvoljno, planiranje kretanja u ovom režimu može se podjednako uspješno sprovoditi i u prostoru spoljašnjih, i u prostoru unutrašnjih koordinata. Ipak je uobičajeno da se planiranje kretanja obavlja u unutrašnjim koordinatama, primenom sledeće proste procedure. Početna r_i , i krajnja r_f tačka putanje obično se zadaju u spoljašnjim koordinatama, koje su od interesa za obavljanje postavljenog zadatka. Zatim se primjenom nekog od algoritama za rješavanje problema inverzne kinematike izračunaju pozicije u unutrašnjim koordinatama za početnu q_i i krajnju q_f tačku putanje.

Dobijanje informacija o početnoj i završnoj tački putanje moguće je izvesti na različite načine. Jedan pristup je već izložen, na osnovu podataka o položaju vrha, riješi se problem inverzne kinematike, i dobiju se položaji u unutrašnjim koordinatama. Drugi popularan način je da se robot, uz pomoć upravljačke konzole, fizički dovede u željene tačke prostora koje predstavljaju krajnje tačke putanje. U tim tačkama, zapamte se trenutne vrijednosti pozicija zglobova, i one predstavljaju polaznu osnovu za sintezu trajektorije. Kod savremenih robota, ovo pomjeranje vrha robota u željene pozicije u prostoru sve češće se obavlja u virtualnom svijetu računarske animacije.

Početna i krajnja pozicija putanje, bez obzira kako su dobijene, zajedno sa vremenskim parametrima, kao što je na primjer ukupno vrijeme izvršavanja pokreta, T_f , predstavljaju zatim polazne parametre za generisanje trajektorije između krajnjih tačaka. Obzirom da sama putanja nije od interesa za izvršavanje zadatka, zadak generisanja trajektorije moguće je riješiti na različite načine. Primijetimo da je problem generisanja trajektorije kod **PTP** kretanja na neki način analogan problemu interpolacije: date su nam početna i krajnja tačka, potrebno je odrediti neke interpolacione funkcije koje će nam odrediti vrijednosti pozicija između krajnjih tačaka.

Prije nego što se upustimo u rješavanje problema generisanja PTP trajektorija, razmotrimo koje bi osobine trebalo da posjeduje generisana trajektorija.

Iskustveno je poznato da živa bića, a mi ćemo kao primjer uzeti čovjeka, izvode uglavnom glatke pokrete. Zaista, kada pokrećemo naše ruke, to uglavnom činimo bez velikih trzaja. Ako bismo matematički analizirali pokrete ruke, mogli bismo reći da su pozicije zglobova ruke glatke funkcije vremena. Podsjetimo se da pojam glatka funkcija, znači ne samo da je funkcija kontinualna, već je i dovoljan broj izvoda funkcije takođe neprekidan. Kada govorimo o kretanju, prvi izvod pozicije je brzina, a drugi izvod je ubrzanje. Prema tome, možemo reći da glatko kretanje podrazumijeva kontinualnu promjenu pozicije, i pri tome kontinualne promjene brzine i ubrzanja.

Ovakv pristup kretanja kod živih bića je potpuno prirodan, jer dovodi do kretanja sa optimalnom potrošnjom energije. Intuitivno je jasno da nagle, ili skokovite promjene, bilo ubrzanja, brzine ili pozicije, zahtijevaju mnogo veći utrošak energije od kontinualnih promjena. Imajući ovo u vidu, i prilikom planiranja kretanja robota, težićemo da dobijeno kretanje u što većoj mjeri bude glatko.

Linearna interpolacija

Polazna osnova za generisanje trajektorije su početna \mathbf{q}_i i krajnja \mathbf{q}_f tačka putanje, zadate u prostoru unutrašnjih koordinata. Mi ćemo posmatrati izolovani slučaj kretanja, odnosno pretpostavićemo da robot kreće iz stanja mirovanja iz početne tačke putanje, a takođe, treba da se zaustavi u krajnjoj tački putanje. To drugim riječima znači $\dot{\mathbf{q}}_i = \mathbf{0}$, odnosno $\dot{\mathbf{q}}_f = \mathbf{0}$. Da bismo generisali trajektoriju, potrebno je još uspostaviti vremensku zavisnost pozicije robota. Pretpostavimo zato da ukupno trajanje pokreta treba da iznosi T_f . Bez umanjavanja opštosti, možemo pretpostaviti da je trenutak od koga računamo vrijeme jednak nuli, odnosno $T_i = 0$.

Primijetimo da je pozicija zglobova vektorska veličina. To dalje znači da i interpolacione funkcije koje tražimo treba da budu vektorske funkcije. Međutim, obzirom na to da kretanje zglobova može biti potpuno nezavisno, jer nas ne interesuje putanja po kojoj se vrh robota kreće, već samo njene krajnje tačke, možemo problem pojednostaviti, i zapravo rješavati n problema interpolacije, za svaki zglob robota pojedinačno. Pri tome, kao polazne podatke, uzimamo odgovarajuće vrste iz vektora položaja, a vrijeme izvršenja trajektorije je isto za sve zglobove.

Na dalje ćemo koristiti skalarne oznake za pozicije i brzine, a radi preglednosti izraza, izostavićemo oznaku indeksa koji pokazuje na koji zglob se izraz odnosi. Pri tome se podrazumijeva da se problem generisanje trajektorije rješava za svaki zglob pojedinačno. Najprostiji metod za interpolaciju jeste linearna interpolaciona funkcija data izrazom

$$q(t) = q_i \frac{T_f - t}{T_f} + q_f \frac{t}{T_f}, \quad (1)$$

dok se brzina može odrediti diferenciranjem izraza po vremenu t , čime se dobija

$$\dot{q}(t) = \frac{q_f - q_i}{T_f} \quad (2)$$

a nakon još jednog diferenciranja po vremenu dobija se izraz za ubrzanje

$$\ddot{q}(t) = 0 \quad (3)$$

Prethodne tri jednačine, izražavaju činjenicu da se linearna interpolacija vrši konstantnom brzinom, pa je ubrzanje jednako nuli. Kretanje konstantnom brzinom je osobina koja može biti od interesa u mnogim aplikacijama robota, pa je problem interpolacije naizgled riješen, i to veoma jednostavno.

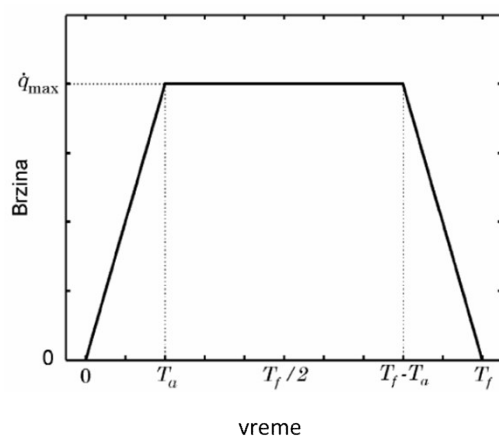
Međutim, možemo uočiti da je druga jednačina u suprotnosti sa graničnim uslovima za brzinu, odnosno zahtjevom da robot krene iz mirovanja i da se na kraju kretanja zaustavi. To znači da je brzina prekidna funkcija, jer prije početka kretanja brzina jednaka nuli, a neposredno nakon početka kretanja brzina je određena izrazom (2). Sličan prekid postoji i pri kraju kretanja, neposredno prije dostizanja krajnje tačke trajektorije, brzina je određena izrazom (2), a odmah zatim pada na nulu, jer se robot zaustavlja.

Nagle promjene brzine (npr. naglo startovanje ili zaustavljanje) dovode do izrazito velikih ubrzanja (teoretski beskonačno velikih) što zahtijeva adekvatne (dovoljno snažne) motore koji takva ubrzanja mogu da ostvare. Obzirom na neopravdanost ovakvog pristupa jasno je da je potrebno da se na početku trajektorije predvidi određeno vrijeme za ubrzanje, i na kraju trajektorije za usporavanje, što nas dovodi do trapeznog profila brzine.

Trapezni profil brzine

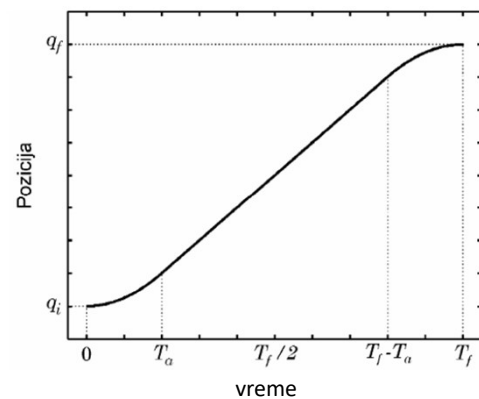
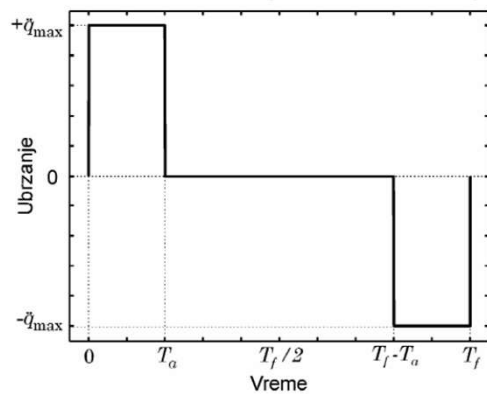
Trapezni profil brzine predstavlja zapravo realno izvodljivu implementaciju kretanja konstantnom brzinom. Naime, obzirom da robot polazi iz stanja mirovanja, a želimo da se kreće nekom konstantnom brzinom, potrebno je izvjesno vrijeme da se ta brzina dostigne. Takođe, zaustavljanje se ne odvija naglo, nego se pri kraju kretanja kontrolisano koči, i brzina smanjuje do nule.

Razmotrimo sada sa matematičkog aspekta opisano kretanje. Ponovo su date početna \mathbf{q}_i i krajnja pozicija \mathbf{q}_f , kao i ukupno vrijeme trajanja pokreta T_f . Ovim parametrima pridružimo još T_a , vrijeme predviđeno za ubrzanje, odnosno usporavanje. Interpolacione funkcije za poziciju, brzinu i ubrzanje kod trapeznog profila brzine prikazane su na Sl. 5.1.



Sa Sl. 5.1. uočavamo, da je u vremenskom intervalu od početnog trenutka do trenutka T_a ubrzanje pozitivno, konstantno i iznosi \ddot{q}_{\max} . Jasno je da se onda radi o ravnomjerno ubranom kretanju, pa brzina kretanja linearno raste, od vrijednosti nula, odnosno mirovanja, do vrijednosti \dot{q}_{\max} , dok se pozicija mijenja po kvadratnom zakonu. Ovaj interval predstavlja period ubrzanja robota

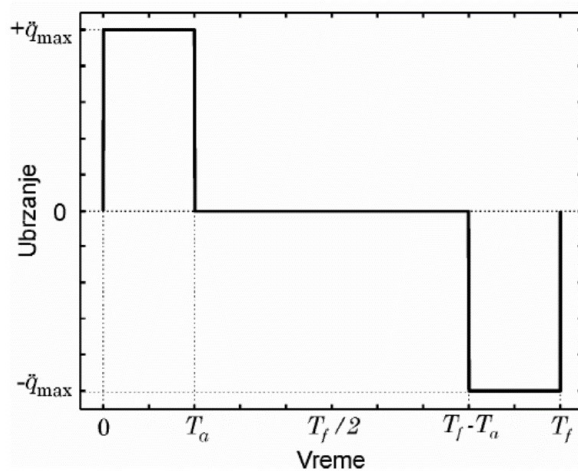
Sl. 5.1. Trapezni profil brzine



Obzirom na izgled grafikona brzine, ovakvo kretanje dobilo je naziv kretanje po trapeznom profilu brzine. Primijetimo još da brzina krstarenja, maksimalno ubrzanje i period ubrzavanja odnosno kočenja nijesu nezavisne veličine, jer mora da važi

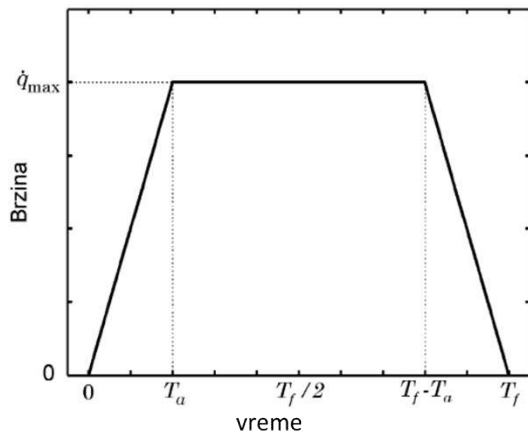
$$\dot{q}_{\max} = \ddot{q}_{\max} \cdot T_a \quad (4)$$

Izraze za interpolacione funkcije možemo odrediti na osnovu Sl. 5.1. Lako možemo zaključiti da je ubrzanje dato izrazom



$$\ddot{q}(t) = \begin{cases} +\ddot{q}_{\max} & 0 < t \leq T_a \\ 0 & T_a < t \leq T_f - T_a \\ -\ddot{q}_{\max} & T_f - T_a < t \leq T_f \end{cases}$$

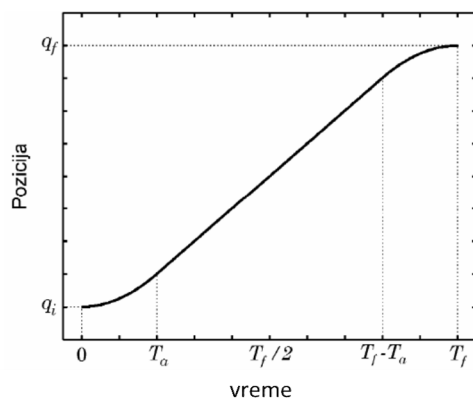
Brzinu možemo odrediti integracijom prethodne jednačine, vodeći računa o početnim uslovima i vezi datoj izrazom za maksimalnu brzinu, u sledećem obliku:



$$\dot{q}_{\max} = \ddot{q}_{\max} \cdot T_a$$

$$\dot{q}(t) = \begin{cases} \ddot{q}_{\max} \cdot t & 0 < t \leq T_a \\ \ddot{q}_{\max} \cdot T_a & T_a < t \leq T_f - T_a \\ \ddot{q}_{\max} \cdot (T_f - t) & T_f - T_a < t \leq T_f \end{cases}$$

Izraz za poziciju može se dobiti integracijom izraza za brzinu, vodeći ponovo računa o početnim uslovima, čime se dobija



$$q(t) = \begin{cases} q_i + 0.5 \cdot \ddot{q}_{\max} \cdot t^2 & 0 < t \leq T_a \\ q_i + \ddot{q}_{\max} \cdot T_a \cdot (t - T_a/2) & T_a < t \leq T_f - T_a \\ q_f - 0.5 \cdot \ddot{q}_{\max} \cdot (T_f - t)^2 & T_f - T_a < t \leq T_f \end{cases}$$

Podsjetimo se da smo pošli od pretpostavke da je zadat opseg promjene pozicije od q_i do q_f , ukupno vrijeme izvršavanja pokreta T_f , kao i vrijeme ubrzanja, odnosno kočenja T_a . Jedina nepoznata veličina u izrazima je maksimalna vrednost ubrzanja \ddot{q}_{\max} , koja, očigledno, mora da zavisi od polaznih parametara.

Potrebnu vrijednost maksimalnog ubrzanja odredićemo na sledeći način. Promjena vrijednosti pozicije od početne do krajnje, odnosno pređeni put, mora biti jednak površini ispod grafikona brzine. S obzirom na činjenicu da grafikon ima oblik trapeza, lako uspostavljamo vezu

$$q_f - q_i = (\ddot{q}_{\max} \cdot T_a) \cdot (T_f - T_a)$$

Sada potrebnu vrednost ubrzanja možemo odrediti kao

$$\ddot{q}_{\max} = \frac{q_f - q_i}{T_a \cdot (T_f - T_a)}$$

Primijetimo još da je u svim prethodnim izrazima logično pretpostavljeno da važi $T_a \leq \frac{T_f}{2}$,

odnosno, periodi ubrzavanja, odnosno kočenja, ne mogu biti duži od polovine ukupnog vremena.

U specijalnom slučaju, kada je $T_a = 0.5T_f$, period ubrzavanja i usporavanja traju po polovinu ukupnog vremena kretanja, odnosno, čim se dostigne brzina krstarenja počinje kočenje. Trapezni profil brzine se tada deformiše, jer se gubi središnji dio kretanja konstantom brzinom, pa za ovakvo kretanje kažemo da se odvija po trougaonom profilu brzine.

Već je rečeno da su kretanja robota, sa stanovišta brzine i ubrzanja, ograničena primijenjenim aktuatorima. Drugim riječima, postoji maksimalno ubrzanje koje dati aktuator može proizvesti u posmatranom zglobu, kao što je i maksimalna brzina kretanja zgloba ograničena. Zbog toga je u praksi uobičajeno da se, umjesto zadavanja vremena trajanja pokreta, i perioda ubrzavanja odnosno usporavanja, zadaju maksimalno ubrzanje, odnosno, željena brzina krstarenja, a da se vremenski parametri izračunaju na osnovu njih.

Prema tome, neka je poznato q_i , q_f , \dot{q}_{\max} i \ddot{q}_{\max} , pri čemu, vrijednost brzine krstarenja i maksimalna vrijednost ubrzanja mogu biti zaista fizičke maksimalne vrijednosti. Tada se dobija najkraće vrijeme trajanja pokreta. Na osnovu veze (4), lako određujemo vrijeme potrebno za ubrzavanje, odnosno kočenje, kao

$$T_a = \frac{\dot{q}_{\max}}{\ddot{q}_{\max}}.$$

Smjenom dobijenog izraza u jednačinu (5), možemo odrediti ukupno trajanje pokreta kao

$$T_f = \frac{q_f - q_i}{\dot{q}_{\max}} + \frac{\dot{q}_{\max}}{\ddot{q}_{\max}}.$$

Trapezni profil brzine se često primjenjuje u praksi, ne samo kod robota, već i kod drugih uređaja koji zahtijevaju kretanje konstantnom brzinom, kao što su na primjer, alatne mašine, portalne dizalice, elektromotorni pogoni, itd.

Primitimo na kraju da je ubrzanje prekidna funkcija vremena, jer se skokovito mijenja na granicama perioda ubrzavanja i kočenja. Ova nagla promjena ubrzanja, odražava se kao trzaj, koji može pobuditi neželjene oscilacije strukture robota, a takođe, nagla promjena ubrzanja može biti teško ostvariva sa stanovišta aktuatora. Zbog toga ćemo pokušati da pronađemo interpolacione funkcije, koje bi nam omogućile kontinualnu promjenu, ne samo pozicije i brzine, već i ubrzanja. Kao pogodni kandidati za interpolacione funkcije, nameću se polinomi, koje ćemo razmotriti u narednom dijelu

Interpolacija polinomom trećeg stepena

Kao i ranije, polazna osnova za generisanje trajektorije su početna \mathbf{q}_i i krajnja \mathbf{q}_f tačka putanje, i ukupno trajanje pokreta \mathbf{T}_f . Prema tome, mi imamo zadata četiri granična uslova, od kojih se dva odnose na početnu tačku putanje,

$$q(0) = q_i, \quad \dot{q}(0) = 0$$

I još dva koja se odnose na krajnju tačku putanje

$$q(t_f) = q_f, \quad \dot{q}(t_f) = 0$$

Zadatak nam je da pronađemo glatku interpolacionu funkciju koja će nam odrediti pozicije tokom vremena, između početne i krajnje tačke. Kao moguće rješenje prirodno se nameću polinomne funkcije, jer je opšte poznato da su polinomi neprekidne beskonačno diferencijabilne glatke funkcije sa svim svojim izvodima.

Prije nego što pređemo na određivanje interpolacionih polinoma, razmotrimo najprije mogućnost da izvršimo normalizaciju problema, što će nam olakšati ne samo izlaganje, nego i primjenu dobijenih polinoma na sintezu kretanja različitih zglobova. Uvedimo najprije normalizaciju po vremenu, uvođenjem smjene:

$$\tau = \frac{t}{T_f}$$

Čime se realno vrijeme \mathbf{t} iz intervala $\mathbf{t} \in [0, \mathbf{T}_f]$ preslikava u normalizovano vrijeme $\boldsymbol{\tau}$ na normalizovanom intervalu $\boldsymbol{\tau} \in [0, 1]$.

Uvedimo zatim i normalizovanu poziciju \mathbf{p} tako da se interval promjene pozicije \mathbf{q} preslikava na normalizovani interval

$$p = \frac{q - q_i}{q_f - q_i}, \quad [q_i, q_f] \xrightarrow{p(q)} [0, 1]$$

Nakon ovih normalizacija, pozicija zgloba može se odrediti na osnovu

$$q(t)|_{t=\tau T_f} = q_i + (q_f - q_i)p(\tau)$$

Diferenciranjem izraza možemo odrediti brzinu zgloba u funkciji normalizovane brzine, kao

$$\dot{q}(t) = (q_f - q_i)p'(\tau) \frac{d\tau}{dt}, \quad \text{odnosno} \quad \dot{q}(t)|_{t=\tau T_f} = \frac{q_f - q_i}{T_f} p'(\tau)$$

Na sličan način možemo odrediti i ubrzanje zgloba u funkciji normalizovanog ubrzanja kao

$$\ddot{q}(t)|_{t=\tau T_f} = \frac{q_f - q_i}{T_f^2} p''(\tau)$$

Dakle, problem sinteze interpolacione funkcije za kretanje zgloba u realnom vremenu, zamijenili smo sintezom interpolacione funkcije za normalizovanu poziciju u normalizovanom vremenu. Pri tome je izvršena i normalizacija graničnih uslova, u obliku

$$p(0) = 0, \quad \dot{p}(0) = 0,$$

$$p(1) = 1, \quad \dot{p}(1) = 0,$$

Obziroma na to da imamo četiri zadata uslova, na osnovu njih možemo odrediti polinom trećeg stepena, koji ima ukupno četiri koeficijenta, u obliku

$$p_3(\tau) = a_3\tau^3 + a_2\tau^2 + a_1\tau + a_0,$$

Da bismo iskoristili granične uslove po brzini normalizovane pozicije, diferencirajmo poslednji izraz , čime se dobija

$$p_3'(\tau) = 3a_3\tau^2 + 2a_2\tau + a_1$$

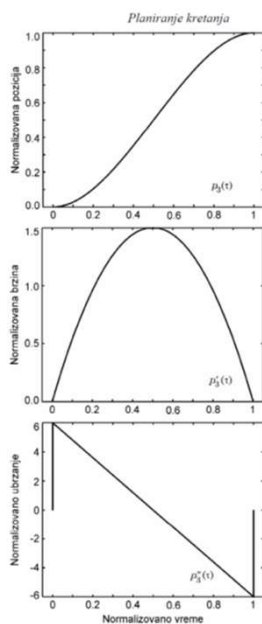
Smjenom graničnih uslova u poslednje dvije relacije lako možemo izračunati slobodne članove i preostale koeficijente

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 3, \quad a_3 = -2$$

tako da su normalizovani interpolacioni polinom brzina i ubrzanje dati sa

$$p_3(\tau) = -2\tau^3 + 3\tau^2 \quad p_3'(\tau) = -6\tau^2 + 6\tau \quad p_3''(\tau) = -12\tau + 6$$

Grafički izgled normalizovane pozicije, brzine i ubrzanja prikazan je na Sl. 5.2.



Sa slike uočavamo da je normalizovana pozicija zaista glatka funkcija vremena, kao i normalizovana brzina. Vidimo takođe da je normalizovano ubrzanje linearna funkcija normalizovanog vremena, međutim nije neprekidna.

Kao što je već rečeno, diskontinuiteti u ubrzanju su nepoželjni, jer mogu pobuditi oscilacije u kretanju robota. Takođe, moramo biti svjesni činjenice da ubrzanje ne može trenutno da se promijeni. Skokovita promjena ubrzanja implicira skokovitu promjenu sile ili momenta aktuatora, a to dalje znači veliki utrošak energije u jako kratkom vremenskom intervalu, odnosno, beskonačnu snagu, što svakako nije fizički ostvarljivo. Drugim riječima, kretanje dobijeno interpolacionim polinomom trećeg stepena nije dovoljno glatko, jer posjeduje prekide u ubrzanju.

Ako želimo da ove prekide eliminišemo, možemo da pokušamo da povećamo stepen interpolacionog polinoma.

Interpolacija polinomom petog stepena

Ako želimo da dobijemo kretanje kod koga ubrzanje neće imati diskontinuitete, onda graničnim uslovima po poziciji i brzini moramo pridružiti i granične uslove po ubrzanju. Naime, s obzirom na to da robot kreće iz stanja mirovanja, ubrzanje u početnom trenutku mora biti jednako nuli. Takođe, obzirom na to da se na kraju kretanja robot zaustavlja, ubrzanje mora biti jednako nuli. Takođe, normalizovano ubrzanje treba da je kontinualna funkcija normalizovanog vremena, pri čemu imamo dva nova granična uslova vezana za ubrzanje. Broj graničnih uslova povećao se sada na šest:

$$\begin{aligned} p(0) = 0, & \quad p'(0) = 0, & \quad p''(0) = 0, \\ p(1) = 1, & \quad p'(1) = 0, & \quad p''(1) = 0, \end{aligned}$$

tako da možemo potražiti interpolacioni polinom petog stepena, oblika

$$p_5(\tau) = a_5\tau^5 + a_4\tau^4 + a_3\tau^3 + a_2\tau^2 + a_1\tau + a_0$$

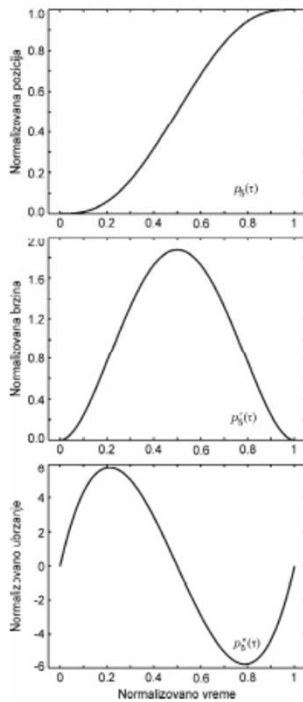
pri čemu koeficijente a_i , ($i = 0, \dots, 5$) određujemo iz graničnih uslova. Konačno rješenje je oblika:

$$p_5(\tau) = 6\tau^5 - 15\tau^4 + 10\tau^3$$

$$p'_5(\tau) = 30\tau^4 - 60\tau^3 + 30\tau^2$$

$$p''_5(\tau) = 120\tau^3 - 180\tau^2 + 60\tau$$

Uporedimo dobijene rezultate sa interpolacionim polinomom trećeg stepena. Primećujemo da je kriva normalizovane pozicije nešto strmija u sredini vremenskog intervala, a nešto više zaravnjena na početku i kraju kretanja, što je posledica činjenice da je tada ubrzanje jednako nuli. Na grafikonu normalizovane brzine, primećujemo da kriva ima izraženiji zvonasti oblik, pri čemu je kriva uža nego kod polinoma trećeg stepena. Obzirom na to da je pređeni put jednak u oba slučaja, a to je površina ispod krive brzine, maksimalna brzina, koja se ponovo javlja u sredini intervala, veća je nego kod polinoma trećeg reda.



Normalizovana pozicija, brzina i ubrzanja kod interpolacije polinomom petog stepena

Vidimo da smo u slučaju interpolacije polinomom petog stepena uspjeli da ostvarimo glatko kretanje koje nema prekide u ubrzanju. Međutim, ni to ponekada nije dovoljno, jer može postojati potreba da i viši izvodi pozicije budu kontinualni. Tada moramo primijeniti polinome još većeg reda.

U nekim aplikacijama robota, od interesa je da izvod ubrzanja, odnosno, treći izvod pozicije, takođe bude kontinualna funkcija. Ovaj izvod ubrzanja u literaturi se obično označava kao "trzaj", što bi bio slobodan prevod engleskog termina "jerk". Da bi se to postiglo, graničnim uslovima dodaju se još uslovi po trećem izvodu pozicije.

Interpolacija polinomom sedmog stepena

$$p'''(0) = 0, \quad p'''(1) = 0, \quad \text{Dodatni granični uslovi}$$

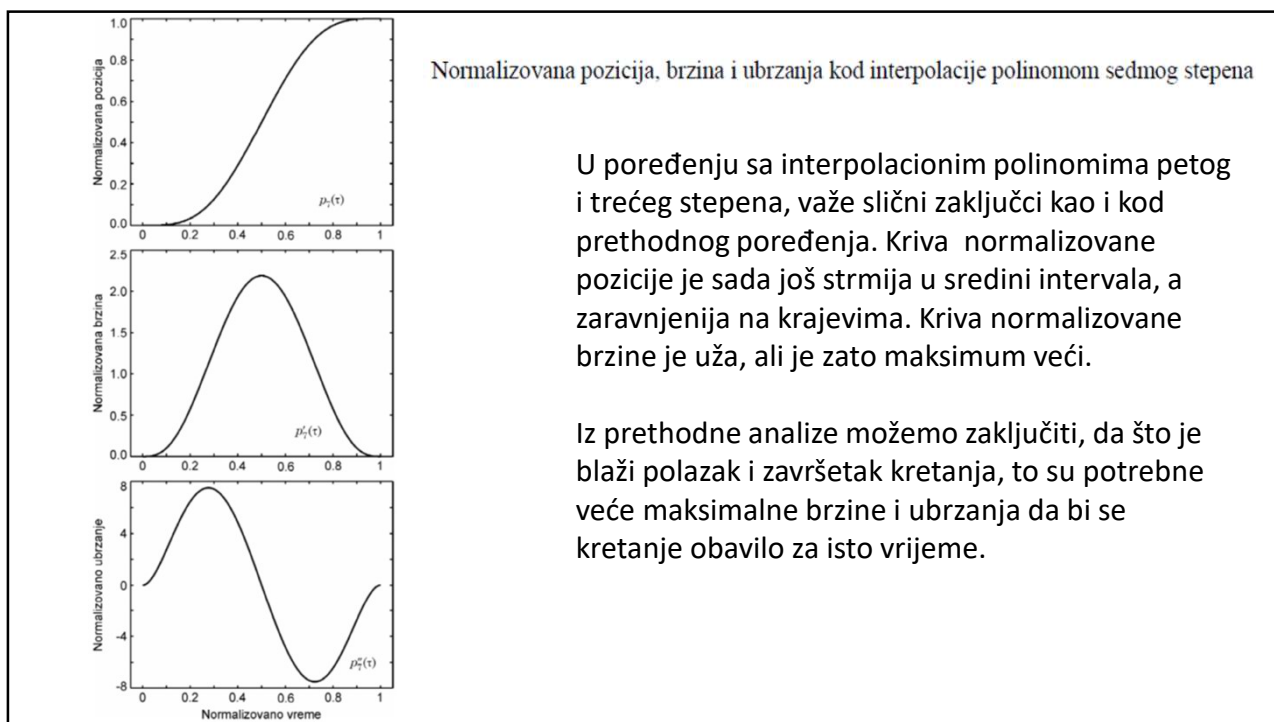
Čime se ukupan broj uslova povećao na osam, pa možemo konstruisati interpolacioni polinom sedmog stepena. Smjenom graničnih uslova u izraz za polinom i njegove izvode, i rješavanjem sistema jednačina određuju se koeficijenti polinoma. Konačno rješenje je oblika

$$p_7(\tau) = -20\tau^7 + 70\tau^6 - 84\tau^5 + 35\tau^4$$

$$p'_7(\tau) = -140\tau^6 + 420\tau^5 - 420\tau^4 + 140\tau^3$$

$$p''_7(\tau) = -840\tau^5 + 2100\tau^4 - 1680\tau^3 + 420\tau^2$$

Grafički prikaz normalizovane pozicije, brzine i ubrzanja dat je na sledećoj slici.



Primjena normalizovanih interpolacionih polinoma

Primjenom normalizacije po vremenu, i po vrijednosti, uspjeli smo da izraze za polinome različitih stepena učinimo nezavisnim od stvarnih parametara kretanja. U praksi, da bi se izrazi iskoristili, izračunavanja iz domena normalizovanih polinoma, moraju se vratiti u realno vrijeme, i realni opseg pozicija. Pri tome je opseg promene pozicija zadat, odnosno, poznate su početna \mathbf{q}_i , i krajnja pozicija \mathbf{q}_f . Već smo vidjeli, da se iz razloga limitiranih mogućnosti aktuatora, u praksi češće barata sa maksimalnim dozvoljenim ubrzanjima i brzinama, nego sa zadatim vremenom trajanja pokreta. Razlog tome je vrlo jednostavan, ako se zada suviše kratko vrijeme, trajektorija se ne može fizički ostvariti. Ako se pak, zada suviše dugačko vreme, onda se pokreti izvode nepotrebno sporo.

Razmotrimo postupak određivanja vremena izvršenja trajektorije, pod pretpostavkom da su poznate maksimalne dozvoljene veličine brzine i ubrzanja zgloba.

Obzirom na vezu između brzine zgloba i normalizovane brzine može se uspostaviti relacija između njihovih maksimalnih vrijednosti

$$\dot{q}_{\max} = \frac{q_f - q_i}{T_{fv}} p'_{\max}$$

odakle se određuje potrebno vrijeme trajanja pokreta sa stanovišta brzine, kao

$$T_{fv} = \frac{q_f - q_i}{\dot{q}_{\max}} p'_{\max}$$

Sličnom analizom može se odrediti potrebno vrijeme izvršavanja pokreta sa stanovišta ubrzanja, kao

$$T_{fa} = \sqrt{\frac{q_f - q_i}{\ddot{q}_{\max}} p''_{\max}}$$

Za potrebno vrijeme trajanja pokreta zgloba usvaja se restriktivnije rješenje, odnosno duže vrijeme,

$$T_f = \max\{T_{fv}, T_{fa}\}$$

da bi se izbjeglo premašivanje fizički ostvarljivih granica.

Imajući u vidu da je izložena procedura nezavisna za svaki zglob, a da zglobovi mogu imati i različite opsege zadatog kretanja, i različite maksimalne vrijednosti brzine i ubrzanja, nakon završetka analize sprovedene izračunavanjem jednačina za svaki od zglobova, imaćemo zapravo skup od n određenih vremena trajanja pokreta

$$\{T_{f1}, \dots, T_{fi}, \dots, T_{fn}\}$$

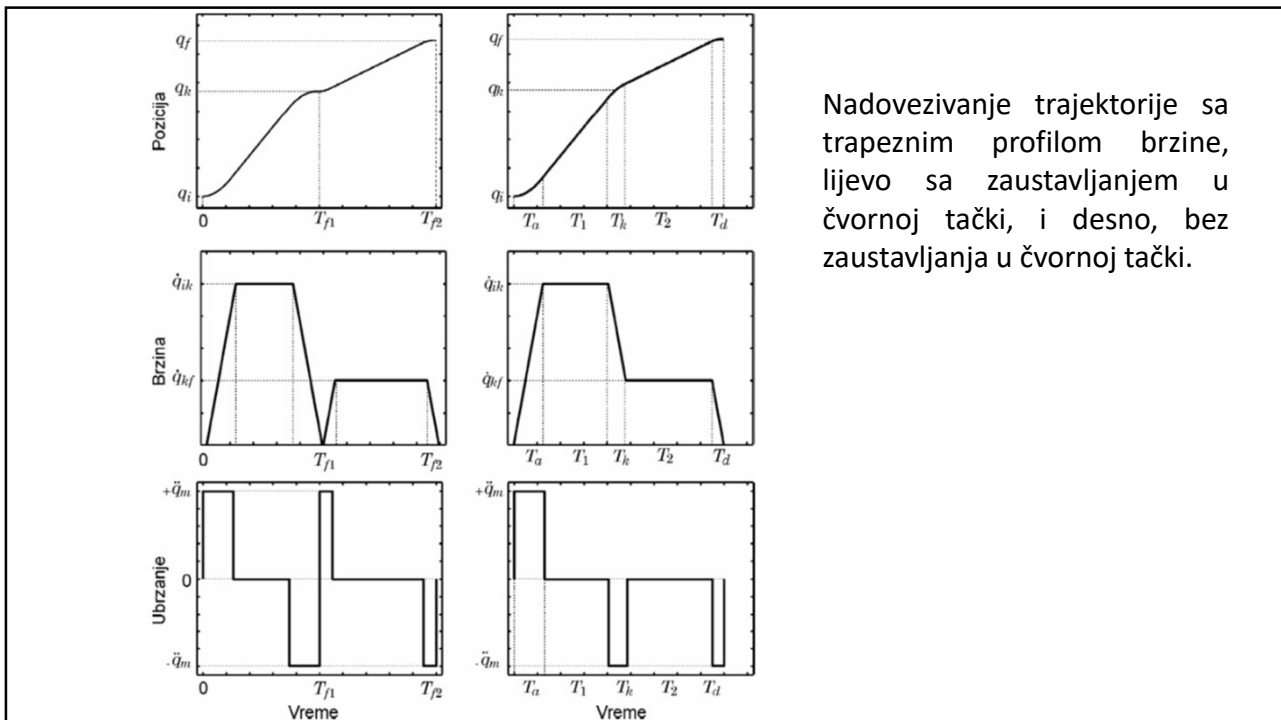
Imamo sada dvije mogućnosti. Prema prvoj, možemo pustiti da se svaki zglob kreće prema svojim maksimalnim parametrima. Pri tome, u opštem slučaju, vremena kretanja za pojedine zglobove biće različita, odnosno, neki zglobovi će ranije završiti pokret od drugih. Prema drugom pristupu, za vrijeme izvršenja pokreta usvojicemo najduže izračunato vrijeme pri čemu će trajanje pokreta za sve zglobove biti jednako. Primijetimo da se pri ovom pokretu, maksimalnom brzinom kreće samo zglob za čije kretanje je potrebno najviše vremena, a preostali zglobovi se usporavaju, tako da vrijeme izvršenja pokreta bude jednako.

$$T_f = \max_i(T_{fi})$$

Nadovezivanje trajektorija

Sve tehnike za sintezu trajektorije koje smo do sada obradili, rješavale su problem izolovanog kretanja od tačke do tačke, odnosno, robot kreće iz stanja mirovanja, i u završnoj tački putanje se ponovo zaustavlja. Sada smo zainteresovani za problem nadovezivanja trajektorija, odnosno, za slučaj kada postoji više segmenata kretanja, pri čemu je završna tačka prethodnog segmenta ujedno i početna tačka narednog, i tako redom. Ove središnje tačke između dva segmenta kretanja često se nazivaju čvorne tačke putanje, prema engleskom terminu "path knot point,,.

Jedno moguće rješenje se samo po sebi nameće. Tretirajmo svaki od segmenata kretanja nezavisno od ostalih, primjenom neke od interpolacionih funkcija koje smo već razmotrili. Time se problem sinteze trajektorije svodi na problem sinteze segmenata trajektorije, koji je već riješen. Osnovna mana ovakvog pristupa je u tome, što će u čvornim tačkama doći do zaustavljanja robota. Ovakvo kretanje, osim što djeluje neprirodno, jer je isprekidano čestim zaustavljanjem, zahtijeva i duže vrijeme izvršavanja u odnosu na kretanje kod kojeg zaustavljanja ne bi bilo.



Nadovezivanje trajektorije sa trapeznim profilom brzine, lijevo sa zaustavljanjem u čvornoj tački, i desno, bez zaustavljanja u čvornoj tački.

Razmotrimo sada problem nadovezivanja trajektorija kod polinomne interpolacije. Nadovezivanje se može sada organizovati na drugačiji način. Naime, dodavanje novih čvornih tačaka predstavlja zapravo dodavanje novih uslova, odnosno novih podataka o trajektoriji. Zbog toga se stepen interpolacionog polinoma može povećati. Ako je, pored pozicije čvorne tačke, zadata i brzina u njoj, onda je broj uslova, odnosno količina informacija o trajektoriji još veća, pa i stepen interpolacionog polinoma može biti veći. Problem polinomne interpolacije kroz čvorne tačke dobro je proučen u numeričkoj matematici, a jedinstveno rješenje za problem sa n čvornih tačaka daje **Lagranžov** interpolacioni polinom n -tog stepena, odnosno, ako su pored pozicija zadate i brzine u m čvornih tačaka, **Hermitov** interpolacioni polinom $(n + m)$ -tog stepena.

Napomenimo odmah da se ovakav pristup veoma rijetko koristi u praksi, iako na prvi pogled djeluje atraktivno. Naime, sa povećanjem broja čvornih tačaka, raste stepen interpolacionog polinoma. Iz algebre je poznato da polinom n -tog stepena ima tačno n nula, pa polinomi visokog stepena teže ka oscilatornom ponašanju. Iako Lagranžov i Hermitov algoritam garantuju da će interpolacioni polinom proći kroz čvorne tačke, nema nikakvih pokazatelja o ponašanju polinoma između čvornih tačaka, a ono je najčešće oscilatorno.

To bi zapravo značilo, da krećući se od jedne čvorne tačke ka narednoj, robot napravi preskok pa se vrati u čvornu tačku, ili najprije pođe u suprotnu stranu, pa onda promijeni smjer i vrati se u čvornu tačku. Svakako, ovo nije ponašanje kakvo bismo željeli da vidimo kod robota. Zbog toga se pribjegava drugačijem rješenju. Ukupan interval interpolacije dijeli se na podintervale, izneđu čvornih tačaka. Na svakom podintervalu usvaja se interpolaciona funkcija u obliku polinoma nižeg stepena, a sistemu uslova, odnosno ograničenja, pored zadatih čvornih tačaka, dodaju se uslovi kontinualnosti brzine odnosno ubrzanja u čvornim tačkama. Drugim riječima, iako brzina i ubrzanje u čvornim tačkama nijesu unaprijed određeni, zahtijeva se njihova neprekidnost, odnosno, vrijednosti dobijene na osnovu prethodnog i narednog interpolacionog polinoma moraju biti jednake. Ovakva vrsta interpolacije naziva se u matematici interpolacija splajnovima, a kada su u pitanju polinomi trećeg stepena, govorimo o interpolaciji kubnim splajnovima. Teorija splajnova izlazi izvan okvira ovog kursa.

PLANIRANJE CP KRETANJA

Pod **CP** kretanjem podrazumijevamo kretanje po kontinualnim putanjama. Imajući u vidu da je za obavljanje zadatka važno kretanje vrha robota, prirodno je da se planiranje ovakvog kretanja obavlja u prostoru spoljašnjih koordinata.

Planiranje CP kretanja je složenije u odnosu na planiranje PTP kretanja iz više razloga. Najprije, kod PTP kretanja, putanje i trajektorije koje nijesu bile od interesa birane su tako da budu jednostavne funkcije. Ovdje to nije slučaj, putanja može biti proizvoljno složena u zavisnosti od tipa zadatka koji robot obavlja, pa prema tome i trajektorija može imati složenu zavisnost tokom vremena. Dalje, kao što ćemo vidjeti u nastavku, za izvršavanje trajektorija najčešće je potrebna informacija o kretanju zglobova. To znači da se problem inverzne kinematike mora riješiti za svaku tačku trajektorije, što može biti složen, računski zahtjevan, pa samim tim i dugotrajan proces.

Jedan pristup sintezi CP kretanja je da se trajektorija zada kao skup bliskih čvornih tačaka kroz koje vrh robota treba da prođe, a da se zatim izvrši interpolacija ovih tačaka. Ovakav pristup ne razlikuje se mnogo od do sada izloženih postupaka za sintezu PTP trajektorija, osim što je generalno potrebno mnogo više čvornih tačaka. Interpolacija splajnovima može se i u ovom slučaju pokazati kao vrlo dobro i primjenljivo rješenje.

Razmotriti ćemo sada drugačiji pristup, a to je analitičko zadavanje putanja i trajektorija vrha robota. Kao što je to već rečeno, u najopštijem slučaju, pozicija i orijentacija vrha robota u prostoru može se opisati sa šest nezavisnih veličina, odnosno vektorom

$$r = \begin{bmatrix} p_{3 \times 1} \\ o_{3 \times 1} \end{bmatrix}_{6 \times 1},$$

gde p predstavlja podvektor koji sadrži tri parametra koja opisuju poziciju vrha robota u prostoru, dok o predstavlja podvektor koji sadrži tri parametra koja predstavljaju orijentaciju vrha robota

U specifičnim slučajevima, vektor r može imati i manje dimenzija od šest, kao na primjer, kada nije potrebna potpuna orijentacija, ili orijentacija uopšte nije od interesa, odnosno u slučajevima ravanskog kretanja, itd. Mi ćemo na dalje razmatrati opšti slučaj, a problem sinteze trajektorije razdvojićemo na problem definisanja pozicije, i problem zadavanja orijentacije, koje ćemo zasebno izučiti.

Parametrizovano zadavanje pozicije u prostoru

Kada se radi o definisanju pozicije u prostoru, tada je vektor pozicije vrha robota najčešće određen u pravouglom dekartovom koordinatnom sistemu

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

gde su x , y i z koordinate željene pozicije vrha izražene u odnosu na referentni koordinatni sistem. Moguće su naravno i druge vrste koordinatnih sistema, na primjer, polarno- cilindrični, ili sferni koordinatni sistem, ali su dekartove pravouglo koordinatne najjasnije i intuitivno najlakše za rad, pa se zbog toga najčešće i koriste.

Imajući u vidu da želimo da izvršimo sintezu trajektorije u prostoru analitičkim putem, najprije moramo pronaći analitički izraz za putanju u prostoru. Primijetimo da je putanja Π zapravo kontinualni skup tačaka u prostoru koji odgovaraju nekoj liniji u prostoru. Imajući to u vidu, Π je zapravo jednodimenzionalni skup, pa se može pronaći jedan parametar σ , tako da putanja bude opisana sa

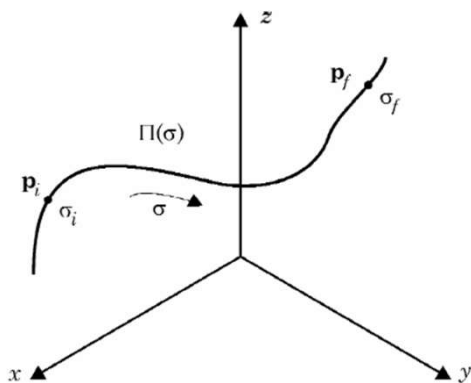
$$p = \Pi(\sigma),$$

odnosno

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x(\sigma) \\ y(\sigma) \\ z(\sigma) \end{bmatrix},$$

gde je σ neki geometrijski parametar koji opisuje putanju. Ovaj koncept prikazan je na Sl. 5.6 .

Početna tačka putanje p_i određena je početnom vrijednošću parametra σ_i , dok je krajnja tačka putanje p_f određena krajnjom vrijednošću parametra σ_f .



Sl. 5.6. Parametarska definicija putanje u prostoru

Primijetimo još da je putanja orijentisana, jer porast vrijednosti parametra σ određuje smjer kretanja duž putanje. Takođe, putanja može biti i zatvorena, kada je $p_f = p_i$, ili otvorena, kao što je to slučaj na Sl. 5.6. Prema tome, promjena parametra σ , od početne vrijednosti σ_i do krajnje vrijednosti σ_f u potpunosti definiše položaj vrha robota duž putanje. Ako parametru σ , odnosno njegovoj promjeni, pridružimo i vremensku zavisnost $\sigma = \sigma(t)$, tada je u potpunosti definisana i trajektorija vrha robota

Funkcionalna zavisnost $\sigma = \sigma(t)$ definiše se preslikavanjem vremenskog intervala $[T_i, T_f]$ za koji treba da se obavi kretanje duž putanje, na interval promjene geometrijskog

parametra koji definiše putanju od početne do krajnje tačke: $[T_i, T_f] \xrightarrow{\sigma = \sigma(t)} [\sigma_i, \sigma_f]$,

pri tome, veličina vremenskog intervala utiče na brzinu kretanja, odnosno, kraći vremenski interval znači brže kretanje, i obrnuto, dok oblik funkcije $\sigma(t)$ definiše profil kretanja.

Za profil kretanja $\sigma(t)$ možemo iskoristiti neku od funkcija koje smo već razmatrali kod sinteze PTP kretanja: trapezni ili trougaoni profil brzine, ili polinomne interpolacije odgovarajućeg stepena, itd.

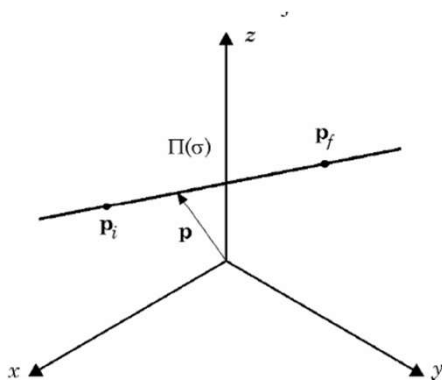
Na ovaj način, problem sinteze trajektorije rješava se u dva koraka. U prvom je potrebno odrediti izraze za parametrizovanu trajektoriju u prostoru, dok se u drugom koraku geometrijskom parametru pridružuje željeni vremenski profil. Razmotrimo sada malo detaljnije prirodu geometrijskog parametra σ . Već je rečeno da on određuje položaj tačke duž putanje Π , odnosno, duž krive u prostoru. Podsjetimo se da kriva u prostoru može biti zadana eksplicitno svojim jednačinama, ili kao presjek dvije površi u prostoru. U svakom slučaju, uvijek postoji mogućnost da se putanja parametrizuje. Od svih mogućih parametara kojima je moguće opisati putanju, kao najvažniji se izdvaja krivolinijska koordinata \mathbf{s} , definisana kao pređeni put, ili dužina luka krive od neke fiksirane tačke usvojene za koordinatni početak, do posmatrane tačke.

Krivolinijska koordinata \mathbf{s} najčešće se definiše preko svog diferencijalnog priraštaja gde su dx , dy i dz diferencijalni priraštaji krive duž koordinatnih osa.

$$ds = \sqrt{dx^2 + dy^2 + dz^2}$$

Kod jednostavnijih putanja, krivolinijska koordinata \mathbf{s} se prirodno nameće kao najpodesnija veličina za parametrizaciju. Kod složenijih putanja, krivolinijska koordinata može biti složena, ili transcendentna funkcija, pa se kao parametar usvaja neka druga, pogodna veličina.

Razmotrimo sada nekoliko primjera određivanja parametrizovanih putanja. Uzmimo najprije najjednostavniji primjer, kretanja po pravoj liniji u prostoru, kao što je to prikazano na Sl. 5.7.



Sl. 5.7. Putanja u obliku prave linije u prostoru

Podsjetimo se da se u analitičkoj geometriji prava u prostoru, koja prolazi kroz dvije tačke zadate vektorima \mathbf{p}_i i \mathbf{p}_f može definisati kao

$$\mathbf{p} - \mathbf{p}_i = \sigma(\mathbf{p}_f - \mathbf{p}_i),$$

gde jednačina izražava činjenicu da vektor povučen iz početne tačke \mathbf{p}_i do bilo koje tačke na pravoj \mathbf{p} , mora biti kolinearan vektoru između tačaka \mathbf{p}_i i \mathbf{p}_f , pri čemu je σ faktor kolinearnosti. Jednačina se može preformulirati tako da odmah dobijemo parametrizovanu putanju oblika

$$\mathbf{p} = \mathbf{p}_i + \sigma(\mathbf{p}_f - \mathbf{p}_i) ..$$

Ili razvijeno po koordinatama

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_i + \sigma(x_f - x_i) \\ y_i + \sigma(y_f - y_i) \\ z_i + \sigma(z_f - z_i) \end{bmatrix}.$$

Ako su \mathbf{p}_i i \mathbf{p}_f početna, odnosno krajnja tačka putanje, tada je opseg promjene parametra σ u intervalu $[0,1]$.

U ovom slučaju, parametar σ koji je uveden kao čisto geometrijski parametar, ima prirodu krivolinijske koordinate s . Zaista određivanjem diferencijala gornje matrice i definicije $d\mathbf{s}$ dobijamo

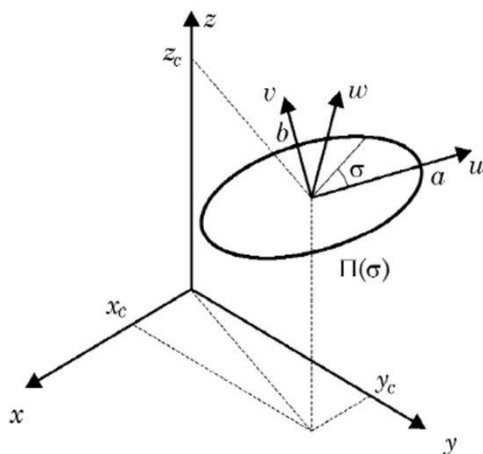
$$ds = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2 + (z_f - z_i)^2} \cdot d\sigma,$$

odnosno, priraštaj krivolinijske koordinate, ili pređenog puta s , proporcionalan je priraštaju geometrijskog parametra σ .

Da bi problem sinteze trajektorije bio u potpunosti riješen, preostaje nam samo da izaberemo vremensku zavisnost, odnosno profil promene $\sigma(t)$. Kao što je već rečeno, za ove svrhu možemo iskoristiti na primjer trapezni profil brzine, pri čemu će se, zbog prirode parametra σ , zaista dobiti trapezna promjena brzine vrha robota.

Razmotrimo sada problem sinteze trajektorije u slučaju da želimo da se vrh robota kreće po nekoj elipsi u prostoru. Obzirom da je elipsa ravanska kriva, i problem određivanja parametarske reprezentacije putanje riješićemo najprije u ravni koja sadrži elipsu, a zatim ćemo tu ravan rotirati i translirati u prostoru na željeni način. Ova situacija prikazana je na Sl.5.8.

Označimo sa u , v i w koordinate koordinatnog sistema koji je vezan za elipsu, tako da se njegov koordinatni početak poklapa sa centrom elipse, a ose u i v određuju ravan u kojoj elipsa leži. Neka su poluose elipse određene sa a i b , tada se jednačina elipse može napisati kao



Sl. 5.8. Izgled putanje oblika elipse u prostoru

$$p = p_c + R \cdot p_{uvw}$$

$$p_c = [x_c, y_c, z_c]^T$$

$$p_{uvw} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a \cos(\sigma) \\ b \sin(\sigma) \\ 0 \end{bmatrix},$$

gde je σ parametar koji promjenom u opsegu $\sigma \in [0, 2\pi]$ opisuje cijelu elipsu.

Neka je početak vezanog koordinatnog sistema transliran, a njegove ose rotirane u odnosu na ose referentnog sistema. Tada se veza između dva koordinatna sistema može uspostaviti preko matrice homogene transformacije između koordinatnih sistema $O-xyz$ i $O-uvw$ ili, što je isto, u razvijenoj formi

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

A \mathbf{R} je matrica rotacije koja izražava vezu između orijentacije osa vezanog i referentnog koordinatnog sistema. Primijetimo da su vektor \mathbf{p}_c i matrica \mathbf{R} konstantni.

Parametarska jednačina elipse u prostoru sada se može izraziti kao

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_c + r_{11}a \cos(\sigma) + r_{12}b \sin(\sigma) \\ y_c + r_{21}a \cos(\sigma) + r_{22}b \sin(\sigma) \\ z_c + r_{31}a \cos(\sigma) + r_{32}b \sin(\sigma) \end{bmatrix},$$

što zajedno sa profilom promene parametra $\sigma(t)$ određuje i trajektoriju u prostoru.

U ovom slučaju parametar σ ima geometrijski smisao koji se ne može lako povezati sa krivolinijskom koordinatom s . Da bismo se u to uvjerali, potražimo diferencijalne priraštaje koordinata duž putanje, i iskoristimo izraz za priraštaj krivolinijske koordinate

Pri tome, imamo u vidu da su koeficijenti r_{ij} elementi rotacione matrice. Za rotacionu matricu \mathbf{R} znamo da su njene kolone, odnosno vrste zapravo jedinični vektori. Takođe važi i da su kolone matrice uzajamno ortogonalne, a ortogonalnost važi i za vrste. Primjenom ovih osobina dobija se izraz

$$ds = \sqrt{a^2 \sin^2(\sigma) + b^2 \cos^2(\sigma)} \cdot d\sigma$$

Iz dobijenog izraza vidimo da postoji složena veza između geometrijskog parametra σ i krivolinijske koordinate s koja je određena eliptičkim integralom, koji se može riješiti samo numerički.

Na ovom primjeru možemo uočiti da je parametrizaciju putanje gotovo uvijek moguće izvršiti nekim intuitivnim parametrom, a njegova veza sa krivolinijskom koordinatom može biti i veoma složena.

Primijetimo još, da ako bi kretanje bilo po kružnici, a ne elipsi, tada bi za $a=b=r$ (gdje je r označen poluprečnik kružnice) priraštaj krivolinijske koordinate postao

$$ds = \sqrt{r^2 \cos^2(\sigma) + r^2 \sin^2(\sigma)} \cdot d\sigma = r \cdot d\sigma$$

pa bi geometrijski parametar imao prirodu pređenog puta.

Na način sličan prethodim primjerima, mogli bismo da parametrizujemo praktično bilo koju prostornu putanju, a zatim usvajanjem profila promjene parametra, da riješimo i problem sinteze trajektorije, odnosno pozicije vrha robota u prostoru. Da bi smo potpuno definisali kretanje vrha robota, preostaje nam još da razmotrimo način zadavanja orijentacije vrha robota u prostoru.

Definisanje orijentacije vrha robota u prostoru

Orijentacija vrha robota u prostoru u potpunosti je određena sa tri nezavisna parametra, videli smo da postoje različiti načini za definisanje orijentacije, na primjer, preko Ojlerovih uglova ZYZ, odnosno ZYX tipa, ili preko uglova skretanja, propinjanja i valjanja. Prema tome, u opštem slučaju vektor koji definiše orijentaciju vrha robota možemo zapisati u obliku

$$o = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} \quad \text{bez obzira na prirodu uglova } \varphi, \theta \text{ i } \psi.$$

Da bi se definisala orijentacija vrha robota u zadatku duž trajektorije, potrebno je zapravo odrediti vrijednosti uglova orijentacije tokom vremena. Kao i u slučaju pozicije, to se može izvesti u dva koraka, tako što se najprije odredi parametrizovana reprezentacija uglova orijentacije, a zatim se parametru dodijeli vremenski profil promjene.

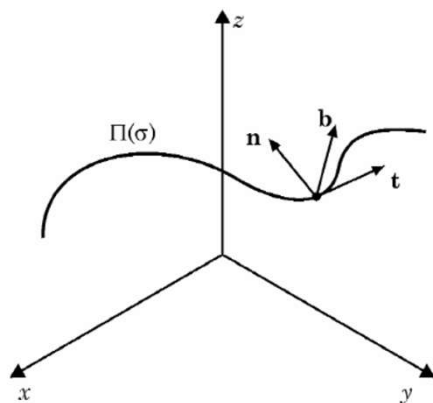
U jednostavnijim slučajevima planiranja orijentacije vrha robota, od interesa je samo orijentacija vrha u početnoj, odnosno krajnjoj tački trajektorije, dok orijentacija duž trajektorije nije od interesa. Tada možemo pristupiti interpolaciji uglova orijentacije na sledeći način

$$o = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \varphi_i + (\varphi_f - \varphi_i)\sigma \\ \theta_i + (\theta_f - \theta_i)\sigma \\ \psi_i + (\psi_f - \psi_i)\sigma \end{bmatrix},$$

gde smo indeksom i označili vrijednosti uglova orijentacije u početnoj, a indeksom f u završnoj tački putanje. Parametar σ mijenja se u intervalu $[0,1]$, i definisanjem njegovog vremenskog profila, zadatak sinteze orijentacije vrha robota duž trajektorije je riješen. Kao i ranije, za profil promene parametra σ možemo usvojiti neki od već proučenih profila.

Napomenim da je ovakav način planiranja orijentacije vrha robota pogodniji za primjene kod PTP kretanja. Ako pozicija vrha robota treba da prati neku putanju u prostoru, onda je vjerovatnije da orijentacija vrha robota treba da bude definisana samom trajektorijom. Na primjer, kod elektrolučnog zavarivanja, putanja vrha treba da prati šav koji se vari, a pri tome želimo da elektroda bude svo vrijeme upravna na šav. Slično je i kod robotskog farbanja, gdje putanja vrha zavisi od površine koja se farba, a orijentacija vrha treba da je ortogonalna u odnosu na tu površinu.

Prema tome, naš zadatak je da sada, na osnovu parametrizovane putanje vrha robota u prostoru, odredimo informacije o orijentaciji. Prisjetimo se da se u diferencijalnoj geometriji svakoj tački prostorne krive mogu pridružiti tri ortogonalna jedinična vektora kao na Sl. 5.9.



Sl. 5.9. Prostorna kriva i prirodni triedar u jednoj tački krive

Tri vektora sa Sl. 5.9, vektor tangente t , vektor normale n i vektor binormale b formiraju akozvani prirodni triedar krive, odnosno koordinatni sistem desne orijentacije. Vektor tangente definiše pravac tangente u datoj tački prostorne krive. Vektor normale ortogonalan je na pravac tangente, a usmjeren je ka centru krivine krive. Konačno, vektor binormale ortogonalan je na oba prethodna vektora.

Ako je prostorna kriva zadata parametarski tako da je parametar krivolinijska koordinata s

$$p = p(s)$$

tada se jedinični vektori prirodnog triedra mogu odrediti na sledeći način

$$t = p' = \frac{dp}{ds},$$

$$n = \frac{p''}{|p''|},$$

$$b = t \times n = \frac{p' \times p''}{|p' \times p''|},$$

pri čemu se diferenciranje vrši po krivolinijskoj koordinati, oznaka \times predstavlja vektorski proizvod dva vektora, a $|*|$ predstavlja oznaku za moduo vektora.

Ako je prostorna kriva zadata po nekom drugom, geometrijskom parametru σ , tada izrazi za jedinične vektore prirodnog triedra postaju nešto složeniji:

$$t = \frac{p'(\sigma)}{|p'(\sigma)|},$$

$$n = \frac{(p'(\sigma) \times p''(\sigma)) \times p'(\sigma)}{|(p'(\sigma) \times p''(\sigma)) \times p'(\sigma)|},$$

$$b = \frac{p'(\sigma) \times p''(\sigma)}{|p'(\sigma) \times p''(\sigma)|},$$

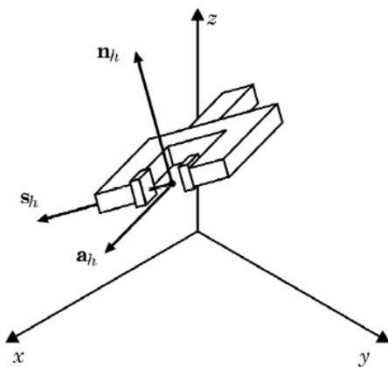
pri čemu se diferenciranje sada vrši po parametru σ .

Jedinične vektore prirodnog triedra možemo iskoristiti za zadavanje orijentacije vrha robota na sledeći način. Podesjetimo se najprije, da smo kod razmatranja kinematike robota uveli koncept matrice homogene transformacije, kojom možemo izraziti položaj i orijentaciju vrha robota, kao

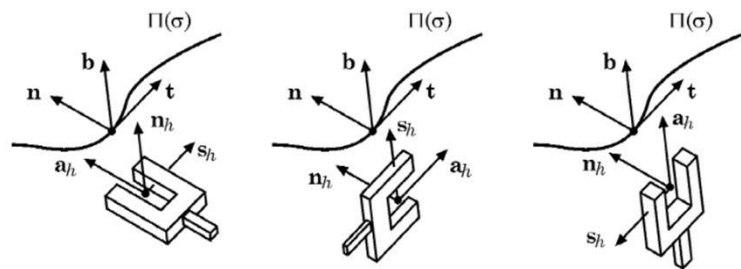
$$T = \left[\begin{array}{ccc|c} n_h & s_h & a_h & p_h \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

gde vektori \mathbf{n}_h , \mathbf{s}_h i \mathbf{a}_h predstavljaju projekcije jediničnih vektora osa koordinatnog sistema vezanog za hvataljku na ose referentnog sistema, a vektor \mathbf{p}_h označava poziciju vrha robota.

Jedinični vektori koordinatnog sistema hvataljke prikazani su na Sl. 5.10.



Sl. 5.10. Jedinični vektori koordinatnog sistema hvataljke



Sl. 5.11. Različite orijentacije hvataljke duž putanje u prostoru

Zadavanje orijentacije hvataljke svodi se onda na uspostavljanje veze između osa koordinatnog sistema hvataljke i prirodnog triedra putanje. Neke od različitih mogućnosti za orijentaciju hvataljke u odnosu na putanju možemo vidjeti na Sl. 5.11.

Na slici su prikazani samo slučajevi kada su ose vezanog koordinatnog sistema hvataljke paralelne osama prirodnog triedra u posmatranoj tački putanje. U opštem slučaju, veza između koordinatnog sistema hvataljke i prirodnog triedra zavisi kako od prirode zadatka, tako i od oblika hvataljke, odnosno alata koji robot nosi, odnosno, od načina na koji je definisan koordinatni sistem hvataljke. Ta veza može biti predstavljena kao

$$[n_h \mid s_h \mid a_h] = [t \mid n \mid b] \cdot R_h$$

pri čemu je R_h konstantna matrica rotacije. Na primjer, za orijentacije hvataljke sa Sl. 5.11, imamo, s lijeva na desno

$$R_h = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad R_h = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad R_h = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

odnosno, matrice R_h definišu permutovanje jediničnih vektora prirodnog triedra, uz eventualnu promjenu smjera vektora, kao na primjer vektora t u posljednjem izrazu. U opštem slučaju, elementi matrice R_h uspostavljaju vezu između osa dva koordinatna sistema.

Imajući u vidu da se jedinični vektori prirodnog triedra mogu odrediti iz izraza:

$$\begin{aligned}
 t &= p' = \frac{dp}{ds}, & t &= \frac{p'(\sigma)}{|p'(\sigma)|}, \\
 n &= \frac{p''}{|p''|}, & n &= \frac{(p'(\sigma) \times p''(\sigma)) \times p'(\sigma)}{|(p'(\sigma) \times p''(\sigma)) \times p'(\sigma)|}, \\
 b &= t \times n = \frac{p' \times p''}{|p' \times p''|}, & b &= \frac{p'(\sigma) \times p''(\sigma)}{|p'(\sigma) \times p''(\sigma)|},
 \end{aligned}$$

odnosno,

u potpunosti je određena orijentacija vrha hvataljke duž putanje. Kada se parametru pridruži i vremenski profil, pitanje orijentacije hvataljke duž trajektorije u potpunosti je riješeno.

Ilustrujmo opisani koncept definisanja orijentacije vrha robota duž putanje uz pomoć primjera. Za početak, razmotrićemo jednostavnu putanju oblika kružnice u prostoru. Neka je kružnica zadata parametarskim jednačinama

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ y_c + r \cos(\sigma) \\ z_c + r \sin(\sigma) \end{bmatrix},$$

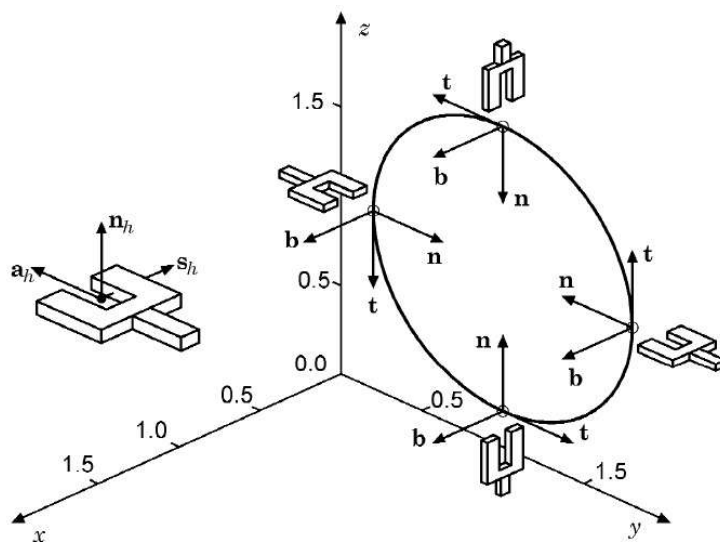
iz kojih zaključujemo da kružnica leži u yz ravni, njen centar nalazi se u tački $[0, y_c, z_c]^T$, a poluprečnik iznosi r . Parametar σ ima prirodu ugla, i svojom promjenom na intervalu $\sigma \in [0, 2\pi]$ definiše cijelu kružnicu. Primijetimo još da je, usled periodičnosti trigonometrijskih funkcija, putanja zatvorena. Da bismo odredili jednačine jediničnih osa prirodnog triedra, diferencirajmo najprije parametarsku jednačinu dva puta, čime se dobija

$$p' = \begin{bmatrix} 0 \\ -r \sin(\sigma) \\ r \cos(\sigma) \end{bmatrix}, \quad |p'| = r \quad p'' = \begin{bmatrix} 0 \\ -r \cos(\sigma) \\ -r \sin(\sigma) \end{bmatrix}, \quad |p''| = r$$

Smjenom izraza u relacije na prethodnom slajdu možemo odrediti jednačinu jediničnog vektora tangente, binormale i normale

$$t = [0 \quad -\sin(\sigma) \quad \cos(\sigma)]^T \quad b = [1 \quad 0 \quad 0]^T \quad n = \frac{p''}{|p''|} = [0 \quad -\cos(\sigma) \quad -\sin(\sigma)]^T$$

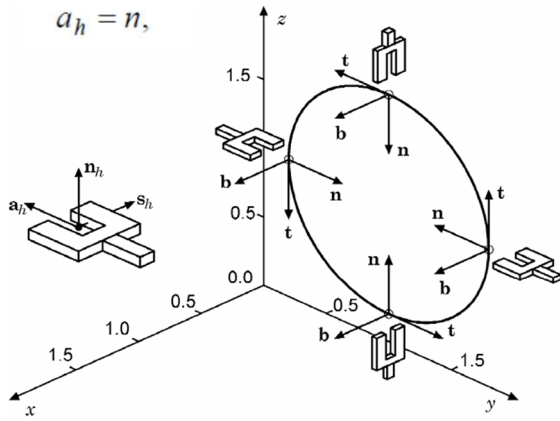
Vidimo da je jedinični vektor binormale konstantan, i paralelan x osi, što je potpuno očekivano, imajući u vidu da je kružnica ravanska kriva koja leži u yz ravni. U ovom primjeru, zbog kretanja po kružnici, vektor normale ima isti pravac kao i vektor p'' ,



Sl. 5.12. Putanja u obliku kružnice i orijentacija vrha robota duž putanje

Izgled kružnice u prostoru, kao i orijentacije prirodnih triedara u nekoliko tačaka na kružnici prikazane su na Sl. 5.12. Kružnica je konstruisana za vrijednosti parametara $r = 0.8$, $x_0 = 0$, $y_0 = 1$, i $z_0 = 1$, a prirodni triedri prikazani su u tačkama definisanim sa $\sigma \in [0, \pi/2, \pi, 3\pi/2]$. Na slici je takođe definisan koordinatni sistem vezan za vrh robota, kao i željena orijentacija hvataljke u zadatku. Sa Sl. 5.12 možemo uočiti vezu između orijentacije koordinatnog sistema hvataljke i prirodnog triedra

$$\begin{aligned}n_h &= t, \\s_h &= -b, \\a_h &= n,\end{aligned}$$



Sl. 5.12. Putanja u obliku kružnice i orijentacija vrha robota duž putanje

koja se može izraziti i preko matrice rotacije R_h , definisane jednačinom

$$\begin{bmatrix} n_h & | & s_h & | & a_h \end{bmatrix} = \begin{bmatrix} t & | & n & | & b \end{bmatrix} \cdot R_h$$

pri čemu je

$$R_h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

odnosno

$$R = \begin{bmatrix} 0 & -1 & 0 \\ -\sin(\sigma) & 0 & -\cos(\sigma) \\ \cos(\sigma) & 0 & -\sin(\sigma) \end{bmatrix}$$

Izborom vremenske zavisnosti $\sigma = \sigma(t)$, odnosno profila kretanja, u potpunosti je izrazom

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ y_c + r \cos(\sigma) \\ z_c + r \sin(\sigma) \end{bmatrix},$$

određena i trajektorija u prostoru, dok je orijentacija vrha robota duž trajektorije određena izrazom

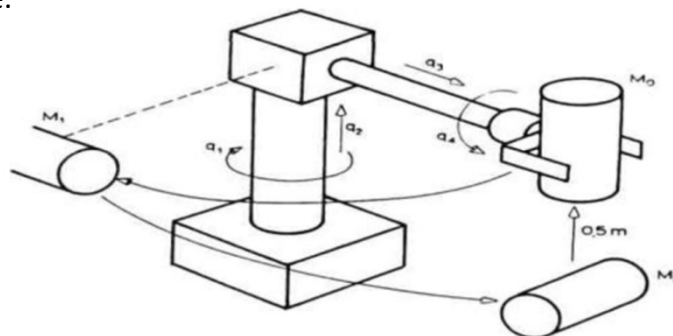
$$R = \begin{bmatrix} 0 & -1 & 0 \\ -\sin(\sigma) & 0 & -\cos(\sigma) \\ \cos(\sigma) & 0 & -\sin(\sigma) \end{bmatrix}$$

U NASTAVKU

Shema postupka za numeričko rešavanje inverznog problema kinematike robota

Iz knjige ROBOTIKA, Prof. dr. Veljko Potkonjak

U nekim prostijim manipulacionim zadacima direktno se zadaju pomjeranja u zglobovima. Primer za to bio bi robot na slici. Zadatak je prikazan shematski na istoj slici. Radni predmet treba prenijeti prvo iz tačke M_0 u tačku M_1 . Pri ovakvom kretanju mijenja se koordinata q_1 za $\pi/2$. Takođe se mijenja koordinata q_4 koja vrši obrtanje radnog predmeta za $\pi/2$. Pri kretanju od M_1 do M_2 mijenja se opet q_1 unatrag za $\pi/2$ i istovremeno se mijenja q_2 spuštajući predmet za 0,5 m. Konačno, pri kretanju od M_2 do početnog položaja M_0 opet se mijenja q_2 ovog puta naviše za 0,5 m i mijenja se q_4 koja obrće predmet unatrag za $\pi/2$ vraćajući ga tako na polazni položaj. Ovakvo zadavanje zadatka pogodno je samo za jednostavne manipulacione zadatke.



U složenijim zadacima uvijek je neophodno ostvariti određeno kretanje hvataljke koja će pri tome izvršiti postavljeni zadatak. Jasno je da je nemoguće unaprijed predvidjeti pomjeranja u zglobovima (tj. $q(t)$) koja bi ostvarila željeno kretanje hvataljke. Zato takve zadatke određujemo preko vektora spoljašnjeg položaja hvataljke (vektor X). Zadaćemo kretanje vrha tako što zadajemo zakon vremenske promene koordinata $x_A(t)$, $y_A(t)$ i $z_A(t)$. Zakon promjene orijentacije zadajemo tako što zadamo vremensku promjenu uglova $\theta(t)$, $\phi(t)$ i $\psi(t)$. U slučaju robota sa pet stepeni slobode $\psi(t)$ je zavisno i ne zadaje se.

Zadali smo manipulacioni zadatak preko vremenske promene vektora $X(t)$. Sada ćemo pokazati kako se može izračunati unutrašnje kretanje $q(t)$ ako je spoljašnje $X(t)$ zadato. Zakon $X(t)$ određuje i $\dot{X}(t)$ i $\ddot{X}(t)$. Sada bi na osnovu X trebalo odrediti q , na osnovu \dot{X} odrediti \dot{q} i konačno, na osnovu \ddot{X} odrediti \ddot{q} . Međutim ovo nije lako realizovati. Već smo ranije rekli da je računanje q iz poznatog X veoma složen posao, često bez jednoznačnog rešenja. Dalje, računanje \dot{q} pomoću relacije $\dot{q} = J^{-1} \dot{X}$ zahtijeva prethodno sračunavanje Jakobijana, a njega ne možemo izračunati jer ne znamo q . Isto vazi i za pokušaj računanja \ddot{q} .

Ovaj složeni problem izračunavanja unutrašnjeg kretanja $q(t)$ iz poznatog spoljašnjeg kretanja $X(t)$ naziva se inverzni problem kinematike robota. Sa nekoliko riječi ćemo objasniti zašto je rješavanje ovog problema važno. Sa jedne strane, zaključili smo da je manipulacioni zadatak pogodno zadati u obliku kretanja hvataljke po zakonu $X(t)$. Sa druge strane robot se pokreće pomoću motora koji deluju u zglobovima i koji izazivaju pomjeranja u zglobovima. Prema tome, za djelovanje motora bitna je promjena koordinata u zglobovima tj. $q_1(t)$, $q_2(t)$, ..., $q_n(t)$. Da bi se izvršila sinteza upravljanja kretanjem robota neophodno je sračunati unutrašnje kretanje $q(t)$ odnosno riješiti inverzni problem kinematike.

Sada ćemo objasniti jedan praktičan postupak za numeričko rješavanje inverznog problema kinematike. Treba naglasiti da je postupak namijenjen za korišćenje na računaru.

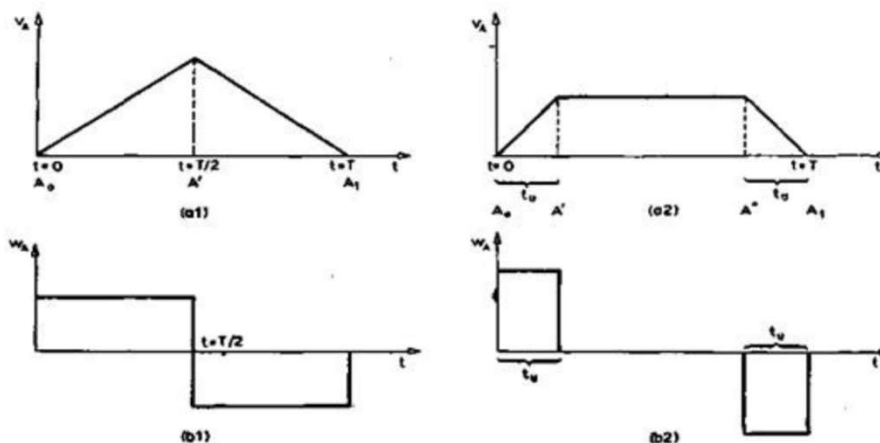
Prije nego što pređemo na objašnjenje samoga postupka uvešćemo pojam stanja mehaničkog sistema. U strogu definiciju pojma stanja u smislu teorije sistema ovdje nećemo ulaziti. Zadovoljićemo se time da kažemo da je stanje mehaničkog sistema u nekom trenutku određeno poznavanjem položaja \mathbf{q} i brzina $\dot{\mathbf{q}}$. Ove dvije veličine (u stvari dvije matrice \mathbf{q} i $\dot{\mathbf{q}}$) definišu stanje zato što su položaj i brzine one veličine koje se ne mogu trenutno promijeniti, već se do narednog beskonačno bliskog trenutka vremena mogu promijeniti samo beskonačno malo. Za razliku od njih ubrzanje se može skokovito mijenjati u trenucima početka ili prestanka dejstva sile. Na ovaj način, položaj i brzine se javljaju kao neophodni početni uslovi za određivanje daljeg kretanja sistema. Zato ove veličine uzete zajedno nazivamo stanjem sistema. Pod stanjem podrazumevamo par kolona matrica, \mathbf{q} i $\dot{\mathbf{q}}$.

Analizirajmo sada početno stanje sistema koji ovdje posmatramo, a to je kinematički lanac robota. Početno stanje određeno je vrednostima \mathbf{q} i $\dot{\mathbf{q}}$ u početnom trenutku vremena t_0 , dakle $\mathbf{q}(t_0)$ i $\dot{\mathbf{q}}(t_0)$. U slučaju neredundantnih i nesingularnih mehanizama početno stanje može se definisati i preko položaja hvataljke i njene brzine u početnom trenutku t_0 , dakle $\mathbf{X}(t_0)$ i $\dot{\mathbf{X}}(t_0)$. Ranije smo već napomenuli da ćemo zadatak zadati preko kretanja hvataljke $\mathbf{X}(t)$. Na taj način biće određeno i početno stanje $\mathbf{X}(t_0)$, $\dot{\mathbf{X}}(t_0)$. Postupak koju ćemo predložiti za rješavanje inverznog problema kinematike zahtijeva poznavanje početnog stanja u obliku $\mathbf{q}(t_0)$, $\dot{\mathbf{q}}(t_0)$. Za taj početni trenutak morao bi se sprovesti ranije spominjani složeni proračun unutrašnjih koordinata na osnovu poznatog položaja hvataljke \mathbf{X} . Srećom, videćemo da se takav proračun obavlja samo jednom i to za početni trenutak vremena. Za kasnije trenutke vremena to neće biti potrebno.

Treba još istaći da je, bez obzira na zadavanje zadataka preko kretanja hvataljke, često poznato početno stanje u obliku $\mathbf{q}(t_0)$, $\dot{\mathbf{q}}(t_0)$. To je otuda što robot obično kreće iz nekog svog uobičajenog polaznog položaja u kome su poznati svi položaji zglobova tj. sve koordinate \mathbf{q} . Početne brzine su tada obično jednake nuli jer robot kreće iz mirovanja.

Sledeće što ćemo analizirati biće način numeričkog zadavanja vremenskih funkcija. Umjesto zadavanja nekim analitičkim izrazom, što u složenijim slučajevima nije ni moguće, funkciju vremena zadajemo u vidu niza tačaka, odnosno nizom njenih vrijednosti koje odgovaraju nizu trenutaka vremena. Primijenimo ovo na slučaj zadavanja vremenske promjene položaja hvataljke $X(t)$. Sada bi za svaku komponentu kolona matrice X zadali niz vrijednosti koje bi definisale promjenu te komponente tokom vremena. Problem nastaje onda kada treba odrediti izvode $\dot{X}(t)$ i $\ddot{X}(t)$ i kada bi trebalo izvršiti numeričko diferenciranje. Kako je numeričko diferenciranje nepoželjan zadatak u svakom proračunu, to ćemo sada izložiti postupak kojim će se ovaj problem izbjeći.

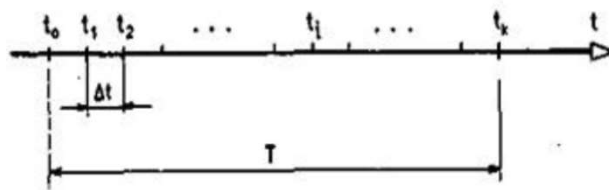
Poći ćemo od činjenice da početno stanje $X(t_0)$, $\dot{X}(t_0)$ i vremenska promjena ubrzanja $\ddot{X}(t)$ jednoznačno određuju kretanje $X(t)$. Zato ćemo manipulacioni zadatak i zadati preko određenog početnog stanja i vremenske promjene ubrzanja. Pokažimo sada na dva primera da je zadavanje kretanja preko ubrzanja veoma pogodno. Zamislimo da želimo pravolinijsko kretanje vrha robota između dve tačke uz trougaoni profil brzine (slika 2.46a) tj. do polovine putanje (deo A_0A_1) vrh ubrzava, a od polovine (deo A_1A_2) usporava tako da se u tački A_1 ponovo zaustavi. Profil ubrzanja tada je prikazan na (slici 2.46b) i može se lako zadati:



Sl. 2.46 - Trougaoni i trapezni profil brzine

odnosno do polovine vremena izvršenja zadatka ubrzanje je konstantno i iznosi $+a$, a od polovine konstantno i negativno i iznosi $-a$. U slučaju trapeznog profila brzine (sl.2.46a2) razlikujemo period ubrzavanja (dio A_0A_1), period konstantne brzine (dio $A'A''$) i period usporavanja (dio $A''A_1$). Gdje je T ukupno vrijeme, a t_u vrijeme ubrzavanja, odnosno usporavanja. Treba reći da ova dva profila brzine spadaju u najčešće profile u praktičnim manipulacionim zadacima. To je važno jer vidimo da se upravo u tim najčešćim slučajevima ubrzanje zadaje veoma jednostavno.

Kako smo već rekli da vrijeme i funkcije vremena posmatramo diskretno tj. u nizu trenutaka, to ćemo vremenski interval T u kome se zadatak izvršava podijeliti na k podintervala Δt uvodeći vremenske trenutke t_0, t_1, \dots, t_k (sl. 2.47)



Sl. 2.47 - Diskretizacija intervala kretanja

Sada ćemo relacije za ubrzanje (2.57) i (2.58) tretirati diskretno tj. zadati nizove vrednosti koje odgovaraju trenucima t_0, t_1, \dots, t_k

$$\ddot{x}_A(t) = \begin{cases} +a_x, & t < T/2 \\ -a_x, & t > T/2 \end{cases}$$

$$\ddot{x}_A(t) = \begin{cases} +a_x, & t < t_u \\ 0, & t_u < t < T - t_u \\ -a_x, & t > T - t_u \end{cases}$$

Manipulacioni zadatak smatramo zadatim preko zadatog početnog stanja $q(t_0)$, $\dot{q}(t_0)$ i poznate vremenske promene ubrzanja hvataljke $\ddot{X}(t)$, Sada ćemo objasniti postupak za rešavanje inverznog problema kinematike tj.za sračunavanje unutrašnjeg kretanja $q(t)$. Postupak počinje sa početnim trenutkom t_0 . Kako je $q(t_0)$ i $\dot{q}(t_0)$ poznato, te za taj trenutak možemo sračunati Jakobijan $J(q)$ i pridruženu matricu $A(q, \dot{q})$. S obzirom na to da je $\ddot{X}(t_0)$ poznato, sada iz relacije (2.56) izračunavamo vrednost $\ddot{q}(t_0)$ u posmatranom trenutku t_0 :

$$\ddot{q} = j^{-1}(\ddot{X} - A)$$

Posmatrajmo sada subinterval između trenutka t_0 i t_1 . Za kratak subinterval Δt možemo smatrati da se q na tom subintervalu neće promijeniti već će zadržati vrijednost izračunatu za t_0 . Tada možemo tokom Δt , promjenu q posmatrati kao jednako ubrzano kretanje i formirati izraze kojima ćemo izračunati stanje $q(t_1)$, $\dot{q}(t_1)$ u trenutku t_1 :

$$q(t_1) = q(t_0) + \dot{q}(t_0)\Delta t + \frac{1}{2}\ddot{q}(t_0)\Delta t^2$$

$$\dot{q}(t_1) = \dot{q}(t_0) + \ddot{q}(t_0)\Delta t$$

Sada, kada znamo stanje $q(t_1)$, $\dot{q}(t_1)$ u trenutku t_1 počinjemo novi ciklus ponavljajući za trenutak t_1 cio račun koji smo sprovodili za t_0 . Tako, rekurzivno izračunavamo stanje q , \dot{q} u trenucima t_2, t_3, \dots, t_k .

Kao rezultat opisanog postupka dobijamo nizove vrednosti q i \dot{q} koji odgovaraju nizu trenutaka vremena, odnosno dobijamo zakon kretanja u unutrašnjim koordinatama $q(t)$ i $\dot{q}(t)$. Ovim smo riješili inverzni problem kinematike robota.

$$X = \eta(q)$$

$$\dot{X} = \frac{\partial \eta}{\partial q} \dot{q}$$

$$\dot{X} = J(q)\dot{q}$$

$$\ddot{X} = \frac{\partial \eta}{\partial q} \ddot{q} + \frac{\partial^2 \eta}{\partial q^2} \dot{q}^2$$

$$\ddot{X} = J(q)\ddot{q} + A(q, \dot{q})$$

Matricu $J = \partial \eta / \partial q$ dimenzije $n \times n$ nazivamo Jakobijan, a matricu dimenzije $n \times 1$ zvaćemo pridružena matrica. Jakobijeve forme služe nam za preračunavanje brzina i ubrzanja.

Shema postupka za numeričko rešavanje inverznog problema kinematike robota

