

Univerza v Ljubljani  
Fakulteta za elektrotehniko



# KINEMATIKA IN DINAMIKA ROBOTOV

---

DIFERENCIJALNA KINEMATIKA,  
STATIKA, GENERIRANJE TRAJEKTORIJE,  
DINAMIKA

*Marko Munih*

LJUBLJANA, SEPTEMBER 2022



# PREDGOVOR

Gradnja pa tudi ustrezna uporaba robotov zahteva znanje z več področij, predvsem fizike oz. mehanike, strojništva, elektrotehnike in računalništva. Specifična znanja zajemajo vsaj kinematiko, diferencialno kinematiko, statiko, dinamiko, vodenje odprtih in zaprtih kinematičnih verig pri stiku robota z okolico, prijemala ter še ožja znanja, npr. mobilne in paralelne robe.

Že pri predmetu Osnove robotike študentje spoznajo direktno in inverzno kinematiko, kot tudi manjše izseke drugih poglavij. To je potrebno predznanje za študij te knjige, skupaj s poznavanjem matematike in fizike iz prvih dveh letnikov. Pri predmetu Kinematika in dinamika robotov v prvem letniku magistrskega študija nadaljujemo z homogenimi transformacijami majhnih in diferencialnih premikov in spoznamo Jacobijevo matriko za manipulator, ki ima svojo uporabo tudi v statiki. Za tem sledi poglavje generiranja trajektorij. Kot naslednje je smiselno obvladati vsebine dinamike, najprej po Langrangeovi in zatem še Newton-Eulerjevi metodi, kar je vse predpogoj za vodenje robotov.

Besedilo pred nami je rezultat razvoja podajanja snovi, predavanj, brisanja in dodajanja podnaslovov skozi leta na Fakulteti za elektrotehniko, UL in je temu še naprej podvrženo. Osnovni okvir od diferencialne kinematike do statike sega v predavanja, kot jih je podajal akademik prof. Alojz Kralj, ki je predaval to snov pred mano. Ključno vlogo pri nastanku je kot kritičen bralec imel še akademik prof. Tadej Bajd.

Vektorji so zapisani z majhnimi poudarjenimi in matrike z velikimi poudarjenimi tiskanimi črkami. Kljub pazljivemu pregledu se zavedam, da vsebinske in tipopisne napake ter druge nedoslednosti niso izključene, za kar se bralcu že vnaprej opravičujem.

Marko Munih

Ljubljana, 28. septembra 2022



# Kazalo

<b>1 HOMOGENE TRANSFORMACIJE MAJHNIH ALI DIFERENCIJALNIH POMIKOV</b>	<b>9</b>
1.1 ODVOD TRANSFORMACIJE . . . . .	10
1.2 DIFERENCIJALNA TRANSLACIJA IN ROTACIJA . . . . .	11
1.3 EKVIVALENT MED DIFERENCIJALNIMI ROTACIJAMI . .	13
1.4 TRANSFORMACIJA DIFERENCIJALNIH POMIKOV MED KOORDINATNIMI SISTEMI . . . . .	18
1.5 DIFERENCIJALNE RELACIJE V TRANSFORMACIJSKIH IZRAZIH . . . . .	23
<b>2 JACOBIJEVA Matrika za manipulator</b>	<b>31</b>
2.1 IZRAČUN JACOBIJEVE MATRIKE ZA STANFORD M- NIPULATOR . . . . .	34
2.2 GEOMETRIJSKA IN ANALITIČNA JACOBIJEVA MATRIKA	43
2.3 IZRAČUN JAKOBIJEVE MATRIKE . . . . .	44
2.3.1 Kotna hitrost . . . . .	45
2.3.2 Translacijska hitrost . . . . .	46
2.3.3 Translacijski sklepi . . . . .	47
2.3.4 Rotacijski sklepi . . . . .	47
2.3.5 Kombinacija Jakobi členov za translacijsko in kotno hitrost . . . . .	48
2.3.6 Primer: 2DOF, 2R ravninski manipulator . . . . .	49
2.3.7 Primer: Jakobijeva matrika za poljubno točko na se- gmentu . . . . .	50
2.3.8 Primer: Stanford manipulator . . . . .	51
2.3.9 Primer: SCARA manipulator . . . . .	53
2.4 HITROST ORODJA . . . . .	54
2.5 ANALITIČNA JAKOBIJEVA MATRIKA . . . . .	56
2.6 SINGULARNOSTI . . . . .	57
2.6.1 Ločitev singularnosti . . . . .	59
2.6.2 Singularnosti zapestja . . . . .	60

2.6.3	Singularnosti roke . . . . .	60
2.6.4	Singularnosti za manipulator v obliki komolca . . . . .	61
2.6.5	Sferični manipulator . . . . .	63
2.6.6	SCARA manipulator . . . . .	63
2.7	INVERZNA JACOBIJEVA MATRIKA . . . . .	64
2.7.1	Ponovitev . . . . .	66
2.8	REDUNDANTNOST IN VODENJE PREKO IZRAČUNANE HITROSTI . . . . .	69
2.9	OPTIMALNA REŠITEV, PSEVDOINVERZNA J MATRIKA	73
<b>3</b>	<b>STATIKA</b>	<b>77</b>
3.1	EKVIVALENTNI MOMENTI SKLEPA . . . . .	77
3.2	DUALNOST KINEMATIKE IN STATIKE . . . . .	81
3.3	TRANSFORMACIJA SIL IN MOMENTOV . . . . .	85
3.4	TOGOST . . . . .	88
<b>4</b>	<b>GENERIRANJE TRAJEKTORIJE</b>	<b>95</b>
4.1	INTERPOLACIJA . . . . .	95
4.1.1	Pomen interpolacije . . . . .	96
4.1.2	Izbira interpolacije . . . . .	100
4.1.3	Absolutni interpolator . . . . .	102
4.1.4	Inkrementalni interpolator . . . . .	103
4.2	APROKSIMACIJA z OSNOVNIMI FUNKCIJAMI . . . . .	106
4.2.1	Polinomaska aproksimacija z kubičnim polinomom za interpolacijo poti z vmesno točko . . . . .	109
4.2.2	Polinomi višjega reda . . . . .	111
4.2.3	Linearni segmenti s paraboličnimi nastavki . . . . .	112
4.2.4	Vmesne točke in premične povezave . . . . .	117
4.2.5	Trajektorija podana v zunanjih koordinatah(WCS) . .	122
4.2.6	Kakšno interplacijsko razdaljo potrebujemo za želeno točnost . . . . .	123
4.2.7	Bezier krivulje . . . . .	124
4.2.8	Lokalno utežena regresija (LWR) . . . . .	134
4.2.9	Gaussova mešana regresija (GMR) . . . . .	138
4.2.10	Gaussian Process Regression (GPR) . . . . .	141
4.3	DINAMIČNI PRIMITIVI GIBANJA . . . . .	146
4.3.1	Dinamični primitivi gibanja (DMP) . . . . .	146
4.3.2	Učenje DMP parametrov iz ene same demonstracije .	152
4.3.3	DMP v kartezičnem prostoru . . . . .	157
4.3.4	DMP tretjega reda . . . . .	157
4.3.5	Moduliranje DMPjev . . . . .	158

4.3.6	Compliant Movement Primitives (CMP) . . . . .	161
4.3.7	Pobabilistic Movement Primitives (PROMP) . . . . .	167
4.3.8	Kernelized Movement Primitives (KMP) . . . . .	167
4.3.9	Gaussian Mixture Models (GMM) . . . . .	167
<b>5</b>	<b>LAGRANGEVA DINAMIKA</b>	<b>173</b>
5.1	LAGRANGEVA FORMULACIJA . . . . .	173
5.1.1	Izračun kinetične energije . . . . .	175
5.1.2	Izračun potencialne energije . . . . .	182
5.1.3	Enačbe gibanja . . . . .	183
5.2	POMEMBNE LASTNOSTI DINAMIČNEGA MODELA . . .	185
5.3	POŠEVNO-SIMETRIČNA MATRIKA $\dot{\mathbf{B}} - 2\mathbf{C}$ . . . . .	186
5.4	LINEARNOST GLEDE NA DINAMIČNE PARAMETRE . .	188
5.5	DINAMIČNI MODEL V ZUNANJIH KOORDINATAH . . .	194
<b>6</b>	<b>NEWTON-EULERJEVA DINAMIKA</b>	<b>199</b>
6.1	UVOD . . . . .	199
6.2	LASTNOSTI ANTISIMETRIČNIH MATRIK . . . . .	200
6.3	IZPELJAVA NEWTON-EULERJEVE FORMULACIJE . .	203
6.4	IZRAČUN KINEMATIČNIH VELIČIN . . . . .	208
6.4.1	Rekurzija naprej: prvi segment . . . . .	213
6.4.2	Rekurzija naprej: drugi segment . . . . .	214
6.4.3	Rekurzija nazaj: drugi segment . . . . .	216
6.4.4	Rekurzija nazaj: prvi segment . . . . .	217
	<b>Literatura</b>	<b>221</b>



# Poglavlje 1

## HOMOGENE TRANSFORMACIJE MAJHNICH ALI DIFERENCIALNIH POMIKOV

Popisati želimo majhen pomik v legi prijemala, torej vrha robota. Zanimajo nas temu pripadajoči diferencialni pomiki posameznih spremenljivk sklepov manipulatorja. V tej zvezi lahko govorimo o diferencialnih pomikih podanih v zunanjih ( $T_6$ ) koordinatah manipulatorja (glede na vrh) ter kako se le-ti ustrezeno preslikajo v diferencialne pomike notranjih koordinat, to je v spremenljivke sklepov manipulatorja. Tako zastavljene naloge srečamo pri praktičnem delu in tudi tedaj, ko moramo diferencialne pomike v koordinatah senzorja umetnega vida, kamere, preslikati v diferencialne pomike vrha manipulatorja, ki so potrebni, da se predpisani premik izvede. V tem primeru nas zanimajo ustrezeni diferencialni pomiki v prostoru notranjih koordinat manipulatorja (sklepih). Prav tako bo ta odnos med diferencialnimi spremembami zunanjih koordinat in notranjih koordinat manipulatorja pomemben za razumevanje in izračunavanje dinamike manipulatorja, kajti pri izračunavanju hitrosti potrebujemo odvod poti po času, torej bomo opazovali odnos, kako so nastajale diferencialne spremembe v zunanjih in notranjih koordinatah v enoti časa ( $dt$ ), ter bomo tako tvorili odvod poti oz. izračunali hitrosti. Prav tako je razumljivo, da so diferencialni odnosi osnova, ko bomo želeli izračunati potrebne pospeške in preko teh gonalne sile ali momente, potrebne za pogajanje sklepov manipulatorja. Pri kalibriranju robotov, orodij, predmetov in karakterističnih točk prav tako potrebujemo diferencialne premike, tako da korigiramo napake oz. odstopanja od natančno predpisane pozicije in orientacije.

## 1.1 ODVOD TRANSFORMACIJE

Podano imamo transformacijo  $\mathbf{P}$ , ki je funkcija spremenljivke  $\mathbf{q}$ , torej lahko zapišemo

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \\ P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} \end{bmatrix} = [P_{i,j}] \quad (1.1)$$

$i = 1 \dots 4$  vrstica,

$j = 1 \dots 4$  stolpec.

Vsak element  $\mathbf{P}$  je funkcija spremenljivke  $\mathbf{q}$ , torej

$$d\mathbf{P} = \begin{bmatrix} \frac{\partial P_{1,1}}{\partial q} & \frac{\partial P_{1,2}}{\partial q} & \frac{\partial P_{1,3}}{\partial q} & \frac{\partial P_{1,4}}{\partial q} \\ \frac{\partial P_{2,1}}{\partial q} & \frac{\partial P_{2,2}}{\partial q} & \frac{\partial P_{2,3}}{\partial q} & \frac{\partial P_{2,4}}{\partial q} \\ \frac{\partial P_{3,1}}{\partial q} & \frac{\partial P_{3,2}}{\partial q} & \frac{\partial P_{3,3}}{\partial q} & \frac{\partial P_{3,4}}{\partial q} \\ \frac{\partial P_{4,1}}{\partial q} & \frac{\partial P_{4,2}}{\partial q} & \frac{\partial P_{4,3}}{\partial q} & \frac{\partial P_{4,4}}{\partial q} \end{bmatrix} d\mathbf{q}, \quad d\mathbf{P} = \begin{bmatrix} \frac{\partial P_{i,j}}{\partial q} \end{bmatrix} d\mathbf{q}. \quad (1.2)$$

$i = 1 \dots 4$   
 $j = 1 \dots 4$

Podobno lahko napišemo glede na DH zapis matriko  $\mathbf{A}$ , ki naj opisuje  $i$ -ti rotacijski sklep manipulatorja, torej je spremenljivka  $\theta$ . Spomnimo se, da je matrika  $\mathbf{A}$  sestavljena iz štirih transformacij

$$\mathbf{A} = \mathbf{Rot}(\mathbf{z}, \theta) \cdot \mathbf{Trans}(0, 0, d) \cdot \mathbf{Trans}(a, 0, 0) \cdot \mathbf{Rot}(\mathbf{x}, \alpha) \quad [\text{Bajd, 2002}].$$

Pri tem so

- $\theta$  ... spremenljivka, ki predstavlja rotacijo v sklepu, se spreminja
- $a$  ... dolžina segmenta, določena s konstrukcijo, se ne spreminja,
- $\alpha$  ... zvitost, zasuk segmenta, določena s konstrukcijo, se ne spreminja,
- $d$  ... razdalja med sosednjima segmentoma, se spreminja pri translaciji.

Ustrezno matriko  $\mathbf{A}_i$  podajata enačbi (1.3) in (1.4).

$$\mathbf{A}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha & \sin \theta_i \sin \alpha & a \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha & -\cos \theta_i \sin \alpha & a \sin \theta_i \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.3)$$

$$d\mathbf{A}_i = \begin{bmatrix} -\sin \theta_i & -\cos \theta_i \cos \alpha & \cos \theta_i \sin \alpha & -a \sin \theta_i \\ \cos \theta_i & -\sin \theta_i \cos \alpha & \sin \theta_i \sin \alpha & a \cos \theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} d\theta_i. \quad (1.4)$$

## 1.2 DIFERENCIJALNA TRANSLACIJA IN ROTACIJA

Omejimo se samo na rotacije in translacije ter izločimo oz. ne gledamo perspektive oz. skaliranja. Pri translacijah in rotacijah nas zanimajo načini za podajanje diferencialnih transformacij translacij in rotacij. Kot možne variante lahko diferencialne transformacije podamo v koordinatnem sistemu transformacijske matrike  $\mathbf{T}$  ali v kakem drugem koordinatnem sistemu npr. v osnovnem ali referenčnem koordinatnem sistemu prostora. Torej, če je podana transformacija  $\mathbf{T}$ , ki predstavlja nek koordinatni sistem, moremo  $\mathbf{T} + d\mathbf{T}$  v splošnem izraziti kot (1.5) v osnovnem (globalnem) koordinatnem sistemu

$$\mathbf{T} + d\mathbf{T} = \mathbf{Trans}(d_x, d_y, d_z) \cdot \mathbf{Rot}(\mathbf{k}, d\theta) \cdot \mathbf{T}. \quad (1.5)$$

Prva dva premultiplicirana člena predstavljata pravilo diferencialne transformacije. V (1.5) imajo členi naslednji pomen

$\mathbf{Trans}(d_x, d_y, d_z)$  je transformacija translacije za  $d_x, d_y, d_z$  glede na koordinate osnovnega koordinatnega sistema  
 $\mathbf{Rot}(\mathbf{k}, d\theta)$  podaja transformacijo rotacije za  $d\theta$  okoli vektorja  $\mathbf{k}$  v osnovnem koordinatnem sistemu.

Iz enačbe (1.5) izračunamo  $d\mathbf{T}$  v osnovnem koordinatnem sistemu:

$$d\mathbf{T} = [\mathbf{Trans}(d_x, d_y, d_z) \cdot \mathbf{Rot}(\mathbf{k}, d\theta) - \mathbf{I}] \mathbf{T}. \quad (1.6)$$

Podobno postopamo v koordinatnem sistemu segmenta (dualno enačbam (1.5) in (1.6)), ko želimo diferencialne transformacije podati v  $\mathbf{T}$ - lastnem koordinatnem sistemu (k.s. segmenta). Sedaj  $\mathbf{T}$  postmultipliciramo z diferencialno transformacijo v (1.5) in dobimo

$$\mathbf{T} + d\mathbf{T} = \mathbf{T} \cdot \mathbf{Trans}(d_x, d_y, d_z) \cdot \mathbf{Rot}(\mathbf{k}, d\theta). \quad (1.7)$$

Zadnja dva člena predstavljata pravilo diferencialne transformacije. V tej enačbi imata diferencialni transformaciji naslednji pomen:

**$\mathbf{Trans}(d_x, d_y, d_z)$**  transformacija diferencialne translacije, ki je podana glede na koordinatni sistem  $\mathbf{T}$

**$\mathbf{Rot}(\mathbf{k}, d\theta)$**  podaja transformacijo rotacije za  $d\theta$  okoli vektorja  $\mathbf{k}$ , ki je opisan in podan v koordinatnem sistemu  $\mathbf{T}$ .

Iz enačbe (1.7) izračunamo  $\mathbf{T}$  in dobimo enačbo (1.8)

$$d\mathbf{T} = \mathbf{T} [\mathbf{Trans}(d_x, d_y, d_z) \cdot \mathbf{Rot}(\mathbf{k}, d\theta) - \mathbf{I}]. \quad (1.8)$$

Spoznamo, da sta izraza v oglatem oklepaju enačb (1.6) in (1.8) podobna in podajata transformacijo za diferencialno translacijo in rotacijo. Za izraz v oklepaju izberemo nov simbol  $\Delta$ .

$$\Delta = \mathbf{Trans}(d_x, d_y, d_z) \cdot \mathbf{Rot}(\mathbf{k}, d\theta) - \mathbf{I}. \quad (1.9)$$

Za diferencialno transformacijo  $d\mathbf{T}$ , ki je podana glede na referenčni ali osnovni koordinatni sistem, bomo to zapisali v obliki (1.10):

$$d\mathbf{T} = \Delta \cdot \mathbf{T}. \quad (1.10)$$

Če bomo želeli podati diferencialno transformacijo glede na  $\mathbf{T}$  koordinatni sistem, bomo to poudarili tako, da bomo na prednjo zgornjo stran znaka  $\Delta$  napisali simbol za koordinatni sistem, torej  ${}^T\Delta$  (1.11)

$$d\mathbf{T} = \mathbf{T} \cdot {}^T\Delta. \quad (1.11)$$

V primeru enačbe (1.11) bodo diferencialne transformacije translacije in rotacije podane glede na koordinatni sistem, ki ga podaja  $\mathbf{T}$ .

V nadaljevanju nas zanima matrika, ki ustreza simbolu diferencialne transformacije  $\Delta$ . V ta namen bomo  $\mathbf{Trans}(d_x, d_y, d_z)$  in  $\mathbf{Rot}(\mathbf{k}, d\theta)$  izpisali in namesto oznake  $\mathbf{I}$  pisali enotsko matriko  $\mathbf{I}$  v obliki matrike.

$$\mathbf{Trans}(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.12)$$

V enačbi (1.12) pomenijo  $d_x, d_y$  in  $d_z$  komponente  $d_x\mathbf{i} + d_y\mathbf{j} + d_z\mathbf{k} = \mathbf{d}$  diferencialnega vektorja za diferencialni pomik.

Za  $\mathbf{Rot}(\mathbf{k}, \theta)$  pišemo že znani izraz iz [Bajd, 2002], matriko, ki opisuje rotacijo okrog poljubne osi in kjer za krajši zapis izraz v obliki  $(1 - \cos \theta)$

nadomestimo z  $\text{vers } \theta$

$$\mathbf{Rot}(\mathbf{k}, \theta) = \begin{bmatrix} k_x k_x \text{vers} \theta + \cos \theta & k_y k_x \text{vers} \theta - k_z \sin \theta & k_z k_x \text{vers} \theta + k_y \sin \theta & 0 \\ k_x k_y \text{vers} \theta + k_z \sin \theta & k_y k_y \text{vers} \theta + \cos \theta & k_z k_y \text{vers} \theta - k_x \sin \theta & 0 \\ k_x k_z \text{vers} \theta - k_y \sin \theta & k_y k_z \text{vers} \theta + k_x \sin \theta & k_z k_z \text{vers} \theta + \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.13)$$

V (1.13) moramo pri  $\theta$  upoštevati  $d\theta$  in bo torej potrebno zamenjati naslednje:

$$\begin{aligned} \lim_{\theta \rightarrow 0} \sin \theta &\rightarrow d\theta \\ \lim_{\theta \rightarrow 0} \cos \theta &\rightarrow 1 \\ \lim_{\theta \rightarrow 0} \text{vers} \theta &= \lim_{\theta \rightarrow 0} (1 - \cos \theta) \rightarrow 0. \end{aligned}$$

Torej se bo (1.13) spremenila v

$$\mathbf{Rot}(\mathbf{k}, d\theta) = \begin{bmatrix} 1 & -k_z d\theta & k_y d\theta & 0 \\ k_z d\theta & 1 & -k_x d\theta & 0 \\ -k_y d\theta & k_x d\theta & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.14)$$

Z nastavkom za  $\Delta$  ob upoštevanju enačbe (1.12) in (1.14) in (1.9) dobimo (1.15):

$$\Delta = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -k_z d\theta & k_y d\theta & 0 \\ k_z d\theta & 1 & -k_x d\theta & 0 \\ -k_y d\theta & k_x d\theta & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.15)$$

$$\Delta = \begin{bmatrix} 0 & -k_z d\theta & k_y d\theta & d_x \\ k_z d\theta & 0 & -k_x d\theta & d_y \\ -k_y d\theta & k_x d\theta & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (1.16)$$

$\Delta$  matrika v (1.16) je bila izračunana za rotacijo okoli poljubnega  $\mathbf{k}$  vektorja, toda rotacijo lahko podamo tudi kot rotacijo okoli  $x$ ,  $y$  in  $z$  osi.

### 1.3 EKVIVALENT MED DIFERENCIALNIMI ROTACIJAMI

Tukaj želimo izdelati povezavo med rotacijo za kot  $d\theta$  okoli neke poljubne osi  $\mathbf{k}$  (ki je bila opisana v  $\Delta$ ) in ustreznimi potrebnimi rotacijami  $\delta_x$  okoli

$x$ , in nato  $\delta_y$  okoli  $y$  ter  $\delta_z$  okoli  $z$  osi za dosego enake končne lege. Želimo matematični zapis te enakosti. Kot se izkaže, je končni rezultat neodvisen od vrstnega reda, po katerem izvajamo te diferencialne rotacije.

Izhajamo iz znanih izrazov za  $\mathbf{Rot}(x, \alpha)$ ,  $\mathbf{Rot}(y, \beta)$ ,  $\mathbf{Rot}(z, \gamma)$  [Bajd, 2002].

$$\mathbf{Rot}(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.17)$$

$$\mathbf{Rot}(y, \beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.18)$$

$$\mathbf{Rot}(z, \gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.19)$$

Za diferencialne rotacije velja, da je  $\sin \theta \equiv d\theta$  in  $\cos \theta$  za  $d\theta$  preide v 1. Če to vstavimo v naše enačbe, dobimo pri upoštevanju

$$\begin{aligned} d\alpha &\equiv \delta_x \\ d\beta &\equiv \delta_y \\ d\gamma &\equiv \delta_z, \end{aligned}$$

$$\mathbf{Rot}(x, \delta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\delta_x & 0 \\ 0 & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.20)$$

$$\mathbf{Rot}(y, \delta_y) = \begin{bmatrix} 1 & 0 & \delta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\delta_y & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.21)$$

$$\mathbf{Rot}(z, \delta_z) = \begin{bmatrix} 1 & -\delta_z & 0 & 0 \\ \delta_z & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.22)$$

Tako dobljene diferencialne rotacije moremo sestaviti npr. v zaporedje  $\delta_x$  okoli  $x$ , in nato  $\delta_y$  okoli  $y$  ter  $\delta_z$  okoli  $z$ , pišemo:

$$\mathbf{Rot}(x, \delta_x) \mathbf{Rot}(y, \delta_y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\delta_x & 0 \\ 0 & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \delta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\delta_y & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.23)$$

Po množenju zanemarimo člene drugega reda, kot npr.  $\delta_x \delta_y$  in dobimo izraz (1.24):

$$\mathbf{Rot}(x, \delta_x) \mathbf{Rot}(y, \delta_y) = \begin{bmatrix} 1 & 0 & \delta_y & 0 \\ 0 & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.24)$$

Poudarimo, da je rezultat neodvisen od zaporedja izvajanja izbranih diferencialnih rotacij, torej

$$\mathbf{Rot}(x, \delta_x) \mathbf{Rot}(y, \delta_y) = \mathbf{Rot}(y, \delta_y) \mathbf{Rot}(x, \delta_x). \quad (1.25)$$

Dodajmo še rotacijo  $\delta_z$  okoli  $z$  osi, to je  $\mathbf{Rot}(z, \delta_z)$

$$\mathbf{Rot}(x, \delta_x) \mathbf{Rot}(y, \delta_y) \mathbf{Rot}(z, \delta_z) = \begin{bmatrix} 1 & 0 & \delta_y & 0 \\ 0 & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\delta_z & 0 & 0 \\ \delta_z & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.26)$$

Tudi v tem zmnožku zanemarimo dvojne diferenciale člene, izraz preide v (1.27)

$$\mathbf{Rot}(x, \delta_x) \mathbf{Rot}(y, \delta_y) \mathbf{Rot}(z, \delta_z) = \begin{bmatrix} 1 & -\delta_z & \delta_y & 0 \\ \delta_z & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.27)$$

Zanimivo je spoznanje, da primerjava izraza (1.27) in izraza (1.14) za rotacijo  $d\theta$  okoli  $k$  osi vodi do naslednjih ekvivalentov (rotacijo  $d\theta$  okoli  $k$  osi moremo nadomestiti z  $\theta_x$  okoli  $x$ ,  $\theta_y$  okoli  $y$  in  $\theta_z$  okoli  $z$  osi)

$$\begin{aligned} k_x d\theta &= \delta_x \\ k_y d\theta &= \delta_y \\ k_z d\theta &= \delta_z. \end{aligned} \quad (1.28)$$

S tako dobljenimi enakostmi lahko na novo napišemo matriko  $\Delta$

$$\Delta = \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.29)$$

$\Delta$  - matrika za diferencialno translacijo in rotacijo je sestavljena iz dveh vektorjev in to  $\mathbf{d}$  vektorja ter  $\boldsymbol{\delta}$  vektorja, kjer je  $\mathbf{d}$  vektor diferencialnih translacij in  $\boldsymbol{\delta}$  vektor diferencialnih rotacij.

$$\begin{aligned} \mathbf{d} &= d_x \mathbf{i} + d_y \mathbf{j} + d_z \mathbf{k} \\ \boldsymbol{\delta} &= \delta_x \mathbf{i} + \delta_y \mathbf{j} + \delta_z \mathbf{k}. \end{aligned} \quad (1.30)$$

S pomočjo obeh vektorjev moremo sestaviti tudi stolpec  $\mathbf{D}$ , ki ga bomo imenovali diferencialni vektor gibanja.  $\mathbf{D}$  ima obliko (1.31):

$$\mathbf{D} = [d_x \ d_y \ d_z \ \delta_x \ \delta_y \ \delta_z]^T \quad (1.31)$$

Ko se bomo posluževali vektorjev  $\mathbf{d}$ ,  $\boldsymbol{\delta}$  in  $\mathbf{D}$  glede na matriko  $\Delta$ , bomo vedno za preprečevanje zmede in pri podajanju diferencialnega pomika specificiranega zapisa glede na  $\mathbf{T}$  matriko pisali  ${}^T\Delta$  (diferencialni pomik za  ${}^T\mathbf{T}$ ).  ${}^T\Delta$  pomeni tudi, da matriko sestavljajo  ${}^T\mathbf{d}$  in  ${}^T\boldsymbol{\delta}$  ozziroma da ustreza  ${}^T\mathbf{D}$ .

Naredimo primer. Dana je transformacija  $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 10 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Iščemo  $d\mathbf{A}$  za vektorje  $\mathbf{d} = 1\mathbf{i} + 0\mathbf{j} + 0,5\mathbf{k}$  in  $\delta = 0\mathbf{i} + 0,1\mathbf{j} + 0\mathbf{k}$ , vse glede na osnovni koordinatni sistem.

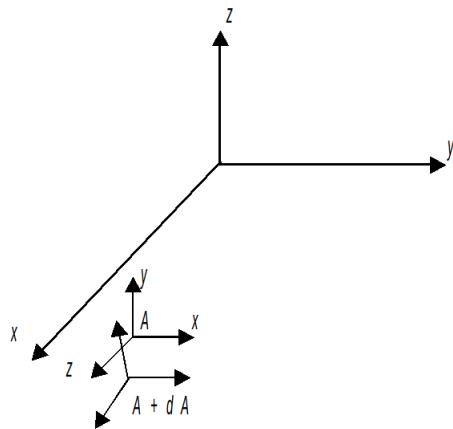
1. Sestavimo diferencialno matriko pomika  $\Delta$  glede na izraz (1.29) vpišemo ustrezne elemente

$$\Delta = \begin{bmatrix} 0 & 0 & 0,1 & 1 \\ 0 & 0 & 0 & 0 \\ -0,1 & 0 & 0 & 0,5 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Ker je  $d\mathbf{T} = \Delta \cdot \mathbf{T}$  (glej en. (1.10)), pišemo za naš primer  $d\mathbf{A} = \Delta \cdot \mathbf{A}$ , torej je

$$d\mathbf{A} = \begin{bmatrix} 0 & 0 & 0,1 & 1 \\ 0 & 0 & 0 & 0 \\ -0,1 & 0 & 0 & 0,5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 10 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0,1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0,1 & -0,5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Če želimo izvedeni račun prikazati grafično, narišimo v osnovnem koordinatnem sistemu transformacijo  $\mathbf{A}$ , ki se zaradi  $d\mathbf{A}$  premakne v novo lego  $\mathbf{A} + d\mathbf{A}$ , kot je prikazano na sliki 1.1.



Slika 1.1: Diferencialni pomik v osnovnem koordinatnem sistemu.

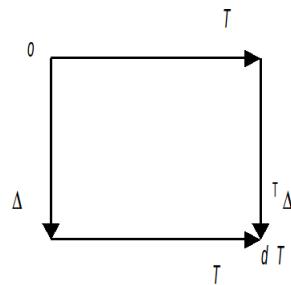
## 1.4 TRANSFORMACIJA DIFERENCIALNIH POMIKOV MED KOORDINATNIMI SISTEMI

Do sedaj smo obravnavali diferencialne transformacije glede na osnovni ali nek dani koordinatni sistem. Tu želimo obravnavati transformacije diferencialnih pomikov med koordinatnimi sistemi. Torej za dano  $\Delta$  iščemo  ${}^T\Delta$ .

Izhajamo iz (1.10) in (1.11), ki ju izenačimo

$$\Delta \cdot \mathbf{T} = \mathbf{T} \cdot {}^T\Delta. \quad (1.32)$$

To transformacijsko enakost lahko prikažemo v transformacijskem grafu. Iz grafa bomo lažje in bolj nazorno uganili, kako izračunati  ${}^T\Delta$ .

Slika 1.2: Povezava matrik  $\Delta$ ,  ${}^T\Delta$  in  $\mathbf{T}$ .

Napravimo izračun, z naslednjimi koraki

$$\begin{aligned}\Delta &= \Delta \\ \Delta \cdot T &= T \cdot {}^T\Delta \quad / \cdot T^{-1} \\ T^{-1} \cdot \Delta \cdot T &= T^{-1} \cdot T \cdot {}^T\Delta \\ T^{-1} \cdot \Delta \cdot T &= {}^T\Delta\end{aligned}$$

Spoznamo, da velja

$${}^T\Delta = T^{-1} \cdot \Delta \cdot T \quad (1.33)$$

Enačba (1.33) je zelo pomembna, ker opisuje relacijo diferencialnih pomikov med različnima koordinatnima sistemoma. Podoben zapis (1.33) bomo najprej izpeljali v splošni obliki, ker pri tem pride do vrste poenostavitev. Na koncu bomo dobili enačbo, ki bo povezovala  ${}^T\Delta$  in vektorja  $\mathbf{d}$  in  $\boldsymbol{\delta}$ . Transformacijo  $T$  poznamo iz [Bajd, 2002] in jo lahko zapišemo v obliki z enotskimi vektorji  $\mathbf{n}$ ,  $\mathbf{o}$ ,  $\mathbf{a}$  in  $\mathbf{p}$ .

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.34)$$

$\mathbf{a}$  ... approach

$\mathbf{o}$  ... orientation

$\mathbf{n}$  ... normal

$\mathbf{p}$  ... position.

Produkt  $\Delta \cdot T$ , ki vsebuje zasuke okrog posameznih osi, zapišemo kot odvisnost od  $\mathbf{d}$  in  $\boldsymbol{\delta}$

$$\Delta \cdot T = \begin{bmatrix} (\boldsymbol{\delta} \times \mathbf{n})_x & (\boldsymbol{\delta} \times \mathbf{o})_x & (\boldsymbol{\delta} \times \mathbf{a})_x & ((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d})_x \\ (\boldsymbol{\delta} \times \mathbf{n})_y & (\boldsymbol{\delta} \times \mathbf{o})_y & (\boldsymbol{\delta} \times \mathbf{a})_y & ((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d})_y \\ (\boldsymbol{\delta} \times \mathbf{n})_z & (\boldsymbol{\delta} \times \mathbf{o})_z & (\boldsymbol{\delta} \times \mathbf{a})_z & ((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d})_z \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.35)$$

Ta izraz še premultipliciramo s  $T^{-1}$  in dobimo

$$T^{-1} \Delta T = \begin{bmatrix} \mathbf{n}(\boldsymbol{\delta} \times \mathbf{n}) & \mathbf{n}(\boldsymbol{\delta} \times \mathbf{o}) & \mathbf{n}(\boldsymbol{\delta} \times \mathbf{a}) & \mathbf{n}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ \mathbf{o}(\boldsymbol{\delta} \times \mathbf{n}) & \mathbf{o}(\boldsymbol{\delta} \times \mathbf{o}) & \mathbf{o}(\boldsymbol{\delta} \times \mathbf{a}) & \mathbf{o}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ \mathbf{a}(\boldsymbol{\delta} \times \mathbf{n}) & \mathbf{a}(\boldsymbol{\delta} \times \mathbf{o}) & \mathbf{a}(\boldsymbol{\delta} \times \mathbf{a}) & \mathbf{a}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.36)$$

Elementi v matriki (1.36) imajo obliko produkta treh vektorjev in sicer  $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$ .

V takšnem produktu smemo člene prepisovati različno in vektorje zamenjati, če le preostalemu tretjemu členu zamenjamo predznak, npr.

$$\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = -\mathbf{b} \cdot (\mathbf{a} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a})$$

Velja tudi, da je vrednost enaka 0, če sta dva od treh vektorjev enaka.

$$\mathbf{a} \cdot (\mathbf{a} \times \mathbf{c}) = 0$$

Na tej podlagi sledi sklep, da so vsi diagonalni členi v matriki (1.36) enaki 0.

$$\mathbf{n} \cdot (\boldsymbol{\delta} \times \mathbf{n}) = 0, \quad \mathbf{o} \cdot (\boldsymbol{\delta} \times \mathbf{o}) = 0, \quad \mathbf{a} \cdot (\boldsymbol{\delta} \times \mathbf{a}) = 0.$$

Napišemo novo enačbo (1.37)

$${}^T \boldsymbol{\Delta} = \begin{bmatrix} 0 & -\boldsymbol{\delta} \cdot \mathbf{a} & \boldsymbol{\delta} \cdot \mathbf{o} & \boldsymbol{\delta}(\mathbf{p} \times \mathbf{n}) + \mathbf{d} \cdot \mathbf{n} \\ \boldsymbol{\delta} \cdot \mathbf{a} & 0 & -\boldsymbol{\delta} \cdot \mathbf{n} & \boldsymbol{\delta}(\mathbf{p} \times \mathbf{o}) + \mathbf{d} \cdot \mathbf{o} \\ -\boldsymbol{\delta} \cdot \mathbf{o} & \boldsymbol{\delta} \cdot \mathbf{n} & 0 & \boldsymbol{\delta}(\mathbf{p} \times \mathbf{a}) + \mathbf{d} \cdot \mathbf{a} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.37)$$

v kateri je upoštevano

$$\begin{aligned} \mathbf{n} \times \mathbf{o} &= \mathbf{a} \\ \mathbf{a} \times \mathbf{n} &= \mathbf{o} \\ \mathbf{o} \times \mathbf{a} &= \mathbf{n}. \end{aligned}$$

Vendar je  ${}^T \boldsymbol{\Delta}$  definirana tudi kot enačba (1.38)

$${}^T \boldsymbol{\Delta} = \begin{bmatrix} 0 & -{}^T \boldsymbol{\delta}_z & {}^T \boldsymbol{\delta}_y & {}^T d_x \\ {}^T \boldsymbol{\delta}_z & 0 & -{}^T \boldsymbol{\delta}_x & {}^T d_y \\ -{}^T \boldsymbol{\delta}_y & {}^T \boldsymbol{\delta}_x & 0 & {}^T d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.38)$$

Izenačimo enakoležne elemente v (1.37) in (1.38) ter dobimo vektorje diferencialne transformacije in rotacije, opisane glede na  $\mathbf{T}$  koordinatni sistem ( ${}^T \boldsymbol{\delta}, {}^T \mathbf{d}$ ), izražene kot diferencialne translacije in rotacije glede na bazne koordinate

$$\begin{aligned} {}^T d_x &= \boldsymbol{\delta}(\mathbf{p} \times \mathbf{n}) + \mathbf{d} \cdot \mathbf{n} \\ {}^T d_y &= \boldsymbol{\delta}(\mathbf{p} \times \mathbf{o}) + \mathbf{d} \cdot \mathbf{o} \\ {}^T d_z &= \boldsymbol{\delta}(\mathbf{p} \times \mathbf{a}) + \mathbf{d} \cdot \mathbf{a} \end{aligned} \quad (1.39)$$

$$\begin{aligned} {}^T \delta_x &= \boldsymbol{\delta} \cdot \mathbf{n} \\ {}^T \delta_y &= \boldsymbol{\delta} \cdot \mathbf{o} \\ {}^T \delta_z &= \boldsymbol{\delta} \cdot \mathbf{a} \end{aligned} \quad (1.40)$$

V tem zapisu so  $\mathbf{n}$ ,  $\mathbf{o}$ ,  $\mathbf{a}$  in  $\mathbf{p}$  stoljni vektorji diferencialne transformacije za  $\mathbf{T}$ . Pri tem lahko enakost enačb (1.39) in (1.40) izrazimo v  $6 \times 6$  matriki:

$$\begin{bmatrix} {}^T d_x \\ {}^T d_y \\ {}^T d_z \\ {}^T \delta_x \\ {}^T \delta_y \\ {}^T \delta_z \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & (\mathbf{p} \times \mathbf{n})_x & (\mathbf{p} \times \mathbf{n})_y & (\mathbf{p} \times \mathbf{n})_z \\ o_x & o_y & o_z & (\mathbf{p} \times \mathbf{o})_x & (\mathbf{p} \times \mathbf{o})_y & (\mathbf{p} \times \mathbf{o})_z \\ a_x & a_y & a_z & (\mathbf{p} \times \mathbf{a})_x & (\mathbf{p} \times \mathbf{a})_y & (\mathbf{p} \times \mathbf{a})_z \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} \quad (1.41)$$

Načeloma lahko enačbi (1.39) in (1.40) napišemo tudi v naslednji obliki:

$$\begin{aligned} {}^T d_x &= \mathbf{n}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ {}^T d_y &= \mathbf{o}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ {}^T d_z &= \mathbf{a}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \end{aligned} \quad (1.42)$$

$$\begin{aligned} {}^T \delta_x &= \mathbf{n} \cdot \boldsymbol{\delta} \\ {}^T \delta_y &= \mathbf{o} \cdot \boldsymbol{\delta} \\ {}^T \delta_z &= \mathbf{a} \cdot \boldsymbol{\delta} \end{aligned} \quad (1.43)$$

Zapisa (1.42) in (1.43) sta pomembna, ker ju bomo kasneje s pridom uporabili.

Napravimo primer: Podan je koordinatni sistem in diferencialna translacija ter rotacija, kot smo to že imeli. Stolpci v matriki so vektorji  $\mathbf{n}$ ,  $\mathbf{o}$ ,  $\mathbf{a}$  in  $\mathbf{p}$ .

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 10 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{aligned} \mathbf{d} &= 1\mathbf{i} + 0\mathbf{j} + 0,5\mathbf{k} \\ \boldsymbol{\delta} &= 0\mathbf{i} + 0,1\mathbf{j} + 0\mathbf{k}. \end{aligned}$$

Želimo izraziti diferencialno rotacijo in translacijo toda tako, da bo izražena v koordinatnem sistemu  $\mathbf{A}$ . Torej iščemo  ${}^A\boldsymbol{\delta}$  in  ${}^A\mathbf{d}$ .

Najprej izračunamo pare (1.42)

$$\boldsymbol{\delta} \times \mathbf{p} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0,1 & 0 \\ 10 & 5 & 0 \end{vmatrix},$$

$$\boldsymbol{\delta} \times \mathbf{p} = 0\mathbf{i} + 0\mathbf{j} - 1\mathbf{k}$$

nato prištejemo  $\mathbf{d}$

$$\begin{aligned} (\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d} &= 0\mathbf{i} + 0\mathbf{j} - 1\mathbf{k} + 1\mathbf{i} + 0\mathbf{j} + 0,5\mathbf{k} \\ &= 1\mathbf{i} + 0\mathbf{j} - 0,5\mathbf{k} \end{aligned}$$

in na podlagi en. (1.42) in (1.43) izračunamo  ${}^A\mathbf{d}$  in  ${}^A\boldsymbol{\delta}$

$${}^A d_x = \mathbf{n}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) = (0\mathbf{i} + 1\mathbf{j} + 0\mathbf{k}) \cdot (1\mathbf{i} + 0\mathbf{j} - 0,5\mathbf{k})$$

$${}^A d_x = 0$$

$${}^A d_y = \mathbf{o}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) = (0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}) \cdot (1\mathbf{i} + 0\mathbf{j} - 0,5\mathbf{k})$$

$${}^A d_y = -0,5$$

$${}^A d_z = \mathbf{a}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) = (1\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) \cdot (1\mathbf{i} + 0\mathbf{j} - 0,5\mathbf{k})$$

$${}^A d_z = 1$$

$${}^A \mathbf{d} = 0\mathbf{i} - 0,5\mathbf{j} + 1\mathbf{k}$$

$${}^A \boldsymbol{\delta} = 0,1\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}.$$

Tako dobljeni rezultati moremo preveriti, če uporabimo enačbo  $d\mathbf{T} = \mathbf{T} \cdot {}^T \Delta$ , torej  $d\mathbf{A} = \mathbf{A} \cdot {}^T \Delta$ .

Sestavimo ustrezne matrike iz vektorjev  ${}^A\mathbf{d}$  in  ${}^A\boldsymbol{\delta}$

$${}^T \Delta = \begin{bmatrix} 0 & -{}^T \delta_z & {}^T \delta_y & {}^T d_x \\ {}^T \delta_z & 0 & -{}^T \delta_x & {}^T d_y \\ -{}^T \delta_y & {}^T \delta_x & 0 & {}^T d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad {}^A \Delta = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -0,1 & -0,5 \\ 0 & 0,1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

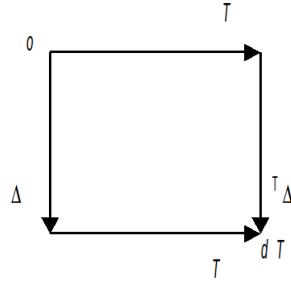
in

$$d\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 10 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -0,1 & -0,5 \\ 0 & 0,1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0,1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0,1 & -0,5 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

$d\mathbf{A}$  smo že narisali na sliki 1.1.

## 1.5 DIFERENCIJALNE RELACIJE V TRANSFORMACIJSKIH IZRAZIH

Enačbi (1.42) in (1.43) podajajo elemente transformacije  ${}^T\Delta$  glede na elemente diferencialne transformacije in to glede na koordinatni sistem  $\mathbf{T}$ ,  $\Delta$  pa podaja diferencialne transformacije glede na osnovni koordinatni sistem. Če želimo najti vektorje v  $\Delta$  in imamo podano  ${}^T\Delta$ , potem si pri preračunavanju pomagamo s sliko enakosti 1.3.



Slika 1.3: Slika enakosti

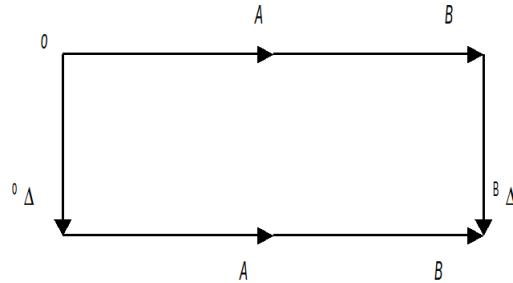
$$\begin{aligned}
 d\mathbf{T} &= \Delta \cdot \mathbf{T} \\
 d\mathbf{T} &= \mathbf{T} \cdot {}^T\Delta \\
 \Delta &= \mathbf{T} \cdot {}^T\Delta \cdot \mathbf{T}^{-1} \\
 \Delta &= (\mathbf{T}^{-1})^{-1} \cdot {}^T\Delta \cdot \mathbf{T}^{-1}
 \end{aligned} \tag{1.44}$$

Končna enačba ima sedaj isto obliko kot en. (1.33). Torej lahko uporabimo en. (1.42) in (1.43) za izračun diferencialnih vektorjev za  $\Delta$ , če seveda uporabimo elemente iz  $\mathbf{T}^{-1}$  kot diferencialno koordinatno transformacijo.

Z namenom da uredimo diferencialne spremembe med dvema koordinatnima sistemoma  $\mathbf{A}$  in  $\mathbf{B}$ , da dobimo  $\Delta$ , če je podana  ${}^B\Delta$ , računamo

$$\Delta \cdot \mathbf{A} \cdot \mathbf{B} = \mathbf{A} \cdot \mathbf{B} \cdot {}^B\Delta \tag{1.45}$$

Za to enačbo moremo narisati sliko 1.4.



Slika 1.4: Preslikava diferencialnih sprememb med dvema sistemoma.

Za  $\Delta$  lahko iz slike direktno napišemo izraz

$$\Delta = \mathbf{A} \cdot \mathbf{B} \cdot {}^B\Delta \cdot \mathbf{B}^{-1} \cdot \mathbf{A}^{-1} \quad (1.46)$$

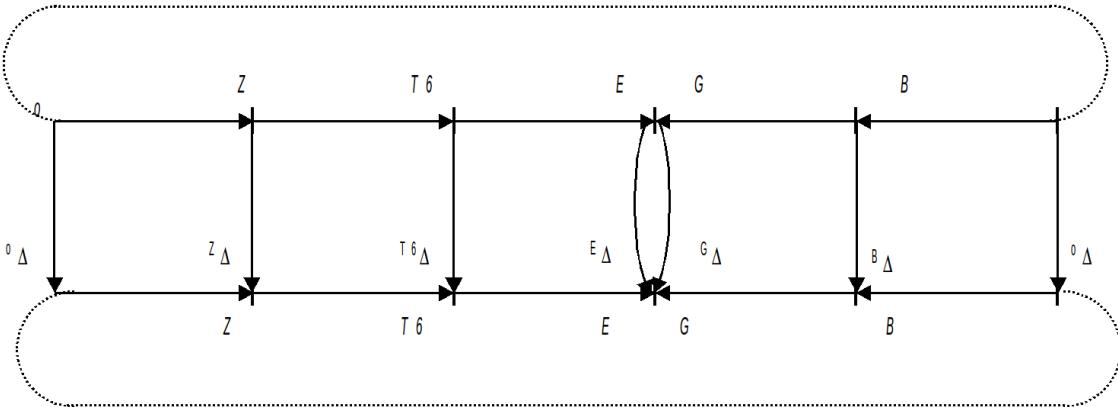
Napišemo v spremenljivi obliki

$$\Delta = (\mathbf{B}^{-1} \cdot \mathbf{A}^{-1})^{-1} \cdot {}^B\Delta \cdot (\mathbf{B}^{-1} \cdot \mathbf{A}^{-1}) \quad (1.47)$$

in nato uporabimo  $(\mathbf{B}^{-1} \cdot \mathbf{A}^{-1})$  kot diferencialno koordinatno transformacijo namesto za  $\mathbf{T}$ , kot to zahtevata enačbi (1.42) in (1.43). Dobimo elemente za  $\Delta$ . Izraze za diferencialno koordinatno transformacijo moremo direktno dobiti iz grafa na sliki, če sledimo poti nazaj od vrha (puščice) vektorja dane diferencialne transformacije  ${}^B\Delta$  proti vrhu (puščici) želene posledice ekvivalentne diferencialne transformacije  $\Delta$ . Iz slike razberemo, da je to ravno  $(\mathbf{B}^{-1} \cdot \mathbf{A}^{-1})$ . Pozor,  $\mathbf{B}^{-1}$  moramo postmoltiplicirati z  $\mathbf{A}^{-1}$ , ker premik opravljamo glede na relativni koordinatni sistem  ${}^B\Delta$ . Ko računamo, računamo  ${}^B\Delta$  kot posledico vzroka  $\Delta$ . Zopet je postmoltipliciranje, ker gre za relativni koordinatni sistem.

Splošen graf za diferencialne premike je podan na sliki 1.5, ki ponazarja naslednjo transformacijo

$$\mathbf{Z} \cdot \mathbf{T}_6 \cdot \mathbf{E} = \mathbf{B} \cdot \mathbf{G} \quad (1.48)$$



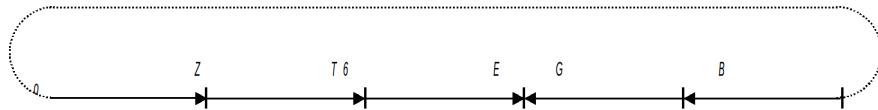
Slika 1.5: Preslikava diferencialnih sprememb med več kordinatnimi sistemi.

Kot vemo, moremo s pomočjo grafa takoj napisati enačbo za izračun poljubnega elementa. Velja, da diferencialna sprememba v enem elementu - koordinatnem sistemu povzroči posledično ustrezno spremembo v drugem koordinatnem sistemu. Npr. iščemo  ${}^{T6}\Delta$  diferencialno spremembo v  $\mathbf{T}_6$ , ki je posledica zaradi spremembe (vzroka) v  $\mathbf{B} \cdot {}^B\Delta$ . Zanima nas sprememba  ${}^{T6}\Delta$ , ki je rezultat, posledica zaradi vzroka  ${}^B\Delta$ . Torej bo ta vpliv prišel od  $\mathbf{B}$  preko  $\mathbf{G} \cdot \mathbf{E}^{-1}$  kot diferencialna koordinatna transformacija, ki jo moramo uporabiti namesto  $\mathbf{T}$  koordinatnega sistema v enačbah (1.42) in (1.43) iz prejšnjega odseka. Seveda bi lahko gledali, kako bi diferencialne spremembe v  ${}^B\Delta$  vplivale na koordinatni sistem  $\mathbf{T}_6$  po nasprotni drugi poti preko  $\mathbf{B}^{-1} \cdot \mathbf{Z} \cdot \mathbf{T}_6$ . Vedno sta možni dve poti (od vrha puščice vzroka diferencialnega pomika do vrha puščice sistema, ki nas zanima). Torej v tej drugi poti bi namesto  $\mathbf{T}$  vstavljeni v en. (1.42) in (1.43) izraz  $\mathbf{B}^{-1} \cdot \mathbf{Z} \cdot \mathbf{T}_6$ .

Nadaljnji primer: sprememba  ${}^Z\Delta$  nas zanima kot posledica vzroka  ${}^B\Delta$ , torej bo  $\mathbf{T}$  ustrezal po eni poti  $\mathbf{B}^{-1} \cdot \mathbf{Z}$ , po drugi poti pa  $\mathbf{G} \cdot \mathbf{E}^{-1} \cdot \mathbf{T}_6^{-1}$ .

Kot smo pokazali, znamo narisati diagram za diferencialne transformacije, če upoštevamo pravilo, da moramo začeti na vrhu diferencialnega koordinatnega sistema, ki povzroča spremembo in ustrezno končati na vrhu tistega koordinatnega sistema, kjer nas zanimajo posledice diferencialnega pomika, transformacije. Za takšno poenostavljeni ponazoritev zadošča poenostavljeni diagram.

## 26 POGLAVJE 1. DIF. RELACIJE V TRANSFORMACIJSKIH IZRAZIH



Slika 1.6: Poenostavljena slika za več koordinatnih sistemov.

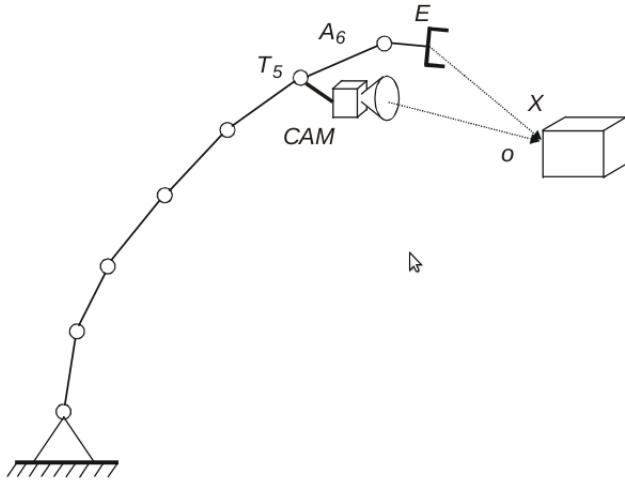
Diferencialno koordinatno transformacijo dobimo direktno kot pot od vrha tiste transformacije, kjer nastopa diferencialna sprememba, do vrha tistega koordinatnega sistema, kjer nas posledica zanima. Diferencialna koordinatna transformacija, ki jo moramo uporabiti namesto  $\mathbf{T}$  v enačbah (1.42) in (1.43) je tako dobljena. Npr., da najdemo  $\Delta$ , ki je posledica spremembe  ${}^{T6}\Delta$ , lahko s pomočjo slike direktno napišemo:

$$\begin{aligned} \mathbf{T} &= \mathbf{T}_6^{-1} \cdot \mathbf{Z}^{-1} \quad \text{ali} \quad \mathbf{T} = \mathbf{E} \cdot \mathbf{G}^{-1} \cdot \mathbf{B}^{-1} \\ \Delta &= (\mathbf{T}_6^{-1} \cdot \mathbf{Z}^{-1})^{-1} \cdot {}^{T6}\Delta \cdot (\mathbf{T}_6^{-1} \cdot \mathbf{Z}^{-1}) \end{aligned} \quad (1.49)$$

Naredimo primer za ilustracijo:

Kamero za umetni vid imamo nameščeno na 5. segmentu manipulatorja. Povezava je definirana z

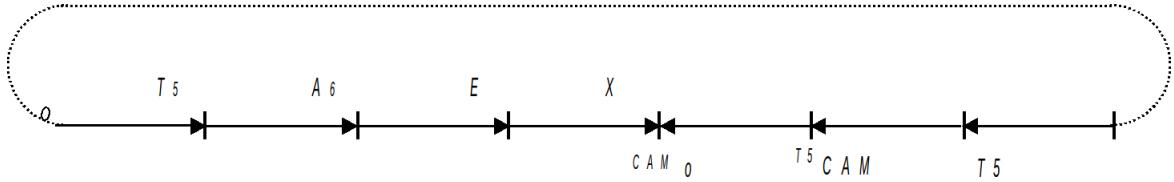
- $\mathbf{T}_5$  ... opisuje lego segmenta 5 glede na osnovni koordinatni sistem,
- $\mathbf{A}_6$  ... opisuje lego segmenta 6 glede na koordinatni sistem segmenta 5,
- $\mathbf{E}$  ... opisuje lego prijemala glede na  $\mathbf{A}_6$ ,
- $\mathbf{X}$  ... je nepoznana transformacija, ki opisuje lego predmeta glede na lego prijemala,
- $\mathbf{o}$  ... opisuje predmet  $o$  v koordinatnem sistemu kamere.



Slika 1.7: Transformacija za prikazani primer.

$${}^T \mathbf{CAM} = \begin{bmatrix} 0 & 0 & -1 & 5 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Narišimo graf razmer:



Slika 1.8: Graf transformacij za kinematično verigo na gornji sliki.

Zadnji segment manipulatorja je opisan v trenutni legi glede na prejšnji segment z

$$\mathbf{A}_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Nek predmet  $\mathbf{o}$ ,  ${}^{CAM}\mathbf{O}$ , je viden v kameri in diferencialni pomik v  $CAM$  koordinatah je podan z namenom, da bi prijemalo  $E$  prišlo v dotik z objektom (torej  $\mathbf{X} = 0$ ).

$${}^{CAM}\mathbf{d} = -1\mathbf{i} + 1\mathbf{j} + 0\mathbf{k} \quad \text{in} \quad {}^{CAM}\boldsymbol{\delta} = 0\mathbf{i} + 0\mathbf{j} + 0,1\mathbf{k}$$

## 28 POGLAVJE 1. DIF. RELACIJE V TRANSFORMACIJSKIH IZRAZIH

Matrika  $\mathbf{CAM}$  je kalibrirana tako, da ko  $\mathbf{E}$  kaže v koordinatno središče  $\mathbf{CAM}$ , potem velja  ${}^{\mathbf{CAM}}\mathbf{d} = {}^{\mathbf{CAM}}\boldsymbol{\delta} = \mathbf{0}$ .

Posledično, če je predmet viden nekje v  $\mathbf{CAM}$ , potem mora  $\mathbf{E}$  kazati tja, to pa bo tedaj, če se bo  $\mathbf{A}_6$  vrh premaknil za ustrezno  ${}^{\mathbf{CAM}}\mathbf{d}$  in  ${}^{\mathbf{CAM}}\boldsymbol{\delta}$ . Tedaj bo tudi vrh  $\mathbf{E}$  kazal v koordinato predmeta.

Hočemo torej izračunati potrebne diferencialne pomike v  $\mathbf{A}_6$  koordinatah.

Razmere so opisane z naslednjimi transformacijskimi enačbami (glej sliko 1.8)

$$\mathbf{T}_5 \cdot \mathbf{A}_6 \cdot \mathbf{E} \cdot \mathbf{X} = \mathbf{T}_5 \cdot \mathbf{CAM} \cdot \mathbf{o}$$

Rešitev: Poiščemo  $\mathbf{T}$  diferencialno koordinatno transformacijo od vrha vzroka  $\mathbf{CAM}$  do vrha posledice  $\mathbf{A}_6$  in to tako, da se izognemo poti preko neznane  $\mathbf{X}$ .

Važno: vrh  $\mathbf{A}_6$  je posledica, ki se lahko premika in ne  $\mathbf{E}$ , ki pripeta na vrh manipulatorja.

$$\begin{aligned}\mathbf{T} &= \mathbf{CAM}^{-1} \cdot \mathbf{T}_5^{-1} \cdot \mathbf{T}_5 \cdot \mathbf{A}_6 \\ \mathbf{T} &= \mathbf{CAM}^{-1} \cdot \mathbf{A}_6\end{aligned}$$

Izračunati moramo  $\mathbf{CAM}^{-1}$  in le to množiti z  $\mathbf{A}_6$ , da dobimo  $\mathbf{T}$ . V nadaljevanju računamo  $\mathbf{CAM}^{-1}$  na osnovi poznavanja splošne oblike za  $\mathbf{T}^{-1}$  [Bajd, 2002]

$$\mathbf{T}^{-1} = \begin{bmatrix} n_x & n_y & n_z & -\mathbf{n} \cdot \mathbf{p} \\ o_x & o_y & o_z & -\mathbf{o} \cdot \mathbf{p} \\ a_x & a_y & a_z & -\mathbf{a} \cdot \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{torej} \quad \mathbf{CAM}^{-1} = \begin{bmatrix} 0 & 0 & -1 & 10 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T} = \mathbf{CAM}^{-1} \cdot \mathbf{A}_6$$

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & -1 & 10 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 2 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Uporabimo enačbi (1.42) in (1.43)

$$\begin{aligned}\boldsymbol{\delta} \times \mathbf{p} &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0 & 0,1 \\ 2 & 0 & 5 \end{vmatrix} = 0\mathbf{i} + 0,2\mathbf{j} + 0\mathbf{k} \\ (\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d} &= 0\mathbf{i} + 0,2\mathbf{j} + 0\mathbf{k} + (-1\mathbf{i} + 1\mathbf{k} + 0\mathbf{k}) \\ &= -1\mathbf{i} + 1,2\mathbf{j} + 0\mathbf{k}\end{aligned}$$

Računamo komponente za  ${}^{A_6}\Delta$  ((1.42) in (1.43))

$$\begin{aligned}{}^{A_6}\mathbf{d}_x &= \mathbf{n}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) = (0\mathbf{i} - 1\mathbf{j} + 0\mathbf{k}) \cdot (-1\mathbf{i} + 1,2\mathbf{j} + 0\mathbf{k}) = -1,2 \\ {}^{A_6}\mathbf{d}_y &= \mathbf{o}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) = (0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}) \cdot (-1\mathbf{i} + 1,2\mathbf{j} + 0\mathbf{k}) = 0 \\ {}^{A_6}\mathbf{d}_z &= \mathbf{a}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) = (-1\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) \cdot (-1\mathbf{i} + 1,2\mathbf{j} + 0\mathbf{k}) = 1 \\ {}^{A_6}\mathbf{d} &= \underline{-1,2\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}}\end{aligned}$$

In še  ${}^{A_6}\boldsymbol{\delta}$

$$\begin{aligned}{}^{A_6}\boldsymbol{\delta}_x &= \mathbf{n} \cdot \boldsymbol{\delta} = (0\mathbf{i} - 1\mathbf{j} + 0\mathbf{k}) \cdot (0\mathbf{i} + 0\mathbf{j} + 0,1\mathbf{k}) = 0 \\ {}^{A_6}\boldsymbol{\delta}_y &= \mathbf{o} \cdot \boldsymbol{\delta} = (0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}) \cdot (0\mathbf{i} + 0\mathbf{j} + 0,1\mathbf{k}) = 0,1 \\ {}^{A_6}\boldsymbol{\delta}_z &= \mathbf{a} \cdot \boldsymbol{\delta} = (-1\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) \cdot (0\mathbf{i} + 0\mathbf{j} + 0,1\mathbf{k}) = 0 \\ {}^{A_6}\boldsymbol{\delta} &= \underline{0\mathbf{i} - 0,1\mathbf{j} + 0\mathbf{k}}\end{aligned}$$

Velja  $d\mathbf{A}_6 = \mathbf{A}_6 \cdot {}^{A_6}\Delta$  in ker smo izračunali  ${}^{A_6}\mathbf{d}$  in  ${}^{A_6}\boldsymbol{\delta}$ , tako poznamo  ${}^{A_6}\Delta$  in brez težav lahko računamo  $d\mathbf{A}_6$ .

*30 POGLAVJE 1. DIF. RELACIJE V TRANSFORMACIJSKIH IZRAZIH*

## Poglavlje 2

# JACOBIJEVA MATRIKA ZA MANIPULATOR

Manipulator je podan s produktom matrik  $A_1 \cdot \dots \cdot A_6$ , ki ustreza podani  $T_6$  matriki. Diferencialni premiki v poziciji in orientaciji  $T_6$  so povzročeni zaradi diferencialnih sprememb v koordinatah sklepov  $dq_i$ . V primeru navadnega rotacijskega sklepa ustreza  $dq_i = d\theta_i$  in za prizmatični translacijski sklep  $dq_i = dd_i$ . Poznano je, da velja enakost

$$T_6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6. \quad (2.1)$$

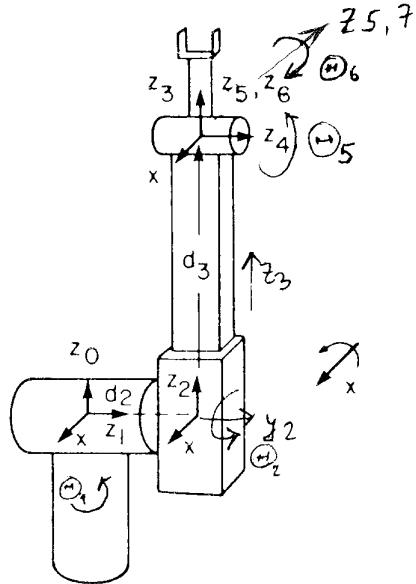
Za to enakost moremo narisati tudi transformacijski graf.



Slika 2.1: Transformacijski graf.

Dogovor za sestavo  $\mathbf{A}$  matrik rotacijskega sklepa pravi, da se rotacija  $\theta$  dogaja okoli  $z$  osi koordinatnega sistema segmenta. Npr., za manipulator Stanford bo za 3. sklep spremenljivka translacija vzdolž  $z$  osi koordinatnega sistema  $A_2$ . 2. sklep rotira okoli  $z$  osi koordinatnega sistema  $A_1$ .

Definirali bomo transformacijo  $\Delta_i$  za diferencialno spremembo, ki bo pomnila diferencialno rotacijo za enoto okoli  $z$  osi, če je seveda  $i$  sklep rotacijski, oz. diferencialno translacijo za enoto vzdolž  $z$  osi v primeru translacijskega  $i$ -tega sklepa. V obeh primerih se diferencialna transformacija nanaša na prejšnji sklep. Za  $i - 1$  koordinate jo zato pišemo kot  ${}^{i-1}\Delta_i$ . Nekateri to



Slika 2.2: Koordinatni sistemi za 5R 1T Stanford Manipulator

opišejo tudi z uporabo vektorjev.

Želimo napisati enačbo za diferencialno spremembo  $d\mathbf{T}_6$ . ki predstavlja diferencialno spremembo v  $\mathbf{T}_6$  koordinatah, torej rabimo  ${}^{T_6}\Delta_i$  in seveda v odvisnosti od diferencialne spremembe poljubnega sklepa  $q_i$

$$d\mathbf{T}_6 = \mathbf{T}_6 \cdot {}^{T_6}\Delta_i \cdot dq_i = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \dots \cdot \mathbf{A}_{i-1} \cdot {}^{i-1}\Delta_i \cdot \mathbf{A}_i \cdot \dots \cdot \mathbf{A}_6 \cdot dq_i \quad (2.2)$$

in torej sledi

$$\frac{\partial \mathbf{T}_6}{\partial q_i} = \mathbf{T}_6^{T_6} \cdot \Delta_i \quad (2.3)$$

kar pomeni, da mora biti  ${}^{T_6}\Delta_i$  enaka

$${}^{T_6}\Delta = (\mathbf{A}_i \cdot \mathbf{A}_{i+1} \cdot \dots \cdot \mathbf{A}_6)^{-1} \cdot {}^{i-1}\Delta_i \cdot (\mathbf{A}_i \cdot \mathbf{A}_{i+1} \cdot \dots \cdot \mathbf{A}_6). \quad (2.4)$$

Pri tem gre za transformacijo koordinatnega sistema  ${}^{i-1}\mathbf{T}_6$

$$(\mathbf{A}_i \cdot \mathbf{A}_{i+1} \cdot \dots \cdot \mathbf{A}_6) = {}^{i-1}\mathbf{T}_6. \quad (2.5)$$

Izraz (2.5) bi lahko dobili direktno iz slike grafa, če bi izhajali iz vrha segmenta  $A_i$  in potovali do vrha segmenta, ki meji na  $T_6$  konico.

Če je  $i$ -ti sklep rotacijski, to pomeni, da je  $d_i = 0$  in prej obravnavani enačbi (1.42) in (1.43) se poenostavita, ter dobimo

$$\begin{aligned} {}^{T_6}d_{i_x} &= \mathbf{n} \cdot (\boldsymbol{\delta} \times \mathbf{p}) \\ {}^{T_6}d_{i_y} &= \mathbf{o} \cdot (\boldsymbol{\delta} \times \mathbf{p}) \\ {}^{T_6}d_{i_z} &= \mathbf{a} \cdot (\boldsymbol{\delta} \times \mathbf{p}) \end{aligned} \quad (2.6)$$

$$\begin{aligned} {}^{T_6}\delta_{i_x} &= \mathbf{n} \cdot \boldsymbol{\delta} \\ {}^{T_6}\delta_{i_y} &= \mathbf{o} \cdot \boldsymbol{\delta} \\ {}^{T_6}\delta_{i_z} &= \mathbf{a} \cdot \boldsymbol{\delta}. \end{aligned} \quad (2.7)$$

Ker vemo, da je  $\delta_i = 0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}$  lahko enačbi (2.6) in (2.7) poenostavimo v

$$\boldsymbol{\delta} \times \mathbf{p} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0 & 1 \\ p_x & p_y & p_z \end{vmatrix} = -p_y\mathbf{i} + p_x\mathbf{j} + 0\mathbf{k}$$

$$\begin{aligned} {}^{T_6}d_{i_x} &= \mathbf{n} \cdot (\boldsymbol{\delta} \times \mathbf{p}) = (n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k}) \cdot (-p_y\mathbf{i} + p_x\mathbf{j} + 0\mathbf{k}) \\ &= -n_xp_y + n_yp_x \end{aligned}$$

$$\begin{aligned} {}^{T_6}d_{i_y} &= \mathbf{o} \cdot (\boldsymbol{\delta} \times \mathbf{p}) = (o_x\mathbf{i} + o_y\mathbf{j} + o_z\mathbf{k}) \cdot (-p_y\mathbf{i} + p_x\mathbf{j} + 0\mathbf{k}) \\ &= -o_xp_y + o_yp_x \end{aligned}$$

$${}^{T_6}d_{i_z} = \mathbf{a} \cdot (\boldsymbol{\delta} \times \mathbf{p}) = -a_xp_y + a_yp_x.$$

Gornje tri rezultate prepišemo skupaj

$${}^{T_6}\mathbf{d}_i = (-n_xp_y + n_yp_x)\mathbf{i} + (-o_xp_y + o_yp_x)\mathbf{j} + (-a_xp_y + a_yp_x)\mathbf{k} \quad (2.8)$$

in za rotacijski del

$${}^{T_6}\delta_{i_x} = \mathbf{n} \cdot \boldsymbol{\delta} = (n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k}) \cdot (0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}) = n_z.$$

Prav tako bi napravili za preostali komponenti.

$${}^{T_6}\boldsymbol{\delta}_i = n_z\mathbf{i} + o_z\mathbf{j} + a_z\mathbf{k}. \quad (2.9)$$

Če imamo translacijski sklep, je  $\boldsymbol{\delta} = 0$  in  $\mathbf{d} = 0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}$  in enačbi (1.42) in (1.43) spremenita obliko

$${}^{T_6}\mathbf{d}_i = n_z\mathbf{i} + o_z\mathbf{j} + a_z\mathbf{k} \quad (2.10)$$

$${}^{T_6}\boldsymbol{\delta}_i = 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}. \quad (2.11)$$

Diferencialne spremembe pozicije in orientacije za  $\mathbf{T}_6$  kot funkcije šestih koordinat sklepov se da napisati kot  $6 \times 6$  matriko, ki je sestavljena iz diferencialnih rotacijskih in diferencialnih translacijskih vektorskih elementov in je znana pod imenom *analitična Jacobijeva matrika*. Vsak stolpec je sestavljen iz diferencialnega translacijskega in rotacijskega vektorja, ki ustreza diferencialnim spremembam za vsak sklep v koordinatah sklepa.

$$\begin{bmatrix} T_6 d_x \\ T_6 d_y \\ T_6 d_z \\ T_6 \delta_x \\ T_6 \delta_y \\ T_6 \delta_z \end{bmatrix} = \begin{bmatrix} T_6 d_{1x} & T_6 d_{2x} & T_6 d_{3x} & T_6 d_{4x} & T_6 d_{5x} & T_6 d_{6x} \\ T_6 d_{1y} & T_6 d_{2y} & T_6 d_{3y} & T_6 d_{4y} & T_6 d_{5y} & T_6 d_{6y} \\ T_6 d_{1z} & T_6 d_{2z} & T_6 d_{3z} & T_6 d_{4z} & T_6 d_{5z} & T_6 d_{6z} \\ T_6 \delta_{1x} & T_6 \delta_{2x} & T_6 \delta_{3x} & T_6 \delta_{4x} & T_6 \delta_{5x} & T_6 \delta_{6x} \\ T_6 \delta_{1y} & T_6 \delta_{2y} & T_6 \delta_{3y} & T_6 \delta_{4y} & T_6 \delta_{5y} & T_6 \delta_{6y} \\ T_6 \delta_{1z} & T_6 \delta_{2z} & T_6 \delta_{3z} & T_6 \delta_{4z} & T_6 \delta_{5z} & T_6 \delta_{6z} \end{bmatrix} \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix} \quad (2.12)$$

$$\begin{bmatrix} T_6 \mathbf{d} \\ T_6 \boldsymbol{\delta} \end{bmatrix} = [\mathbf{J}] [d\mathbf{q}_i]$$

## 2.1 IZRAČUN JACOBIJEVE Matrike za STANFORD MANIPULATOR

Izračun Jacobijeve matrike bomo ilustrirali z izračunom Jacobijeve matrike za Stanford 5R + 1T manipulator. Za ta manipulator najprej določimo  $\mathbf{A}_1$  do  $\mathbf{A}_6$  [Bajd, 2002]

$$\mathbf{A}_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.13)$$

Za izračun vseh stolpcev v Jacobijevi matriki potrebujemo diferencialne koordinatne spremembe, ki so posledica vsake od šestih diferencialnih sprememb spremenljivk in to  $d\theta_1, d\theta_2, dd_3, d\theta_4, d\theta_5$  in  $d\theta_6$  glede na en. (2.5), ki podaja elemente v  $\mathbf{T}$

$$\mathbf{T} = (\mathbf{A}_i \cdot \mathbf{A}_{i+1} \cdot \dots \cdot \mathbf{A}_6) \cdot {}^{i-1}\mathbf{T}_6$$

## 2.1. IZRAČUN JACOBIJEVE MATRIKE ZA STANFORD MANIPULATOR35

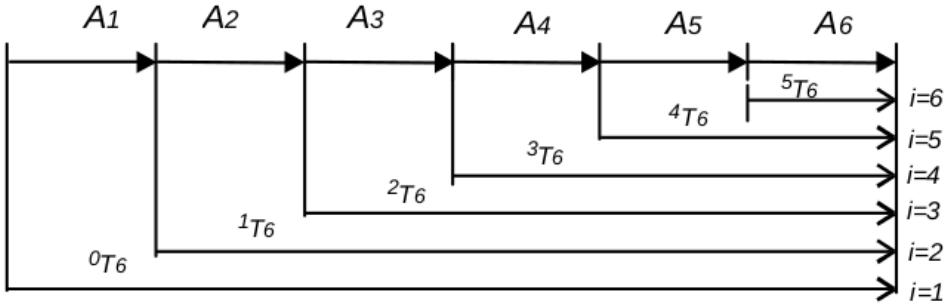
Za prvi sklep ali transformacijo  $\mathbf{A}_1$ , to je  $i = 1$  velja

$$\mathbf{T} = {}^{i-1}\mathbf{T}_6 = {}^0\mathbf{T}_6.$$

Izračunati bomo morali, kako se zaradi  $d\theta_1$  spremenijo komponente  ${}^T_6 d_{1_x}$ ,  ${}^T_6 d_{1_y}$ ,  ${}^T_6 d_{1_z}$ ,  ${}^T_6 \delta_{1_x}$ ,  ${}^T_6 \delta_{1_y}$  in  ${}^T_6 \delta_{1_z}$ . Za drugo spremenljivko ( $i = 2$ )

$$\mathbf{T} = {}^{2-1}\mathbf{T}_6 = {}^1\mathbf{T}_6 \quad \text{in tretjo} \quad \mathbf{T} = {}^{3-1}\mathbf{T}_6 = {}^2\mathbf{T}_6.$$

Spoznamo, da moramo izhajati pri izračunavanju stolpcev zapored iz matrik  ${}^0\mathbf{T}_6$ ,  ${}^1\mathbf{T}_6$ ,  ${}^2\mathbf{T}_6$  itd., ki so definirane, kot sledi s slike 2.3. Te matrike so bile izračunane, ko smo računali matriko  $\mathbf{T}_6$  kot odvisnost od  $\mathbf{A}_1 \dots \mathbf{A}_6$ .



Slika 2.3: Matrike za izračun Jacobijeve matrike.

Torej bo glede na enačbo (2.5) in sliko 2.3 prvi stolpec v Jacobijevi matriki ustrezal odvodu  $\partial\mathbf{T}_6/\partial\theta_1$  in za to odvisnost moramo izračunati glede na potrebe naših enačb odvisnosti  ${}^T_6 d_{1_x}$ ,  ${}^T_6 d_{1_y}$ ,  ${}^T_6 d_{1_z}$ ,  ${}^T_6 \delta_{1_x}$ ,  ${}^T_6 \delta_{1_y}$ ,  ${}^T_6 \delta_{1_z}$ , kot smo to pokazali v en. (2.8) in (2.9).

Elemente matrike

$$\mathbf{T}_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

ki smo jih že izračunali za Stanford manipulator [Bajd, 2002], bomo tukaj

prepisali in označili za en. (2.14).

$$\begin{aligned}
 n_x &= C_1 [C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \\
 n_y &= S_1 [C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6) \\
 n_z &= -S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \\
 o_x &= C_1 [-C_2(C_4C_5C_6 - S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6) \\
 o_y &= S_1 [-C_2(C_4C_5C_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5S_6 + C_4C_6) \\
 o_z &= S_2(C_4C_5C_6 + S_4C_6) + C_2S_5S_6 \\
 a_x &= C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5 \\
 a_y &= S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5 \\
 a_z &= -S_2C_4S_5 + C_2C_5 \\
 p_x &= C_1S_2d_3 - S_1d_2 \\
 p_y &= S_1S_2d_3 + C_1d_2 \\
 p_z &= C_2d_3
 \end{aligned} \tag{2.14}$$

Te vrednosti moramo zapored vstaviti v  ${}^{T_6}d_{1_x}$ ,  ${}^{T_6}d_{1_y}$ ,  ${}^{T_6}d_{1_z}$ ,  ${}^{T_6}\delta_{1_x}$ ,  ${}^{T_6}\delta_{1_y}$ ,  ${}^{T_6}\delta_{1_z}$  in dobimo npr.

$$\begin{aligned}
 {}^{T_6}d_{1_x} &= -n_xp_y + n_yp_x \\
 {}^{T_6}d_{1_x} &= \\
 &- \{C_1 [C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6)\} \{S_1S_2d_3 + C_1d_2\} + \\
 &+ \{S_1 [C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6)\} \{C_1S_2d_3 - S_1d_2\}
 \end{aligned} \tag{2.15}$$

$$\begin{aligned}
 {}^{T_6}d_{1_y} &= \\
 &- \{C_1 [C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(-S_4C_5S_6 + C_4C_6)\} \{S_1S_2d_3 + C_1d_2\} + \\
 &+ \{S_1 [-C_2(C_4C_5C_6 + S_4S_6) + S_2S_5C_6] + C_1(-S_4C_5S_6 + C_4C_6)\} \{C_1S_2d_3 - S_1d_2\}
 \end{aligned} \tag{2.16}$$

$$\begin{aligned}
 {}^{T_6}d_{1_z} &= \\
 &- \{C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5\} \{S_1S_2d_3 + C_1d_2\} + \\
 &+ \{S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5\} \{C_1S_2d_3 - S_1d_2\}
 \end{aligned} \tag{2.17}$$

$${}^{T_6}\delta_{1_x} = -S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \tag{2.18}$$

$${}^{T_6}\delta_{1_y} = S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \tag{2.19}$$

$${}^{T_6}\delta_{1_z} = -S_2C_4S_5 + C_2C_5. \tag{2.20}$$

## 2.1. IZRAČUN JACOBIJEVE MATRIKE ZA STANFORD MANIPULATOR37

V  $\mathbf{d}$  komponentah zmnožimo in poenostavimo člene v en. (2.15), (2.16) in (2.17) do

$${}^{T_6}d_{1_x} = -d_2 [C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + S_2d_3(S_4C_5C_6 + C_4S_6) \quad (2.21)$$

$${}^{T_6}d_{1_y} = -d_2 [-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + S_2d_3(-S_4C_5S_6 + C_4C_6) \quad (2.22)$$

$${}^{T_6}d_{1_z} = -d_2(C_2C_4S_5 + S_2C_5) + S_2d_3S_4S_5. \quad (2.23)$$

Enačbe (2.18) do (2.20) za komponente  ${}^{T_6}\delta_{1_x}$ ,  ${}^{T_6}\delta_{1_y}$  in  ${}^{T_6}\delta_{1_z}$  prepišemo in tako dobimo vse elemente prvega stolpca Jacobijeve matrike v

$$\begin{aligned} \frac{\partial \mathbf{T}_6}{\partial \theta_1} &= \begin{bmatrix} -d_2[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + S_2d_3(S_4C_5C_6 + C_4S_6) \\ -d_2[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + S_2d_3(-S_4C_5S_6 + C_4C_6) \\ -d_2(C_2C_4S_5 + S_2C_5) + S_2d_3S_4S_5 \\ -S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \\ S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \\ -S_2C_4S_5 + C_2C_5 \end{bmatrix} \\ &= \begin{bmatrix} {}^{T_6}d_{1_x} \\ {}^{T_6}d_{1_y} \\ {}^{T_6}d_{1_z} \\ {}^{T_6}\delta_{1_x} \\ {}^{T_6}\delta_{1_y} \\ {}^{T_6}\delta_{1_z} \end{bmatrix}. \end{aligned} \quad (2.24)$$

Rotacijski členi, npr.  ${}^{T_6}\delta_{1_x}$ ,  ${}^{T_6}\delta_{1_y}$  in  ${}^{T_6}\delta_{1_z}$  so po vrednosti vedno med -1 in 1.

Podobno bomo postopali in izračunali vseh ostalih 5 stolpcov. Lotimo se izračuna za drugi stolpec Jacobijeve matrike

$${}^{T_6}d_{i_x} = -n_x p_y + n_y p_x$$

$${}^{T_6}d_{i_y} = -o_x p_y + o_y p_x$$

$${}^{T_6}d_{i_z} = -a_x p_y + a_y p_x$$

$${}^{T_6}\delta_{i_x} = n_z$$

$${}^{T_6}\delta_{i_y} = o_z$$

$${}^{T_6}\delta_{i_z} = a_z.$$

Drugi stolpec Jacobijeve matrike dobimo kot  $\partial \mathbf{T}_6 / \partial \theta_2$ . Glede na enačbo (2.5) bomo v  $\mathbf{T}$  vstavljal transformacijo  ${}^1\mathbf{T}_6$ . Le to smo že izračunali in je enaka

$${}^1\mathbf{T}_6 = \begin{bmatrix} C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6 & -C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6 & S_2d_3 \\ & C_2C_4S_5 + S_2C_5 & \\ S_2(C_4C_5C_6 - S_4S_6) + C_2S_5C_6 & -S_2(C_4C_5S_6 + S_4C_6) - C_2S_5S_6 & -C_2d_3 \\ & S_2C_4S_5 - C_2C_5 & \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & d_2 \\ & S_4S_5 & \\ 0 & 0 & 1 \\ & 0 & \end{bmatrix}$$

Komponente drugega stolpca v Jacobijevi matriki računamo po nastavku enačb za  $\mathbf{d}_i$  in  $\boldsymbol{\delta}_i$  to je po en. (2.8) in (2.9)

$$\begin{aligned} {}^{T_6}d_{2_x} &= -\{C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6\}\{-C_2d_3\} \\ &\quad + \{S_2(C_4C_5C_6 - S_4S_6) + C_2S_5C_6\}\{S_2d_3\} \end{aligned} \quad (2.25)$$

$$\begin{aligned} {}^{T_6}d_{2_y} &= -\{-C_2(C_4C_5C_6 + S_4C_6) + S_2S_5S_6\}\{-C_2d_3\} \\ &\quad + \{-S_2(C_4C_5C_6 + S_4C_6) - C_2S_5S_6\}\{S_2d_3\} \end{aligned} \quad (2.26)$$

$$\begin{aligned} {}^{T_6}d_{2_z} &= -\{C_2C_4S_5 + S_2C_5\}\{-C_2d_3\} \\ &\quad + \{S_2C_4S_5 - C_2C_5\}\{S_2d_3\} \end{aligned} \quad (2.27)$$

$$\begin{aligned} {}^{T_6}\delta_{2_x} &= S_4C_5C_6 + C_4S_6 \\ {}^{T_6}\delta_{2_y} &= -S_4C_5S_6 + C_4C_6 \\ {}^{T_6}\delta_{2_z} &= S_4S_5 \end{aligned} \quad (2.28)$$

Prve tri enačbe za pozicijske člene okrajšamo, prav tako orientacijske tri, da dobimo drugi stolpec v Jacobijevi matriki  $\partial \mathbf{T}_6 / \partial \theta_2$

$$\partial \mathbf{T}_6 / \partial \theta_2 = \begin{bmatrix} d_3(C_4C_5C_6 - S_4S_6) \\ -d_3(C_4C_5C_6 + S_4S_6) \\ d_3C_4S_5 \\ S_4C_5C_6 + C_4S_6 \\ -S_4C_5S_6 + C_4C_6 \\ S_4S_5 \end{bmatrix} \quad (2.29)$$

## 2.1. IZRAČUN JACOBIJEVE MATRIKE ZA STANFORD MANIPULATOR39

Diferencialna koordinatna transformacija  $\mathbf{T}$  za izračun tretjega stolpca  ${}^2\mathbf{T}_6$  tudi že znana

$${}^2\mathbf{T}_6 = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & C_4S_5 & 0 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_4S_5 & 0 \\ -S_5C_6 & S_5S_6 & C_5 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.30)$$

Tokrat imamo opraviti s translacijskim sklepom, zato bomo morali za izračun  $\mathbf{d}$  in  $\boldsymbol{\delta}$  komponent uporabiti ustreerne enačbe za translacijo. Ker je  $\boldsymbol{\delta}_i = 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$ , ni rotacije in bo tretji stolpec enostaven

$$\begin{aligned} {}^{T_6}d_{i_x} &= n_z & {}^{T_6}d_{i_y} &= o_z & {}^{T_6}d_{i_z} &= a_z \\ \partial\mathbf{T}_6/\partial d_3 &= \begin{bmatrix} -S_5C_6 \\ S_5S_6 \\ C_5 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \quad (2.31)$$

Spoznamo tudi, da za vsak naslednji korak, ko smo bliže zadnjemu segmentu, postajajo enačbe čedalje bolj enostavne in seveda krajše. Naslednji, četrti stolpec, izračunamo, tako da za diferencialno koordinatno transformacijo  $\mathbf{T}$  upoštevamo  ${}^{i-1}\mathbf{T}_6 = {}^{4-1}\mathbf{T}_6 = {}^3\mathbf{T}_6$ , ki smo jo izračunali in je imela obliko

$${}^3\mathbf{T}_6 = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & C_4S_5 & 0 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_4S_5 & 0 \\ -S_5C_6 & S_5S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.32)$$

Velja poudariti, da imajo zadnje tri spremenljivke  $\theta_4$ ,  $\theta_5$  in  $\theta_6$  nalogo zagotoviti orientacijo vrha manipulatorja, zato spremenjanje katerekoli od teh spremenljivk ne vpliva na pozicijo. Od tod bodo tudi diferencialne spremembe nične za  $d_i = 0$  in bo samo  $\delta_i$  imel vrednost. Zapišemo lahko

$$\partial\mathbf{T}_6/\partial\theta_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -S_5C_6 \\ S_5S_6 \\ C_5 \end{bmatrix}. \quad (2.33)$$

Analogno postopamo za peto spremenljivko  $\theta_5$ , za katero je diferencialna koordinatna transformacija enaka  ${}^{i-1}\mathbf{T}_6 = {}^{5-1}\mathbf{T}_6 = {}^4\mathbf{T}_6$  in ima obliko

$${}^4\mathbf{T}_6 = \begin{bmatrix} C_5C_6 & -C_5S_6 & S_5 & 0 \\ S_5C_6 & -S_5S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.34)$$

$$\partial\mathbf{T}_6/\partial\theta_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ S_6 \\ C_6 \\ 0 \end{bmatrix}. \quad (2.35)$$

Nato preostane še spremenljivka  $\theta_6$ , kjer je diferencialna koordinatna transformacija enaka  ${}^{i-1}\mathbf{T}_6 = {}^{6-1}\mathbf{T}_6 = {}^5\mathbf{T}_6$

$${}^5\mathbf{T}_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.36)$$

Zopet napišemo zadnji, šesti stolpec za Jacobijevo matriko

$$\partial\mathbf{T}_6/\partial\theta_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.37)$$

Tako so določeni vsi stolpci za Jacobijevo matriko.

Primer: Imamo Stanfordski manipulator, ki je v položaju, kot ga podaja  $\mathbf{T}_6$  matrika

$$\mathbf{T}_6 = \begin{bmatrix} 0 & 1 & 0 & 20 \\ 1 & 0 & 0 & 6 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Ta lega ustreza naslednjemu stanju notranjih koordinat

## 2.1. IZRAČUN JACOBIJEVE MATRIKE ZA STANFORD MANIPULATOR41

koordinata	vrednost	$\sin \theta$	$\cos \theta$
$\theta_1$	$0^\circ$	0	1
$\theta_2$	$90^\circ$	1	0
$d_3$	20 cm		
$\theta_4$	$0^\circ$	0	1
$\theta_5$	$90^\circ$	1	0
$\theta_6$	$90^\circ$	1	0

Tabela 2.1:

Za  $d_2$  upoštevamo podano vrednost 6 cm.

Izračunajte Jacobijevo matriko in določite diferencialni premik  $d\mathbf{T}_6$  za naslednje diferencialne premike koordinat

$$d\mathbf{q}_i = \begin{bmatrix} 0, 1 \\ -0, 1 \\ 2, 0 \\ 0, 1 \\ 0, 1 \\ 0, 1 \end{bmatrix}.$$

Rešitev:

Najprej izračunamo stolpce Jacobijeve matrike. Za ta namen uporabimo enačbe za posamezne stolpce v matriki za izračun  $\partial\mathbf{T}_6/\partial\theta_i$ . Izračun poteka tako, da v člene po stolpcih vstavljamo vrednosti iz tabele. Npr. za zadnji člen v prvem stolpcu imamo  $-S_2C_4S_5 + C_2C_5 = -1 \cdot 1 \cdot 1 + 0 \cdot 0 = -1$ . Če to napravimo za vse elemente vseh šestih stolpcev, lahko napišemo Jacobijevo matriko

$$\frac{\partial\mathbf{T}_6}{\partial q_i} = \begin{bmatrix} 20, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 \\ -6, 0 & 0, 0 & 1, 0 & 0, 0 & 0, 0 & 0, 0 \\ 0, 0 & 20, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 \\ 0, 0 & 1, 0 & 0, 0 & 0, 0 & 1, 0 & 0, 0 \\ 0, 0 & 0, 0 & 0, 0 & 1, 0 & 0, 0 & 0, 0 \\ -1, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 & 1, 0 \end{bmatrix} = \mathbf{J}.$$

Diferencialna sprememba v poziciji in orientaciji je dana z

$$\frac{\partial\mathbf{T}_6}{\partial q_i} \cdot d\mathbf{q}_i = \mathbf{J} \cdot d\mathbf{q}_i = \begin{bmatrix} {}^{T_6}\mathbf{d} \\ {}^{T_6}\boldsymbol{\delta} \end{bmatrix},$$

zato pišemo

$$\begin{bmatrix} 2, 0 \\ 1, 4 \\ -2, 0 \\ 0, 0 \\ 0, 1 \\ 0, 0 \end{bmatrix} = \begin{bmatrix} 20, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 \\ -6, 0 & 0, 0 & 1, 0 & 0, 0 & 0, 0 & 0, 0 \\ 0, 0 & 20, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 \\ 0, 0 & 1, 0 & 0, 0 & 0, 0 & 1, 0 & 0, 0 \\ 0, 0 & 0, 0 & 0, 0 & 1, 0 & 0, 0 & 0, 0 \\ -1, 0 & 0, 0 & 0, 0 & 0, 0 & 0, 0 & 1, 0 \end{bmatrix} \begin{bmatrix} 0, 1 \\ -0, 1 \\ 2, 0 \\ 0, 1 \\ 0, 1 \\ 0, 1 \end{bmatrix}$$

oz. od tod napišemo  ${}^T_6\boldsymbol{d}$  in  ${}^T_6\boldsymbol{\delta}$ .

$$\begin{aligned} {}^T_6\boldsymbol{d} &= 2, 0\boldsymbol{i} + 1, 4\boldsymbol{j} - 2, 0\boldsymbol{k} \\ {}^T_6\boldsymbol{\delta} &= 0, 0\boldsymbol{i} + 0, 1\boldsymbol{j} + 0, 0\boldsymbol{k}. \end{aligned}$$

Vse to lahko tudi narišemo. S pomočjo podatkov za  $\mathbf{T}_6$  in kotov v tabeli narišemo manipulator in seveda pozicijo in orientacijo prijemala. S pomočjo  ${}^T_6\boldsymbol{d}$  in  ${}^T_6\boldsymbol{\delta}$  lahko narišemo tudi novo lego.

Ob poznavanju vseh podatkov lahko sestavimo tudi matriko  ${}^T_6\boldsymbol{\Delta}$  in to tako, da vrednosti  ${}^T_6d_x$ ,  ${}^T_6d_y$ ,  ${}^T_6d_z$ ,  ${}^T_6\delta_x$ ,  ${}^T_6\delta_y$ ,  ${}^T_6\delta_z$  vstavljamo v enačbo za  ${}^T_6\boldsymbol{\Delta}$

$$\begin{aligned} {}^T_6\boldsymbol{\Delta} &= \begin{bmatrix} 0 & -{}^T_6\delta_z & {}^T_6\delta_y & {}^T_6d_x \\ {}^T_6\delta_z & 0 & -{}^T_6\delta_x & {}^T_6d_y \\ -{}^T_6\delta_y & {}^T_6\delta_x & 0 & {}^T_6d_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{torej} \\ {}^T_6\boldsymbol{\Delta} &= \begin{bmatrix} 0, 0 & 0, 0 & 0, 1 & 2, 0 \\ 0, 0 & 0, 0 & 0, 0 & 1, 4 \\ -0, 1 & 0, 0 & 0, 0 & -2, 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Ker vemo, da je  $\mathbf{T}_6 \cdot {}^T_6\boldsymbol{\Delta} = d\mathbf{T}_6$ , lahko izračunamo  $d\mathbf{T}_6$  (lahko bi ga dobili tudi iz nove lege  $\mathbf{T}'_6$ , od katere bi odšteli  $\mathbf{T}_6$ ).

$$d\mathbf{T}_6 = \begin{bmatrix} 0, 0 & 1, 0 & 0, 0 & 20, 0 \\ 1, 0 & 0, 0 & 0, 0 & 6, 0 \\ 0, 0 & 0, 0 & -1, 0 & 0, 0 \\ 0, 0 & 0, 0 & 0, 0 & 1, 0 \end{bmatrix} \begin{bmatrix} 0, 0 & 0, 0 & 0, 1 & 2, 0 \\ 0, 0 & 0, 0 & 0, 0 & 1, 4 \\ -0, 1 & 0, 0 & 0, 0 & -2, 0 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 \end{bmatrix} = \begin{bmatrix} 0, 0 & 0, 0 & 0, 0 & 1, 4 \\ 0, 0 & 0, 0 & 0, 1 & 2, 0 \\ 0, 1 & 0, 0 & 0, 0 & 2, 0 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 \end{bmatrix}$$

## 2.2 GEOMETRIJSKA IN ANALITIČNA JACOBIJEVA MATRIKA

Vektor diferencialnega gibanja  $\mathbf{D} = \begin{bmatrix} d \\ \delta \end{bmatrix}$  podaja diferencialne pomike in rotacije v zunanjih koordinatah, torej

$$\mathbf{D} = [d_x, d_y, d_z, \delta_x, \delta_y, \delta_z]^T \quad (2.38)$$

in

$${}^T_6 \mathbf{D} = [{}^T_6 d_x, {}^T_6 d_y, {}^T_6 d_z, {}^T_6 \delta_x, {}^T_6 \delta_y, {}^T_6 \delta_z]^T. \quad (2.39)$$

Analogno je vektor diferencialnih pomikov sklepnih spremenljivk

$\mathbf{D}_\Theta = \begin{bmatrix} d\Theta_p \\ d\Theta_o \end{bmatrix}$  definiran kot

$$\mathbf{D}_\Theta = [dq_1, dq_2, dq_3, dq_4, dq_5, dq_6]^T. \quad (2.40)$$

Povezava med  ${}^T_6 \mathbf{D}$  in  $\mathbf{D}_\Theta$  je določena z in jo opisuje zgoraj prikazana analitična Jakobijeva matrika  $\mathbf{J}_A$

$${}^T_6 \mathbf{D} = \mathbf{J}_A \cdot \mathbf{D}_\Theta \quad (2.41)$$

medtem ko povezavo med  $\mathbf{D}$  in  $\mathbf{D}_\Theta$  določa geometrijska Jakobijeva matrika (izpeljava je v [Schiavocco L., 2002])

$$\mathbf{D} = \mathbf{J}_G \cdot \mathbf{D}_\Theta. \quad (2.42)$$

Analogni inverzni problem je definiran kot

$$\mathbf{D}_\Theta = \mathbf{J}_G^{-1} \cdot \mathbf{D} \quad (2.43)$$

ter

$$\mathbf{D}_\Theta = \mathbf{J}_A^{-1} \cdot {}^T_6 \mathbf{D}. \quad (2.44)$$

$\mathbf{J}^{-1}$  je inverzna matrika  $6 \times 6$ , njen izračun pa je zapleten.

Nadalje velja, da zapis v obliki  $\mathbf{D} = \mathbf{J}_G \cdot \mathbf{D}_\Theta$  podaja diferencialne premike oz. s tem povezane hitrosti. Če definiramo hitrosti kot

$$\dot{\mathbf{X}} = \left( \frac{1}{\Delta t_{sam}} \cdot \right) \mathbf{D}. \quad (2.45)$$

Tu je  $\dot{\mathbf{X}}$  hitrost izražena v kartezijevih koordinatah,  $\Delta t_{sam}$  pa je skalar, čas vzorčnega intervala. Definirajmo še elemente v vektorju hitrosti

$$\dot{\mathbf{X}} = [\dot{x}, \dot{y}, \dot{z}, \dot{\varphi}_x, \dot{\varphi}_y, \dot{\varphi}_z]^T. \quad (2.46)$$

Pri geometrijskem  $\mathbf{J}_G$  je mogoče prepoznati prispevek vsakega sklepa posebej k skupni hitrosti.

Za definirani  $\Delta t_{samp} = \text{konst}$  lahko pišemo ob upoštevanju  $\mathbf{X}$  izraz

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi}_x \\ \dot{\varphi}_y \\ \dot{\varphi}_z \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & J_{66} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix}. \quad (2.47)$$

Spoznamo, da velja

$$\dot{x} = J_{11}\dot{\theta}_1 + J_{12}\dot{\theta}_2 + \dots + J_{16}\dot{\theta}_6, \quad (2.48)$$

torej so elementi v  $\mathbf{J}$  matriki parcialni odvodi

$$J_{11} = \frac{\partial \mathbf{X}}{\partial \theta_1} \quad J_{12} = \frac{\partial \mathbf{X}}{\partial \theta_2} \quad \dots \quad (2.49)$$

Tako smo pokazali, da je možno iz enačbe  ${}^{T_6}\mathbf{D} = \mathbf{J}_A \cdot \mathbf{D}_\Theta$  ob definiranem stanju izraziti in najti povezave med hitrostmi notranjih koordinat in hitrostmi v zunanjih koordinatah.

$$\dot{\mathbf{X}} = \mathbf{J} \cdot \dot{\mathbf{D}}_\Theta \quad (2.50)$$

## 2.3 IZRAČUN JAKOBIJEVE Matrike

Upoštevajmo serijski manipulator z  $n$  segmenti, spremeljivke v sklepih so označene kot  $q_1, \dots, q_n$ . Celotna transformacija vrha glede na osnovni k.s. je podana z

$${}^0\mathbf{T}_n(\mathbf{q}) = \begin{bmatrix} {}^0\mathbf{R}_n(\mathbf{q}) & {}^0\mathbf{o}_n(\mathbf{q}) \\ 0 & 1 \end{bmatrix}. \quad (2.51)$$

Pri tem vektor  $\mathbf{q} = [q_1, \dots, q_n]^T$  predstavlja spremenljivke v sklepih. Med gibanjem robota se spreminja tako sklepne spremenljivke  $q_i$  in posledično tudi pozicija vrha  ${}^0\mathbf{o}_n$  in orientacija  ${}^0\mathbf{R}_n$ , vse kot funkcija časa. Namen tega podoglavlja je povezati linearne in kotne hitrosti vrha robota (kot posledica) z vektorjem sklepnih spremenljivk  $\dot{\mathbf{q}}(t)$  (kot vzrokom). Naj zapis

$$\mathbf{S}({}^0\boldsymbol{\omega}_n) = {}^0\dot{\mathbf{R}}_n({}^0\mathbf{R}_n)^T \quad (2.52)$$

definira kotno hitrost vektorja  ${}^0\boldsymbol{\omega}_n$  za vrh robota in naj

$${}^0\boldsymbol{v}_n = {}^0\dot{\boldsymbol{o}}_n \quad (2.53)$$

predstavlja linearo hitrost vrha robota. Iščemo zapise v obliki

$$\begin{aligned} {}^0\boldsymbol{v}_n &= \mathbf{J}_v \dot{\boldsymbol{q}} \\ {}^0\boldsymbol{\omega}_n &= \mathbf{J}_\omega \dot{\boldsymbol{q}}, \end{aligned} \quad (2.54)$$

kjer sta  $\mathbf{J}_v$  in  $\mathbf{J}_\omega$  matriki dimenzij  $3 \times n$ . Obe zgornji enačbi lahko skupaj zapišemo kot

$$\boldsymbol{\xi} = \mathbf{J} \dot{\boldsymbol{q}}, \quad (2.55)$$

kjer sta  $\boldsymbol{\xi}$  ( $\boldsymbol{\xi} = \dot{\mathbf{X}}$ ) in  $\mathbf{J}$  podana kot

$$\boldsymbol{\xi} = \begin{bmatrix} {}^0\boldsymbol{v}_n \\ {}^0\boldsymbol{\omega}_n \end{bmatrix} \quad \text{in} \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_\omega \end{bmatrix} \quad (2.56)$$

Vektor  $\boldsymbol{\xi}$  tudi poimenujemo hitrost telesa. Poudariti velja, da ta vektor hitrosti ni časovni odvod pozicijske spremenljivke, saj vektor kotne hitrosti ni odvod nobene od spremenljivk odvisnih od časa. Matriko  $\mathbf{J}$  poimenujemo **Jakobi manipulatorja** ali kratko **Jakobi**.  $\mathbf{J}$  je matrika dimenzij  $6 \times n$ , pri čemer je  $n$  število segmentov. V nadaljevanju je izveden izraz za Jakobi kateregakoli manipulatorja.

### 2.3.1 Kotna hitrost

Z enačbo je mogoče pokazati seštevanje kotnih hitrosti kot neodvisnih vektorjev v primeru njihovega definiranja relativno glede na skupen k.s.. Lahko torej določimo kotno hitrost vrha manipulatorja relativno glede na osnovni k.s. s seštevkom vseh kotnih hitrosti, če so vse kotne hitrosti nastale zaradi gibanja posameznih sklepov izražene glede na osnovni k.s..

Če je  $i$  sklep rotacijski, potem je sklepna spremenljivka  $\boldsymbol{q}_i$  v sklepu  $i$  enaka  $\theta_i$ , os rotacije pa je  $\boldsymbol{z}_{i-1}$ . Oznaka  ${}^{i-1}\boldsymbol{\omega}_i$  pomeni kotno hitrost sklepa  $i$ , kot posledico rotacije sklepa  $i$ , izraženo relativno glede na k.s.  $o_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ . Ta kotna hitrost je izražena relativno glede na k.s.  $i - 1$  kot

$${}^{i-1}\boldsymbol{\omega}_i = \dot{\boldsymbol{q}}_i {}^{i-1}\boldsymbol{z}_{i-1} = \dot{\boldsymbol{q}}_i \mathbf{k} \quad (2.57)$$

v katerem  $\mathbf{k}$  predstavlja enotin vektor  $[0, 0, 1]^T$ .

Če je  $i$  sklep translacijski, potem je premikanje k.s.  $i$  relativno glede na  $i - 1$  translacijsko in tako

$${}^{i-1}\boldsymbol{\omega}_i = 0. \quad (2.58)$$

Ko je  $i$  sklep translacijski, potem kotna hitrost vrha ni odvisna od  $\theta_i$ , je pa odvisna od  $\dot{\mathbf{q}}_i$ , saj se ta spreminja translacijsko  $\mathbf{d}_i$ .

Skupna kotna hitrost vrha manipulatorja, označena kot  ${}^0\boldsymbol{\omega}_n$ , v osnovnem k.s. je določena z enačbo

$${}^0\boldsymbol{\omega}_n = \rho_1 \dot{\mathbf{q}}_1 \mathbf{k} + \rho_2 \dot{\mathbf{q}}_2 {}^0\mathbf{R}_1 \mathbf{k} + \dots + \rho_n \dot{\mathbf{q}}_n {}^0\mathbf{R}_{n-1} \mathbf{k} = \sum_{i=1}^n \rho_i \dot{\mathbf{q}}_i {}^0\mathbf{z}_{i-1} \quad (2.59)$$

v kateri je  $\rho_i$  enak 1 v primeru rotacijskega sklepa in 0, če je sklep  $i$  translacijski, saj je

$${}^0\mathbf{z}_{i-1} = {}^0\mathbf{R}_{i-1} \mathbf{k}. \quad (2.60)$$

Seveda je  ${}^0\mathbf{z}_0 = [0, 0, 1]^T$ .

Spodnja polovica Jakobija  $\mathbf{J}_\omega$ , povezana z orientacijo je torej podana kot

$$\mathbf{J}_\omega = [\rho_1 \mathbf{z}_0, \dots, \rho_n \mathbf{z}_{n-1}]. \quad (2.61)$$

V tej enačbi smo izpustili superskripte pri enotinih vektorjih okrog  $z$ -osi, saj so ti podani glede na osnovni k.s..

### 2.3.2 Translacijska hitrost

Linearna, translacijska hitrost vrha manipulatorja je označena kot  ${}^0\dot{\mathbf{o}}_n$ . Z zaporednim odvajanjem pridemo do

$${}^0\dot{\mathbf{o}}_n = \sum_{i=1}^n \frac{\partial {}^0\mathbf{o}_n}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_i. \quad (2.62)$$

Iz tega je mogoče zapisati za  $i$  stolpec podmatrike  $\mathbf{J}_v$  označen kot  $\mathbf{J}_{v_i}$  v obliki

$$\mathbf{J}_{v_i} = \frac{\partial {}^0\mathbf{o}_n}{\partial \mathbf{q}_i}. \quad (2.63)$$

Izraz pravzaprav predstavlja samo linearno hitrost vrha manipulatorja, če bi bil  $\dot{\mathbf{q}}_i$  enak ena, ostali  $\dot{\mathbf{q}}_j$  pa enaki nič. Z drugimi besedami,  $i$ -ti stolpec Jakobijeve matrike dobimo s fiksiranjem vseh ostalih sklepov in aktivacijo samo  $i$  z enotino hitrostjo. To opažanje vodi do preproste in intitivne izpeljave za linearno hitrost Jakobija, kot je pokazano v nadaljevanju. Sledita dva primera, ločeno za translacijske in rotacijske sklepe.

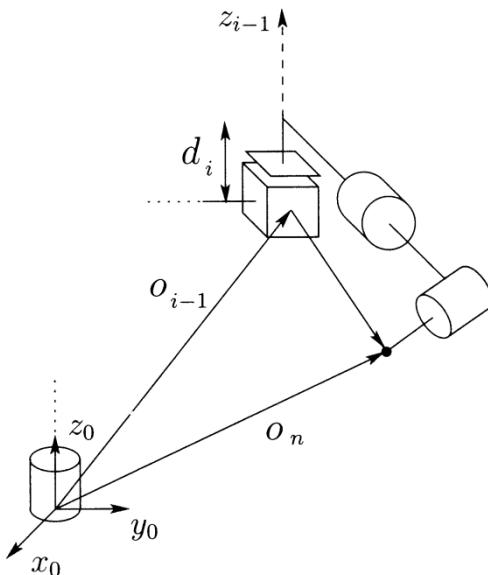
### 2.3.3 Translacijski sklepi

Slika 2.4 prikazuje primer z vsemi sklepi fiksiranimi, razen enega translacijskega sklepa. Ker je sklep  $i$  translacijski, povzroča na vrhu manipulatorja čisto translacijo. Smer translacije je vzporedna osi  $\mathbf{z}_{i-1}$ , velikost translacije pa je za  $\mathbf{d}_i$ , kjer je  $\mathbf{d}_i$  spremenljivka znana iz DH. Glede na orientacijo osnovnega k.s. imamo

$${}^0\dot{\mathbf{o}}_n = \dot{\mathbf{d}}_i {}^0\mathbf{R}_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \dot{\mathbf{d}}_i {}^0\mathbf{z}_{i-1}, \quad (2.64)$$

kjer je  $\mathbf{d}_i$  spremenljivka v sklepu za translacijski sklep  $i$ . Po brisanju super-skriptov imamo torej v primeru translacijskih sklepov izraz

$$\mathbf{J}_{v_i} = \mathbf{z}_{i-1}. \quad (2.65)$$



Slika 2.4: Premik vrha robota zaradi translacijskega sklepa  $i$ .

### 2.3.4 Rotacijski sklepi

Slika 2.5 prikazuje primer z vsemi sklepi fiksiranimi, razen enega rotacijskega sklepa. Ker je sklep  $i$  rotacijski, to pomeni  $q_i = \theta_i$ . Skladno s sliko 2.5 so vsi sklepi fiksirani, razen sklepa  $i$ . Translacijska hitrost vrha manipulatorja ima obliko  $\boldsymbol{\omega} \times \mathbf{r}$ , kjer je

$$\boldsymbol{\omega} = \dot{\theta}_i \mathbf{z}_{i-1} \quad (2.66)$$

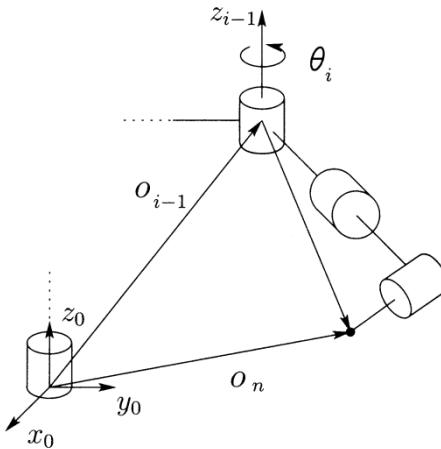
in

$$\mathbf{r} = \mathbf{o}_n - \mathbf{o}_{i-1}. \quad (2.67)$$

Po vstavitevi obeh enačb v en skupni zapis in zapisu glede na osnovni k.s. dobimo za rotacijske sklepe

$$\mathbf{J}_{v_i} = \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}), \quad (2.68)$$

kjer so izpuščeni superskripti 0, saj so veličine izražene glede na osnovni k.s..



Slika 2.5: Premik vrha robota zaradi rotacijskega sklepa  $i$ .

### 2.3.5 Kombinacija Jakobi členov za translacijsko in kotonno hitrost

Glede na izpeljavo zgoraj, je zgornja polovica Jakobijeve matrike  $\mathbf{J}_v$  podana kot

$$\mathbf{J}_v = [\mathbf{J}_{v_1}, \dots, \mathbf{J}_{v_n}], \quad (2.69)$$

kjer je  $i$ -ti stolpec v  $\mathbf{J}_{v_i}$  enak

$$\mathbf{J}_{v_i} = \begin{cases} \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}) & \text{za rotacijski sklep } i \\ \mathbf{z}_{i-1} & \text{za translacijski sklep } i. \end{cases} \quad (2.70)$$

Spodnja polovica Jakobijeve matrike je podana kot

$$\mathbf{J}_\omega = [\mathbf{J}_{\omega_1}, \dots, \mathbf{J}_{\omega_n}], \quad (2.71)$$

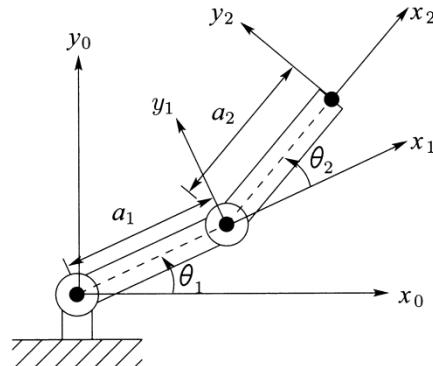
kjer je  $i$ -ti stolpec v  $\mathbf{J}_{\omega_i}$  enak

$$\mathbf{J}_{\omega_i} = \begin{cases} \mathbf{z}_{i-1} & \text{za rotacijski sklep } i \\ 0 & \text{za translacijski sklep } i. \end{cases} \quad (2.72)$$

Zgornje enačbe omogočajo izračun Jakobijeve matrike za katerikoli manipulator na enostaven način, saj so po končanem izračunu direktne kinematike na voljo vse potrebne spremenljivke. Pravzaprav so za izračun členov Jakobijeve matrike potrebni enotni vektorji  $\mathbf{z}_i$  in koordinate izhodišč  $\mathbf{o}_1, \dots, \mathbf{o}_n$ . Koordinate izhodišča za  $\mathbf{z}_i$  podane glede na osnovni k.s. so definirane s prvimi tremi elementi v tretjem stolpcu matrike  ${}^0\mathbf{T}_i$ . Vrednosti  $\mathbf{o}_i$  so podane v prvih treh elementih četrtega stolpca v matriki  ${}^0\mathbf{T}_i$ . Torej sta potrebna le tretji in četrtri stolpec matrike  $\mathbf{T}$ , da bi skladno s podanimi enačbami lahko izračunali člene Jakobijeve matrike.

Prikazani način izračuna je ustrezen ne samo za določanje hitrosti vrha manipulatorja, ampak tudi za izračun hitrosti drugih točk na manipulatorju, npr. za izračun hitrosti masnega središča segmenta.

### 2.3.6 Primer: 2DOF, 2R ravninski manipulator



Slika 2.6: 2DOF, 2R ravninski manipulator  $z$ -osi vse gledajo iz lista proti nam in niso narisane.

Obravnavajmo dvosegmentni ravninski manipulator, oba sklepa naj bosta rotacijska. To pomeni Jakobijevu matriko dimenzij  $6 \times 2$  v obliki

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{o}_2 - \mathbf{o}_0) & \mathbf{z}_1 \times (\mathbf{o}_2 - \mathbf{o}_1) \\ \mathbf{z}_0 & \mathbf{z}_1 \end{bmatrix} \quad (2.73)$$

Glede na vrednosti v matriki  $\mathbf{T}$  imajo različni členi v gornjem zapisu obliko

$$\mathbf{o}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{o}_1 = \begin{bmatrix} a_1 C_1 \\ a_1 S_1 \\ 0 \end{bmatrix} \quad \mathbf{o}_2 = \begin{bmatrix} a_1 C_1 + a_2 C_{12} \\ a_1 S_1 + a_2 S_{12} \\ 0 \end{bmatrix}. \quad (2.74)$$

Po vstavitevi teh vrednosti v matriko in matričnem množenju dobimo

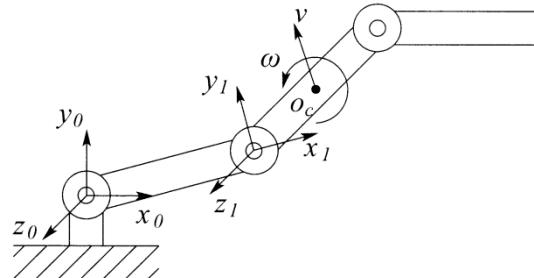
$$\boldsymbol{z}_0 = \boldsymbol{z}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.75)$$

Vrednosti členov v matriki je mogoče primerjati z znanima enačbama

$$\boldsymbol{J} = \begin{bmatrix} -a_1 S_1 - a_2 S_{12} & -a_2 S_{12} \\ a_1 C_1 + a_2 C_{12} & a_2 C_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}. \quad (2.76)$$

Prvi dve vrstici v matriki sta ravno  $2 \times 2$  Jakobijeva matrika iz enačbe in podajata translacijsko hitrost izhodišča  $\boldsymbol{o}_2$  relativno glede na bazo. Tretja vrstica v Jakobijevi matriki je translacijska hitrost v smeri  $\boldsymbol{z}_0$ , ki je seveda v tem primeru zmeraj enaka nič. Zadnje tri vrstice v matriki predstavljajo kotno hitrost k.s. vrha, ki je preprosto rotacija okrog vertikalne osi v vrednosti  $\dot{\theta}_1 + \dot{\theta}_2$ .

### 2.3.7 Primer: Jakobijeva matrika za poljubno točko na segmentu



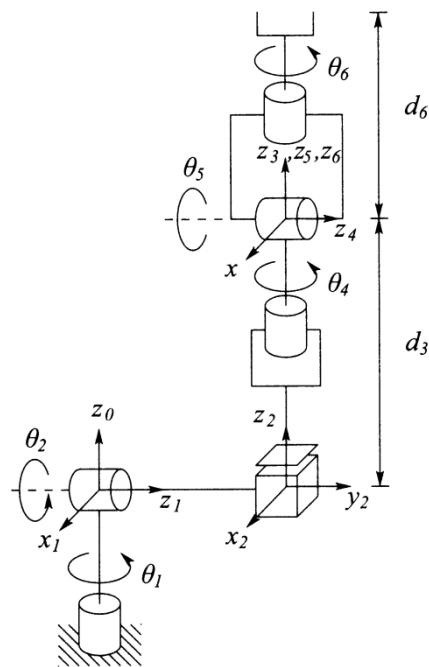
Slika 2.7: Računanje hitrosti drugega segmenta na trisegmentnem ravninskem robotu.

Obravnavajmo trisegmentni ravninski manipulator, skladno s sliko 2.7. Ena izmed naših želja je npr. izračun translacijske hitrosti  $v$  in kotne hitrosti  $\omega$  za center segmenta, za masno središče. V tem primeru imamo zapis za Jakobijovo matriko v obliki

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{z}_0 \times (\boldsymbol{o}_c - \boldsymbol{o}_0) & \boldsymbol{z}_1 \times (\boldsymbol{o}_c - \boldsymbol{o}_1) & 0 \\ \boldsymbol{z}_0 & \boldsymbol{z}_1 & 0 \end{bmatrix}. \quad (2.77)$$

To je pravzaprav že znana Jakobijeva matrika, kjer je  $\mathbf{o}_c$  na mestu  $\mathbf{o}_n$ . Ne smemo spregledati tretjega stolpca matrike, z členi enakimi nič, saj je hitrost drugega segmenta neodvisna od gibanja tretjega segmenta. Kinematični vplivi tretjega na drugi segment so enaki nič, medtem ko obstaja vpliv sl in navorov tretjega segmenta na drugega. Vektor  $\mathbf{o}_c$  je pri tem mogoče izračunati s pomočjo ustreznih  $\mathbf{T}$  matrik.

### 2.3.8 Primer: Stanford manipulator



Slika 2.8: Stanford manipulator.

Obravnajmo Stanford manipulator (slika 2.8), skupaj s pripadajočo DH transformacijo in ustrezno nameščenimi k.s.. Tretji sklep je translacijski, prav tako velja enakost  $\mathbf{o}_3 = \mathbf{o}_4 = \mathbf{o}_5$  kot posledica sferičnega zapestja in namestitve k.s.. Z uporabo črke  $\mathbf{o}$  za to skupno izhodišče postanejo stolpci

v Jakobijevi matriki enaki

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{o}_6 - \mathbf{o}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} \quad i = 1, 2 \quad (2.78)$$

$$\mathbf{J}_3 = \begin{bmatrix} \mathbf{z}_2 \\ 0 \end{bmatrix} \quad (2.79)$$

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{o}_6 - \mathbf{o}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} \quad i = 4, 5, 6. \quad (2.80)$$

V nadaljevanju izračuna uporabimo vrednosti matrik A

$$\mathbf{A}_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

in  $\mathbf{T}$  matrik izračunanih kot produkt matrik  $\mathbf{A}$ . Najprej je  $\mathbf{o}_j$  podan s prvimi tremi členi v zadnjem stolpcu matrike  ${}^0\mathbf{T}_i = \mathbf{A}_1 \cdot \dots \cdot \mathbf{A}_j$  z  $\mathbf{o}_0 = [0, 0, 0]^T = \mathbf{o}_1$ . Vektor  $\mathbf{z}_j$  je podan kot  $\mathbf{z}_j = {}^0\mathbf{R}_j \mathbf{k}$ , kjer  ${}^0\mathbf{R}_j$  predstavlja rotacijski del matrike  ${}^0\mathbf{T}_j$ . Torej je za izračun Jakobijske matrike potrebno izračunati le matrike  ${}^0\mathbf{T}_j$ . Ko opravimo te preračune dobimo

$$\mathbf{o}_6 = \begin{bmatrix} C_1 S_2 d_3 - S_1 d_2 + d_6(C_1 C_2 C_4 S_5 + C_1 C_5 S_2 - S_1 S_4 S_5) \\ S_1 S_2 d_3 - C_1 d_2 + d_6(C_1 S_4 S_5 + C_2 C_4 S_1 S_5 - C_5 S_1 S_2) \\ C_2 d_3 + d_6(C_2 C_5 - C_4 S_2 S_5) \end{bmatrix} \quad (2.81)$$

in

$$\mathbf{o}_3 = \begin{bmatrix} C_1 S_2 d_3 - S_1 d_2 \\ S_1 S_2 d_3 + C_1 d_2 \\ C_2 d_3 \end{bmatrix}. \quad (2.82)$$

Pri tem so  $\mathbf{z}_i$  podani kot

$$\mathbf{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{z}_1 = \begin{bmatrix} -S_1 \\ C_1 \\ 0 \end{bmatrix} \quad (2.83)$$

$$\mathbf{z}_2 = \begin{bmatrix} C_1 S_2 \\ S_1 S_2 \\ C_2 \end{bmatrix} \quad \mathbf{z}_3 = \begin{bmatrix} C_1 S_2 \\ S_1 S_2 \\ C_2 \end{bmatrix} \quad (2.84)$$

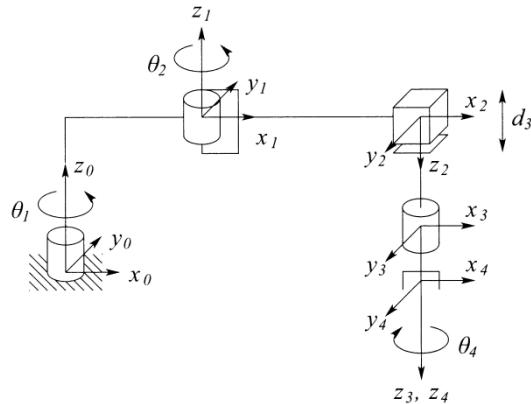
$$\mathbf{z}_4 = \begin{bmatrix} -C_1 C_2 S_4 - S_1 C_4 \\ -S_1 C_2 S_4 - C_1 C_4 \\ S_2 S_4 \end{bmatrix} \quad (2.85)$$

$$\mathbf{z}_5 = \begin{bmatrix} C_1 C_2 C_4 S_5 - S_1 S_4 S_5 + C_1 S_2 C_5 \\ S_1 C_2 C_4 S_5 + C_1 S_4 S_5 + S_1 S_2 C_5 \\ -S_2 C_4 S_5 + C_2 C_5 \end{bmatrix} \quad (2.86)$$

Jakobijevu matriko za Stanford manipulator sedaj dobimo s kombinacijo teh izrazov v skladu s podanimi formulami.

### 2.3.9 Primer: SCARA manipulator

Tukaj bomo izračunali Jakobijevu matriko za SCARA manipulator (slika 2.9)



Slika 2.9: SCARA manipulator.

Jakobijeva matrika za SCARA manipulator je dimenzijs  $6 \times 4$ , saj ima SCARA manipulator le štiri prostostne stopnje. Podobno kot prej moramo določiti le vrednosti matrik  ${}^0\mathbf{T}_i = \mathbf{A}_1 \cdot \dots \cdot \mathbf{A}_j$  pri čemer so vrednosti  $\mathbf{A}$

matrik podane kot

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} C_2 & S_2 & 0 & a_2 C_2 \\ S_2 & -C_2 & 0 & a_2 S_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{A}_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \mathbf{A}_4 &= \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Sklepi 1, 2 in 4 so rotacijski, tretji sklep pa je translacijski.  $\mathbf{o}_4 - \mathbf{o}_3$  je vzporeden z  $\mathbf{z}_3$  (torej velja  $\mathbf{z}_3 \times [\mathbf{o}_4 - \mathbf{o}_3] = 0$ ), zato ima Jakobijeva matrika obliko

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{o}_4 - \mathbf{o}_0) & \mathbf{z}_1 \times (\mathbf{o}_4 - \mathbf{o}_1) & \mathbf{z}_2 & 0 \\ \mathbf{z}_0 & \mathbf{z}_1 & 0 & \mathbf{z}_3 \end{bmatrix}. \quad (2.87)$$

Izhodišča k.s., definirana skladno z D-H so enaka

$$\mathbf{o}_1 = \begin{bmatrix} a_1 C_1 \\ a_1 S_1 \\ 0 \end{bmatrix} \quad \mathbf{o}_2 = \begin{bmatrix} a_1 C_1 + a_2 C_{12} \\ a_1 S_1 + a_2 S_{12} \\ 0 \end{bmatrix} \quad (2.88)$$

$$\mathbf{o}_4 = \begin{bmatrix} a_1 C_1 + a_2 C_{12} \\ a_1 S_1 + a_2 S_{12} \\ -d_3 - d_4 \end{bmatrix} \quad (2.89)$$

Velja tudi  $\mathbf{z}_0 = \mathbf{z}_1 = \mathbf{k}$  in  $\mathbf{z}_2 = \mathbf{z}_3 = -\mathbf{k}$ . Torej je Jakobijeva matrika za SCARA manipulator enaka

$$\mathbf{J} = \begin{bmatrix} -a_1 S_1 - a_2 S_{12} & -a_2 S_{12} & 0 & 0 \\ a_1 C_1 + a_2 C_{12} & a_2 C_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix}. \quad (2.90)$$

## 2.4 HITROST ORODJA

Večino robotskih nalog zahteva pritrditev orodja na vrh manipulatorja. V takšnih primerih je smiselno uporabljati k.s. orodja in hitrost k.s. orodja, podano glede na hitrost k.s. vrha robota. Predpostavimo, da je orodje fiksno

pritrjeno na vrh robota, pri tem je relacija med k.s. vrha in k.s. orodja podana s konstantno homogeno matriko

$${}^6\mathbf{T}_{tool} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ 0 & 1 \end{bmatrix}. \quad (2.91)$$

Predpostavimo, da je hitrost za k.s. orodja podana in izražena relativno glede na k.s. vrha, ta hitrost je  ${}^6\xi_6$ . V tem podpoglavlju bomo zapisali hitrost za k.s. orodja, zapisano relativno glede na k.s. orodja, torej  ${}^{tool}\xi_{tool}$ .

Ker sta omenjena k.s. pritrjena fiksno, je kotna hitrost k.s. orodja enaka kotni hitrosti za k.s. vrha. V to se lahko prepričamo z izračunom kotnih hitrosti za vsak k.s. preko odvajanja ustreznih rotacijskih matrik. Ker je  $\mathbf{R}$  konstantna in  ${}^0\mathbf{R}_{tool} = {}^0\mathbf{R}_6 \cdot \mathbf{R}$  imamo

$$\begin{aligned} {}^0\mathbf{R}_{tool} &= {}^0\mathbf{R}_6 \cdot \mathbf{R} \\ \mathbf{S}({}^0\boldsymbol{\omega}_{tool}) \cdot {}^0\mathbf{R}_{tool} &= \mathbf{S}({}^0\boldsymbol{\omega}_6) \cdot {}^0\mathbf{R}_6 \cdot \mathbf{R} \\ \mathbf{S}({}^0\boldsymbol{\omega}_{tool}) &= \mathbf{S}({}^0\boldsymbol{\omega}_6). \end{aligned} \quad (2.92)$$

Torej je  $\boldsymbol{\omega}_{tool} = \boldsymbol{\omega}_6$ . Za izračun hitrosti za k.s. orodja relativno na k.s. orodja uporabimo rotacijsko transformacijo

$${}^{tool}\boldsymbol{\omega}_{tool} = {}^{tool}\boldsymbol{\omega}_6 = {}^{T_6}\mathbf{R}\boldsymbol{\omega}_6. \quad (2.93)$$

Če se k.s. vrha premika s hitrostjo  $\xi = [\mathbf{v}_6^T, \boldsymbol{\omega}_6^T]^T$ , potem je translacijska hitrost izhodišča za k.s. orodja, ki je fiksno pritrjen na k.s. vrha enaka

$$\mathbf{v}_{tool} = \mathbf{v}_6 + \boldsymbol{\omega}_6 \times \mathbf{r}. \quad (2.94)$$

$\mathbf{r}$  je vektor od izhodišča k.s. vrha do izhodišča k.s. orodja. V enačbi (2.91) podaja  $\mathbf{d}$  koordinate izhodišča za k.s. orodja glede na k.s. vrha. Torej lahko izrazimo  $\mathbf{d}$  in koordinate relativno glede na k.s. orodja kot  ${}^{tool}\mathbf{r} = \mathbf{R}^T \mathbf{d}$ . Zapišemo torej  $\boldsymbol{\omega}_6 \times \mathbf{r}$  v koodrinatah glede na k.s. orodja kot

$$\begin{aligned} {}^{tool}\boldsymbol{\omega}_6 \times {}^{tool}\mathbf{r} &= \mathbf{R}^T \cdot {}^6\boldsymbol{\omega}_6 \times (\mathbf{R}^T \cdot \mathbf{d}) \\ &= -\mathbf{R}^T \cdot \mathbf{d} \times \mathbf{R}^T \cdot {}^6\boldsymbol{\omega}_6 \\ &= -\mathbf{S}(^T\mathbf{R} \cdot \mathbf{d})^T \cdot \mathbf{R} \cdot {}^6\boldsymbol{\omega}_6 \\ &= -\mathbf{R}^T \cdot \mathbf{S}(\mathbf{d}) \cdot \mathbf{R}^T \mathbf{T} \cdot \mathbf{R} \cdot {}^6\boldsymbol{\omega}_6 \\ &= -\mathbf{R}^T \cdot \mathbf{S}(\mathbf{d}) \cdot {}^6\boldsymbol{\omega}_6 \end{aligned} \quad (2.95)$$

$${}^{tool}\mathbf{v}_6 = \mathbf{R}^T \cdot {}^6\mathbf{v}_6. \quad (2.96)$$

S kombinacijo enačb (2.94), (2.95) in (2.96) dobimo translacijsko hitrost za k.s. orodja, zatem pa še enačbo (2.93) za določitev kotne hitrosti za k.s. orodja

$${}^{tool}\boldsymbol{v}_{tool} = \boldsymbol{R}^T \cdot {}^6\boldsymbol{v}_6 - \boldsymbol{R}^T \cdot \boldsymbol{S}(\boldsymbol{d}) \cdot {}^6\boldsymbol{\omega}_6 \quad (2.97)$$

$${}^{tool}\boldsymbol{w}_{tool} = \boldsymbol{R}^T \cdot {}^6\boldsymbol{\omega}_6. \quad (2.98)$$

Kar je mogoče zapisati v matrični obliki kot

$${}^{tool}\boldsymbol{\xi}_{tool} = \begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T \cdot \boldsymbol{S}(\boldsymbol{d}) \\ 0_{3 \times 3} & \boldsymbol{R}^T \end{bmatrix} {}^6\boldsymbol{\xi}_6. \quad (2.99)$$

V mnogo primerih je koristno razrešiti inverzni problem, torej izračunati želeno hitrost vrha, da bi izračunali želeno hitrost orodja. Ker je

$$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{S}(\boldsymbol{d}) \cdot \boldsymbol{R} \\ 0_{3 \times 3} & \boldsymbol{R} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T \cdot \boldsymbol{S}(\boldsymbol{d}) \\ 0_{3 \times 3} & \boldsymbol{R}^T \end{bmatrix}^{-1} \quad (2.100)$$

lahko razrešimo enačbo (2.99), računamo  ${}^6\boldsymbol{\xi}_6$ ,

$${}^6\boldsymbol{\xi}_6 = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{S}(\boldsymbol{d}) \cdot \boldsymbol{R} \\ 0_{3 \times 3} & \boldsymbol{R} \end{bmatrix} {}^{tool}\boldsymbol{\xi}_{tool}. \quad (2.101)$$

Izraz predstavlja splošen zapis za transformacijo hitrosti med dvema premikajočima se, vendar medsebojno fiksno pritrjenima k.s.

$${}^A\boldsymbol{\xi}_A = \begin{bmatrix} {}^A\boldsymbol{R}_B & \boldsymbol{S}({}^A\boldsymbol{d}_B) \cdot {}^A\boldsymbol{R}_B \\ 0_{3 \times 3} & {}^A\boldsymbol{R}_B \end{bmatrix} {}^B\boldsymbol{\xi}_B. \quad (2.102)$$

## 2.5 ANALITIČNA JAKOBIJEVA MATRIKA

Zgoraj izračunana Jakobijeva matrika je pogosto poimenovana Geometrijska Jakobijeva matrika, za razliko od Analitične Jakobijeve matrike, označene z  $\boldsymbol{J}_a(\boldsymbol{q})$ , ki pa je podana glede na k.s. vrha robota. Naj

$$\begin{bmatrix} \boldsymbol{d}(\boldsymbol{q}) \\ \boldsymbol{\alpha}(\boldsymbol{q}) \end{bmatrix} \dots \text{v } \boldsymbol{T}_6 \text{ k.s.} \quad (2.103)$$

predstavlja lego vrha, kjer je  $\boldsymbol{d}(\boldsymbol{q})$  kot običajno vektor iz izhodišča globalnega k.s. do k.s. vrha.  $\boldsymbol{\alpha}$  predstavlja orientacijo k.s. vrha relativno glede na

globalni k.s.. Npr.  $\boldsymbol{\alpha} = [\phi, \theta, \psi]^T$  je vektor podan z Eulerjevi koti. Iščemo torej zapis v obliki

$$\begin{bmatrix} \dot{\mathbf{d}} \\ \dot{\boldsymbol{\alpha}} \end{bmatrix} = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}} \quad (2.104)$$

za definiranje Analitičnega Jakobija.

Za zapis z Eulerjevimi koti  $\mathbf{R} = \mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta}\mathbf{R}_{z,\phi}$  je mogoče pokazati

$$\dot{\mathbf{R}} = \mathbf{S}(\boldsymbol{\omega}) \cdot \mathbf{R} \quad (2.105)$$

kjer  $\boldsymbol{\omega}$  definira kotno hitrost kot

$$\boldsymbol{\omega} = \begin{bmatrix} C_\psi S_\theta \dot{\phi} - S_\psi \dot{\theta} \\ S_\psi S_\theta \dot{\phi} + C_\psi \dot{\theta} \\ \dot{\psi} + C_\theta \dot{\phi} \end{bmatrix} = \begin{bmatrix} C_\psi S_\theta & -S_\psi & 0 \\ S_\psi S_\theta & C_\psi & 0 \\ C_\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{B}(\boldsymbol{\alpha})\dot{\boldsymbol{\alpha}} \quad (2.106)$$

Komponente v  $\boldsymbol{\omega}$  poimenujemo kot zvitost, vrtenje in zavrtitev. S kombinacijo te relacije s prejšnjo definicijo Jakobija

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{d}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (2.107)$$

dobimo

$$\mathbf{J}_G(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{d}} \\ \mathbf{B}(\boldsymbol{\alpha})\dot{\boldsymbol{\alpha}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{B}(\boldsymbol{\alpha}) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{d}} \\ \dot{\boldsymbol{\alpha}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{B}(\boldsymbol{\alpha}) \end{bmatrix} \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}. \quad (2.108)$$

Torej lahko izračunamo Analitično jakobijevu matriko  $\mathbf{J}_a(\mathbf{q})$  iz Geometrijske Jakobijeve matrike  $\mathbf{J}(\mathbf{q})$  preko

$$\mathbf{J}_a(\mathbf{q}) = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{B}^{-1}(\boldsymbol{\alpha}) \end{bmatrix} \mathbf{J}_G(\mathbf{q}) \text{ pri čemer } \mathbf{B}(\boldsymbol{\alpha}) \neq 0. \quad (2.109)$$

V naslednjem podpoglavlju so obravnave singularnosti jakobijeve matrike, obravnavane so torej konfiguracije, lege, v katerih Jakobijeva matrika izgubi rang. Singularnosti matrike  $\mathbf{B}(\boldsymbol{\alpha})$  so poimenovane tudi reprezentacijske singularnosti. Za  $s_\theta \neq 0$  e mogoče pokazati obstoj inverzne matrike  $\mathbf{B}(\boldsymbol{\alpha})$ . To pomeni, da singularnosti Analitičnega Jakobija  $\mathbf{J}_a(\mathbf{q})$  vsebujejo tako singularnosti Geometrijskega Jakobija kot tudi reprezentacijske singularnosti.

## 2.6 SINGULARNOSTI

Jakobijeva matrika  $\mathbf{J}(\mathbf{q})$  dimenzij  $6 \times n$  definira preslikavo

$$\boldsymbol{\xi} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (2.110)$$

Med vektorjem sklepnih hitrosti  $\dot{\boldsymbol{q}}$  in vektorjem hitrosti vrha  $\boldsymbol{\xi} = [\boldsymbol{v}^T, \boldsymbol{\omega}^T]^T$ . Zapis nakazuje, da so vse možne hitrosti vrha pravzaprav linearne kombinacije stolpcev iz Jakobijeve matrike.

$$\boldsymbol{\xi} = \mathbf{J}_1 \dot{\boldsymbol{q}}_1 + \mathbf{J}_2 \dot{\boldsymbol{q}}_2 + \dots + \mathbf{J}_n \dot{\boldsymbol{q}}_n. \quad (2.111)$$

Na primer, pri 2DOF ravninskem manipulatorju ima Jakobijeva matrika dva stolpca. Edina logična posledica je tudi translacijska hitrost samo v xy ravni, saj v nobenem od dveh stolpcev ni tretjega člena, oz. je ta enak nič. Ker je  $\boldsymbol{\xi} \in \mathbb{R}^6$  mora imeti matrika  $\mathbf{J}$  šest linearno neodvisnih stolpcev, da bi se vrh lahko gibal s šestimi neodvisnimi hitrostmi

Rang matrike je število linearно neodvisnih stolpcev (ali vrstic) v matriki. Torej, pri  $\mathbf{J} = 6$  se lahko vrh giblje s poljubno hitrostjo. Za matriko  $\mathbf{J} \in \mathbb{R}^{6 \times n}$  vedno velja  $\mathbf{J} \leq \min(6, n)$ . Pri 2DOF ravninskem manipulatorju imamo vedno rang  $\mathbf{J} \leq 2$ , pri antropomorfni roki s sferičnim zapestjem pa je rang vedno  $\mathbf{J} \leq 6$ .

Rang matrike ni nujno konstanten. Pravzaprav je rang Jakobijeve matrike odvisen od konfiguracije manipulatorja  $\boldsymbol{q}$ . Lege, pri katerih je je rang matrike  $\mathbf{J}(\boldsymbol{q})$  manjši od maksimalne vrednosti so poimenovane **singularnosti** ali **singularne konfiguracije**.

Poznavanje manipulatorjevih singularnosti je pomembno iz več razlogov.

- Singularnosti predstavljajo konfiguracije, iz katerih nekatere smeri gibanja niso možne.
- V singularnostih lahko končne hitrosti vrha nenadno postanejo neomejene hitrosti v sklepih.
- V singularnostih lahko končne vrednosti navorov v sklepih postanejo neomejene sile in navori na vrhu.
- Singularnosti pogosto nastopijo v točkah blizu roba manipulatorjevega delovnega prostora, torej v točkah maksimalnega iztega manipulatorja.
- Singularnosti ustrezajo tistim točkam v delovnem prostoru manipulatorja, ki so nedosegljive kljub majhnim spremembam pri parametrih segmentov, npr. dolžina, odmik in podobno.

Obstoja več metod s katerimi lahko določimo singularnosti jakobijeve matrike. V nadaljevanju bomo izrabili dejstvo, da je kvadratna matrika singularna pri determinanti enaki nič. V splošnem je težko rešiti nelinearno enačbo  $\det \mathbf{J}(\boldsymbol{q}) = 0$ . Zato sledi predstavitev metode za ločitev singularnosti, ki je uporabna pri npr. manipulatorjih s sferičnim zapestjem.

### 2.6.1 Ločitev singularnosti

Komplet enačb za direktno kinematiko je mogoče določiti za vsak manipulator, tako da najprej po želji, vendar togo prilepimo koordinatne sisteme na vsak segment, na osnovi tega pa izrazimo komplet homogenih enačb za koordinatne sisteme. Njihov produkt predstavlja celotno transformacijo. DH dogovor je samo sistematičen način za pravil in izvedbe. Čeprav so na ta način pridobljene enačbe odvisne od uporabljenih koordinatnih sistemov, so konfiguracije manipulatorja geometrijske veličine, neodvisne od uporabljenih k.s. s katerimi jih opišemo. Ob tem spoznanju je mogoče ločiti singularne konfiguracije pri tistih manipulatorjih s sferičnim zapestjem v dva bolj preprosta problema. Prvi je določitev tako imenovanih **singularnosti roke**, torej singularnosti kot posledica gibanja roke, ki je sestavljena iz treh ali več segmentov. Drugi predstavlja določitev **singularnosti zapestja** kot posledica gibanja sferičnega zapestja.

Upoštevajmo primer z  $n = 6$ , imamo torej manipulator sestavljen iz 3DOF roke z 3DOF zapestjem. V tem primeru je Jakobijeva matrika  $6 \times 6$  matrika in konfiguracija  $\mathbf{q}$  je singularna, če in samo če velja

$$\det \mathbf{J} = 0 \quad (2.112)$$

Jakobijevu matriko  $\mathbf{J}$  lahko razdelimo v podmatrike reda  $3 \times 3$

$$\mathbf{J} = [\mathbf{J}_P | \mathbf{J}_O] = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix}. \quad (2.113)$$

Zadnji trije sklepi so vedno rotacijski, zato velja

$$\mathbf{J}_O = \begin{bmatrix} \mathbf{z}_3 \times (\mathbf{o}_6 - \mathbf{o}_3) & \mathbf{z}_4 \times (\mathbf{o}_6 - \mathbf{o}_4) & \mathbf{z}_5 \times (\mathbf{o}_6 - \mathbf{o}_5) \\ \mathbf{z}_3 & \mathbf{z}_4 & \mathbf{z}_5 \end{bmatrix}. \quad (2.114)$$

Če se osi v zapestju sekajo v skupni točki  $\mathbf{o}$  kar je v primeru izbora za k.s.  $\mathbf{o}_3 = \mathbf{o}_4 = \mathbf{o}_5 = \mathbf{o}$  potem  $\mathbf{J}_O$  postane

$$\mathbf{J}_O = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{z}_3 & \mathbf{z}_4 & \mathbf{z}_5 \end{bmatrix}. \quad (2.115)$$

Jakobijeva matrika ima torej trikotno obliko

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & 0 \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix}. \quad (2.116)$$

z determinanto

$$\det \mathbf{J} = \det \mathbf{J}_{11} \cdot \det \mathbf{J}_{22},$$

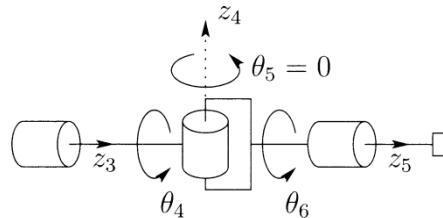
kjer sta  $\mathbf{J}_{11}$  in  $\mathbf{J}_{22}$  obe  $3 \times 3$  matriki.  $\mathbf{J}_{11}$  ima  $i$ -ti stolpec enak  $\mathbf{z}_{i-1} \times [\mathbf{o} - \mathbf{o}_{i-1}]$ , je sklep rotacijski in  $\mathbf{z}_{i-1}$ , če je sklep translacijski, medtem ko je

$$\mathbf{J}_{22} = [\mathbf{z}_3 \ \mathbf{z}_4 \ \mathbf{z}_5]. \quad (2.117)$$

Torej je skupno število singularnosti za cel manipulator sestavljeni iz tistih singularnosti pripadajočih roki pri  $\det \mathbf{J}_{11} = 0$  in tistih pripadajočih zapestju pri  $\det \mathbf{J}_{22} = 0$ . Ni nujno, da takšna oblika Jakobija podaja ustrezno relacijo med hitrostmi vrha in hitrostmi v sklepih. Predstavitev in ta način podajanja je namenjena zgolj poenostavitvi pri določanju singularnosti.

### 2.6.2 Singularnosti zapestja

Iz enačbe (2.117) je mogoče prepoznati singularnost sferičnega zapestja vedno pri medsebojno odvisnih vektorjih  $\mathbf{z}_3$ ,  $\mathbf{z}_4$  in  $\mathbf{z}_5$ . Skladno s sliko 2.10 se to zgodi pri sovpadanju osi  $\mathbf{z}_3$  in  $\mathbf{z}_5$ , torej pri  $\theta_5 = 0$  ali  $\pi$ . To sta edini singularnosti sferičnega zapestja in se jim ni mogoče izogniti brez uporabe mehanskih omejitev pri načrtovanju zapestja, izognemo se jim torej z omejitvijo gibanja tako, da se obe osi  $\mathbf{z}_3$  in  $\mathbf{z}_5$  nikoli ne poravnata. Pravzaprav nastopa singularnost vedno pri poravnavi dveh rotacijskih sklepov, saj enako in nasprotno predznačeno gibanje obeh sklepov ne vodi do gibanja vrha, torej je gibanje obeh osi medsebojno odvisno.



Slika 2.10: Singularnost sferičnega zapestja.

### 2.6.3 Singularnosti roke

Za določitev singularnosti roke je potrebno izračunati le  $\det \mathbf{J}_{11}$  kar je mogoče s pomočjo enačbe 2.70, vendar s centrom zapestja  $\mathbf{o}$  na mestu  $\mathbf{o}_n$ . V nadaljevanju so določene singularnosti za tri pogosto uporabljene robotske roke, za manipulator v obliki komolca (3R), za sferični manipulator in za SCARA manipulator.

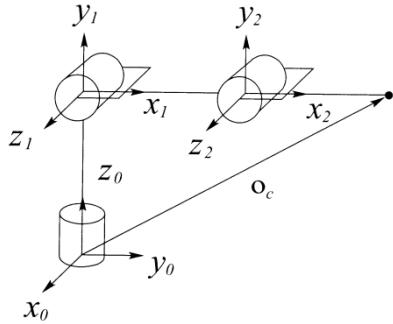
### 2.6.4 Singularnosti za manipulator v obliki komolca

Uporabimo trisegmentni manipulator s koordinatnimi sistemi nameščenimi skladno s sliko 2.11. Za vajo bi lahko izračunali tukaj že podano Jakobijevu matriko

$$\mathbf{J}_{11} = \begin{bmatrix} -a_2S_1C_2 - a_3S_1C_{23} & -a_2S_2C_1 - a_3S_{23}C_1 & -a_3C_1S_{23} \\ a_2C_1C_2 + a_3C_1C_{23} & -a_2S_1S_2 - a_3S_1S_{23} & -a_3S_1S_{23} \\ 0 & a_2C_2 + a_3C_{23} & a_3C_{23} \end{bmatrix} \quad (2.118)$$

in determinanto za  $\mathbf{J}_{11}$

$$\det \mathbf{J}_{11} = a_2a_3S_3(a_2C_2 + a_3C_{23}). \quad (2.119)$$



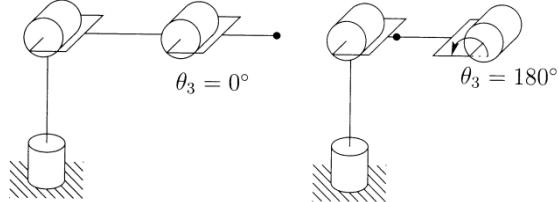
Slika 2.11: Manipulator v obliki komolca.

Iz zadnje enačbe prepoznamo singularnost manipulatorja pri

$$S_3 = 0, \text{ pri } \theta_3 = 0 \text{ ali } \pi$$

in vedno pri

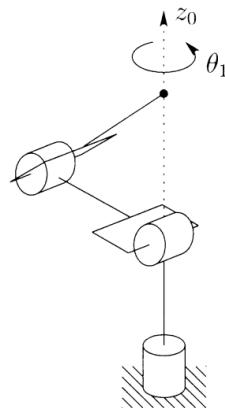
$$a_2C_2 + a_3C_{23} = 0.$$



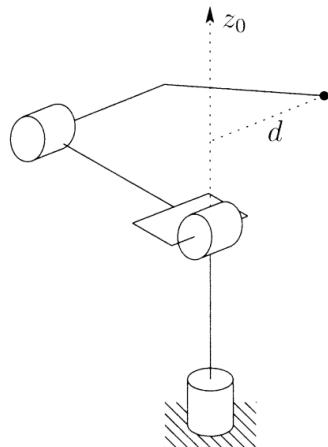
Slika 2.12: Singularnosti za manipulator v obliki komolca.

Lega za singularnost za prve enačbe (prvega pogoja) je pri polno iztegnjenem ali pa polno pokrčenem komolcu skladno s sliko 2.12.

Situacija za pogoj v drugi enačbi je pokazana na sliki 2.13. Ta primer nastopi pri sekjanju točke centra zapestja z osjo osnovne rotacije  $z_0$ . V tem primeru je mogoče prepoznati neskončno mnogo veliko singularnih leg in neskončno veliko rešitev za pozicijo pri inverzni kinematiki. To je kadar center zapestja leži vzdolž te osi.



Slika 2.13: Singularnost za manipulator v obliki komolca brez odmikov.

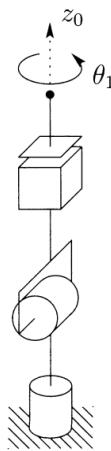


Slika 2.14: Singularnost za manipulator v obliki komolca z odmiki.

Manipulator v obliki komolca z odmikom je pokazan na Sliki 2.14 v singularni legi. Podobno kot prej tudi tukaj center zapestja ne sme sekati osi  $z_0$ . To opažanje dodatno utemeljuje trditev, da singularnih ročk ni mogoče doseči kljub majhnim spremembam manipulatorjevih parametrov, v tem primeru odmika v komolcu ali odmika v rami.

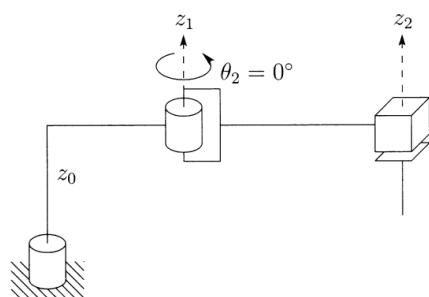
### 2.6.5 Sferični manipulator

Uporabimo sferično roko podano na sliki 2.15. Ta manipulator je v singularni legi kadar središče zapestja seka os  $z_0$ , kot je pokazano, saj podobno kot prej, kljub rotaciji ostaja ta točka vedno ista.



Slika 2.15: Singularnost sferičnega manipulatorja brez odmikov.

### 2.6.6 SCARA manipulator



Slika 2.16: Singularnost SCARA manipulatorja.

Jakobjevo matriko za SCARA manipulator smo že izračunali. Ta matrika je dovolj preprosta, da jo lahko uporabimo direktno, namesto prilagojene oblike Jakobia, kot je napravljeno zgoraj. Glede na sliko 2.16 je iz geometrije mogoče prepoznati edino singularno SCARA roke pri polno iztegnjenih ali

polno skrčenih segmentih. Ker je del SCARA Jakobija podan kot

$$\mathbf{J}_{11} = \begin{bmatrix} \alpha_1 & \alpha_3 & 0 \\ \alpha_2 & \alpha_4 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.120)$$

kjer je

$$\begin{aligned} \alpha_1 &= -a_1 S_1 - a_2 S_{12} \\ \alpha_2 &= a_1 C_1 + a_2 C_{12} \\ \alpha_3 &= -a_2 S_{12} \\ \alpha_4 &= a_2 C_{12} \end{aligned} \quad (2.121)$$

prepoznamo, da je rang  $\mathbf{J}_{11}$  manj kot tri točno pri  $\alpha_1\alpha_4 - \alpha_2\alpha_3 = 0$ . Preprosto je izračunati pogoj za takšno enakost kot

$$S_2 = 0, \text{ kar je izpolnjeno pri } \theta_2 = 0 \text{ ali } \pi.$$

Omeniti velja podobnost med tem primerom in singularnostjo za manipulator v obliki komolca na sliki 2.12. Pravzaprav je del robotske roke samo dvosegmentni ravninski manipulator. Skladno z enačbo (2.90) izgubi Jakobijeva matrika za dvosegmentni manipulator rang pri  $\theta_2 = 0$  ali  $\pi$ .

## 2.7 INVERZNA JACOBIJEVA MATRIKA

Včasih nas zanima obrnjena rešitev

$$\begin{bmatrix} {}^{T_6}\mathbf{d} \\ {}^{T_6}\boldsymbol{\delta} \end{bmatrix} = \mathbf{J} \cdot d\mathbf{q}_i. \quad (2.122)$$

Rešitev iščemo v obliki, kjer sta dana vektorja diferencialnega premika  ${}^{T_6}\mathbf{d}$  in orientacije  ${}^{T_6}\boldsymbol{\delta}$  ter hočemo izračunati diferencialne spremembe  $d\mathbf{q}_i$ . Moremo jo napisati v obliki

$$\mathbf{J}^{-1} \cdot \mathbf{J} \cdot d\mathbf{q}_i = \mathbf{J}^{-1} \cdot \begin{bmatrix} {}^{T_6}\mathbf{d} \\ {}^{T_6}\boldsymbol{\delta} \end{bmatrix} \quad (2.123)$$

$$d\mathbf{q}_i = \mathbf{J}^{-1} \cdot \begin{bmatrix} {}^{T_6}\mathbf{d} \\ {}^{T_6}\boldsymbol{\delta} \end{bmatrix} \quad (2.124)$$

ali tudi v daljši obliki in po analogiji glede na enačbo (2.12) napišemo

$$\begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix} = \begin{bmatrix} T_6 d_{1x} & T_6 d_{2x} & T_6 d_{3x} & T_6 d_{4x} & T_6 d_{5x} & T_6 d_{6x} \\ T_6 d_{1y} & T_6 d_{2y} & T_6 d_{3y} & T_6 d_{4y} & T_6 d_{5y} & T_6 d_{6y} \\ T_6 d_{1z} & T_6 d_{2z} & T_6 d_{3z} & T_6 d_{4z} & T_6 d_{5z} & T_6 d_{6z} \\ T_6 \delta_{1x} & T_6 \delta_{2x} & T_6 \delta_{3x} & T_6 \delta_{4x} & T_6 \delta_{5x} & T_6 \delta_{6x} \\ T_6 \delta_{1y} & T_6 \delta_{2y} & T_6 \delta_{3y} & T_6 \delta_{4y} & T_6 \delta_{5y} & T_6 \delta_{6y} \\ T_6 \delta_{1z} & T_6 \delta_{2z} & T_6 \delta_{3z} & T_6 \delta_{4z} & T_6 \delta_{5z} & T_6 \delta_{6z} \end{bmatrix}^{-1} \begin{bmatrix} T_6 d_x \\ T_6 d_y \\ T_6 d_z \\ T_6 \delta_x \\ T_6 \delta_y \\ T_6 \delta_z \end{bmatrix} \quad (2.125)$$

Torej potrebujemo inverzno matriko k Jacobijevi. To invertiranje je včasih možno celo v algebrajski obliki, najpogosteje pa to ni možno. Možna pot je tudi, da rešujemo nalogu invertiranja numerično, toda postopek je vezan na zamudno računanje. Se večjo nevšečnost predstavljajo singularnosti, ki vedno nastopijo, ko postane oblika manipulatorja (v neki legi) degenerirana. Seveda je numerična inverzija zahtevna, ker rabimo šest rešitev enačb. Zaradi tega je način, ko smo za znano  $\mathbf{T}_6$  in robot izračunali ustrezne notranje spremenljivke lažji, kajti za že dobljene izraze bomo računali diferencialne odvisnosti.

Izhajamo iz enačbe

$$\mathbf{T}_6 = \mathbf{A}_1 \cdot \dots \cdot \mathbf{A}_6, \quad \mathbf{A}_1^{-1} \cdot \mathbf{T}_6 = {}^1\mathbf{T}_6 \quad (2.126)$$

Izenačimo obe strani in izračunamo  $\theta_1$  ter nato  $d\theta_1$ .

Nato iz enačbe  $\mathbf{A}_2^{-1} \cdot \mathbf{A}_1^{-1} \cdot \mathbf{T}_6 = {}^2\mathbf{T}_6$  izračunamo  $\theta_2$  in druge notranje spremenljivke. Tako dobljene enačbe bomo odvajali in tako izračunali diferencialne odvisnosti med npr.  $d\theta_i$  in parametri  $\mathbf{T}_6$  matrike, kar vse sestavlja elemente  $d\mathbf{T}_6$  matrike. Tak pristop pogosto vodi hitreje do rešitev in z manj truda. V nadaljevanju bomo predpostavili, da obstaja algebrajska rešitev za manipulator in da ta rešitev podaja sinuse in cosinuse koordinat vsakega sklepa.

Z namenom, da izračunamo  $d\mathbf{T}_6$ , so podani vektorji diferencialne translacije  $\mathbf{D}$  ( $\mathbf{d}$  in rotacije  $\boldsymbol{\delta}$ ), ki jih najprej prevedemo v diferencialne translacije in rotacije glede na  $\mathbf{T}_6$ , to je izračunamo  ${}^{T_6}\mathbf{d}$  in  ${}^{T_6}\boldsymbol{\delta}$  ob uporabi en. (1.42) in (1.43). Nato izračunamo in sestavimo  ${}^{T_6}\Delta$ , nakar sledi izračun  $d\mathbf{T}_6$ . Celoten postopek bom prikazali za Stanfordski manipulator in za že izpeljane rešitve [Bajd, 2002].

Začnemo z  $d\theta_1$ . Izhajamo iz enačb

$$-S_1 p_x + C_1 p_y = d_2 \quad (2.127)$$

enačbo odvajamo in dobimo povezavo  $d\theta_1(dp_x, dp_y)$

$$\begin{aligned} -C_1 p_x d\theta_1 - S_1 ddp_x - S_1 p_y d\theta_1 + C_1 ddp_y &= 0 \\ -d\theta_1(C_1 p_x + S_1 p_y) &= -C_1 ddp_y + S_1 ddp_x \\ d\theta_1 &= \frac{C_1 ddp_y - S_1 ddp_x}{C_1 p_x + S_1 p_y} \end{aligned} \quad (2.128)$$

Ko smo tako dobili povezavo  $d\theta_1(dp_x, dp_y)$ , najprej preverimo, če  $d\theta_1$  ni premik v področje izven obsega področja  $\theta_1$ . Splošno to pomeni, da sprememba  $q_i + dq_i$  ni večja od  $q_{i_{max}}$ . Zaradi tega preverimo  $dq_i$  glede na  $q_i + dq_i \leq q_{i_{max}}$ .

$$dq_i \leq q_{i_{max}} - q_i \quad (2.129)$$

Ko izračunamo posamezne  $d\theta_i$ , tudi preverimo  $d(\sin \theta_i)$  in  $d(\cos \theta_i)$ .

$$d(\sin \theta_i) = \cos \theta_i d\theta_i \quad (2.130)$$

$$d(\cos \theta_i) = -\sin \theta_i d\theta_i \quad (2.131)$$

Izračun za  $d\theta_2, dd_3, d\theta_4, d\theta_5$  ter  $d\theta_6$  bomo zaradi dolžine izpustili.

Tako smo dobili numerične vrednosti  $d\mathbf{q}_i$  ne da bi pravzarav izračunali  $\mathbf{J}^{-1}$ , saj smo  $d\mathbf{q}_i$  računali z odvajanjem spremenljivk  $\theta_i$  in  $d_i$ . Izračun preostalih kotov ne bomo prikazali, poteka pa na način podoben za kot v prvem sklepu.

### 2.7.1 Ponovitev

Spoznali smo, da  $d\mathbf{T}$  lahko izračunamo ob uporabi diferencialne transformacije  $\Delta$  tako da izračunamo  $d\mathbf{T} = \Delta \cdot \mathbf{T}$ . Matrika  $\Delta$  je podana glede na osnovni koordinatni sistem, zaradi tega smo  $\mathbf{T}$  premultiplicirali s transformacijo  $\Delta$ .

Če želimo izraziti diferencialne spremembe v poziciji in orientaciji glede na relativni koordinatni sistem, ima  $\mathbf{T}$  diferencialna transformacija druge vrednosti in dobimo matriko  ${}^T\Delta$ . S to transformacijo izračunamo enak premik  $d\mathbf{T}$ , tako da  $\mathbf{T}$  postmultipliciramo (relativni koordinatni sistem).

$$dT = T \cdot {}^T\Delta.$$

Transformacija za diferencialne spremembe je definirana z dvema vektorjema:  $\mathbf{d}$  - vektor premika, translacije in  $\boldsymbol{\delta}$  - vektor rotacije, sprememba orientacije. Oba vektorja v celoti definirata  $\Delta$  kot

$$\Delta = \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ podana glede na osnovni koordinatni sistem.}$$

Elementi v  $\boldsymbol{\delta}$  predstavljajo diferencialne rotacije okoli  $x$ ,  $y$  in  $z$  osi referenčnega koordinatnega sistema. Izkaže se, da zaporedje operacij diferencialnih rotacij ne vpliva na rezultat. Te diferencialne rotacije so tudi ekvivalentne diferencialni rotaciji  $d\theta$  okolinekega enotskega vektorja  $\mathbf{k}$ , če ustreza

$$\begin{aligned} \delta_x &= k_x d\theta \\ \delta_y &= k_y d\theta \\ \delta_z &= k_z d\theta. \end{aligned}$$

Nadalje smo v tem poglavju pokazali povezavo med  ${}^T\Delta$  in  $\Delta$

$${}^T\Delta = \mathbf{T}^{-1} \cdot \Delta \cdot \mathbf{T}.$$

Tu je transformacija  ${}^T\Delta$  sestavljena iz elementov vektorjev

$$\begin{aligned} {}^T\mathbf{d} &= {}^T d_x \mathbf{i} + {}^T d_y \mathbf{j} + {}^T d_z \mathbf{k} \\ {}^T\boldsymbol{\delta} &= {}^T \delta_x \mathbf{i} + {}^T \delta_y \mathbf{j} + {}^T \delta_z \mathbf{k} \end{aligned}$$

poznana kot diferencialna koordinatna transformacija  ${}^T\Delta$ .

Elementi v  ${}^T\Delta$  so glede na elemente  $\Delta$  takole definirani

$$\begin{aligned} {}^T d_x &= \mathbf{n}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ {}^T d_y &= \mathbf{o}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ {}^T d_z &= \mathbf{a}((\boldsymbol{\delta} \times \mathbf{p}) + \mathbf{d}) \\ {}^T \delta_x &= \mathbf{n} \cdot \boldsymbol{\delta} \\ {}^T \delta_y &= \mathbf{o} \cdot \boldsymbol{\delta} \\ {}^T \delta_z &= \mathbf{a} \cdot \boldsymbol{\delta}. \end{aligned}$$

Nato smo ugotovili, za nek transformacijski zapis  $\mathbf{A} \cdot \mathbf{B}$ , ki ga ponazorimo z grafom

- vzroke in
- posledice.

$$\begin{aligned} {}^T\Delta &= \mathbf{T}^{-1} \cdot \Delta \cdot \mathbf{T} \\ \Delta &= (\mathbf{T}')^{-1} \cdot {}^T\Delta \cdot (\mathbf{T}'). \end{aligned}$$

Ustrezno diferencialno koordinatno transformacijo, ki povezuje diferencialne transformacije med fiksnim in gibljivim koordinatnim sistemom zapišemo tako, da bomo začeli na vrhu tiste transformacije, kjer je diferencialni pomik podan. Torej začnimo od vrha transformacije, kjer je ekvivalentni diferencialni pomik posledica oz. iskana transformacija  $\Delta$

$$\Delta = (\mathbf{B}^{-1} \cdot \mathbf{A}^{-1})^{-1} \cdot {}^{AB}\Delta \cdot (\mathbf{B}^{-1} \cdot \mathbf{A}^{-1}).$$

Za obratni primer  ${}^{AB}\Delta = (\mathbf{A} \cdot \mathbf{B})^{-1} \cdot \Delta \cdot (\mathbf{A} \cdot \mathbf{B})$  dobimo  ${}^{AB}\Delta$ .

Primer: Podana je  ${}^Y\Delta$  in hočemo določiti  ${}^X\Delta$

$${}^X\Delta = (\mathbf{Y}^{-1} \cdot \mathbf{C}^{-1})^{-1} \cdot {}^Y\Delta \cdot (\mathbf{Y}^{-1} \cdot \mathbf{C}^{-1}).$$

S pomočjo tako izpeljanih diferencialnih koordinatnih transformacij smo se lotili izračuna Jacobijeve matrike. V tem primeru je diferencialna sprememba (transformacija) v  $i$ -tem sklepu - koordinatnem sistemu  $i - 1$  - povzročila diferencialno koordinatno spremembo.

Diferencialni translacijski in rotacijski vektorji, ki predstavljajo stolpec Jacobijeve matrike za rotacijski sklep, so podani kot

$$\begin{aligned} {}^{T_6}\mathbf{d} &= (-n_x p_y + n_y p_x) \mathbf{i} + (o_x p_y + o_y p_x) \mathbf{j} + (-a_x p_y + a_y p_x) \mathbf{k} \\ {}^{T_6}\boldsymbol{\delta} &= n_z \mathbf{i} + o_z \mathbf{j} + a_z \mathbf{k}. \end{aligned}$$

Za translacijski sklep smo izpeljali vrednosti

$$\begin{aligned} {}^{T_6}\mathbf{d} &= n_z \mathbf{i} + o_z \mathbf{j} + a_z \mathbf{k} \\ {}^{T_6}\boldsymbol{\delta} &= 0 \mathbf{i} + 0 \mathbf{j} + 0 \mathbf{k}. \end{aligned}$$

Tako smo uspeli izračunati Jacobijevo matriko za Stanfordski manipulator.

## 2.8. REDUNDANTNOST IN VODENJE PREKO IZRAČUNANE HITROSTI 69

tor, ki podaja premik  ${}^T_6 \mathbf{d}$  in  ${}^T_6 \boldsymbol{\delta}$ .

$$\begin{bmatrix} {}^T_6 d_x \\ {}^T_6 d_y \\ {}^T_6 d_z \\ {}^T_6 \delta_x \\ {}^T_6 \delta_y \\ {}^T_6 \delta_z \end{bmatrix} = \begin{bmatrix} {}^T_6 d_{1x} & {}^T_6 d_{2x} & {}^T_6 d_{3x} & {}^T_6 d_{4x} & {}^T_6 d_{5x} & {}^T_6 d_{6x} \\ {}^T_6 d_{1y} & {}^T_6 d_{2y} & {}^T_6 d_{3y} & {}^T_6 d_{4y} & {}^T_6 d_{5y} & {}^T_6 d_{6y} \\ {}^T_6 d_{1z} & {}^T_6 d_{2z} & {}^T_6 d_{3z} & {}^T_6 d_{4z} & {}^T_6 d_{5z} & {}^T_6 d_{6z} \\ {}^T_6 \delta_{1x} & {}^T_6 \delta_{2x} & {}^T_6 \delta_{3x} & {}^T_6 \delta_{4x} & {}^T_6 \delta_{5x} & {}^T_6 \delta_{6x} \\ {}^T_6 \delta_{1y} & {}^T_6 \delta_{2y} & {}^T_6 \delta_{3y} & {}^T_6 \delta_{4y} & {}^T_6 \delta_{5y} & {}^T_6 \delta_{6y} \\ {}^T_6 \delta_{1z} & {}^T_6 \delta_{2z} & {}^T_6 \delta_{3z} & {}^T_6 \delta_{4z} & {}^T_6 \delta_{5z} & {}^T_6 \delta_{6z} \end{bmatrix} \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix}$$

$$\begin{bmatrix} {}^T_6 \mathbf{d} \\ {}^T_6 \boldsymbol{\delta} \end{bmatrix} = \mathbf{J} d\mathbf{q}$$

Jacobijeva matrika je podana v zunanjih koordinatah (vrha) kot posledica diferencialnih premikov v sklepih. Bolj zanimiva je inverzna Jacobijeva matrika, ki omogoča izračun potrebnih (posledičnih) sprememb v spremenljivkah sklepov, ki ustrezajo podani diferencialni spremembi  ${}^T_6 \mathbf{d}$  in  ${}^T_6 \boldsymbol{\delta}$  matrike. Izračun  $\mathbf{J}^{-1}$  je težak in zahteven.

## 2.8 REDUNDANTNOST IN VODENJE PREKO IZRAČUNANE HITROSTI

Enačba

$$\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}} \quad (2.132)$$

- $\dot{\mathbf{p}}$  ... zunanje spremenljivke
- $\dot{\mathbf{q}}$  ... notranje spremenljivke
- $\mathbf{J}$  ... Jacobijeva matrika za  $n = 6$  je  $6 \times 6$  matrika
- $\dot{\mathbf{q}} = [\dot{q}_1 \dots \dot{q}_n]^T$  je  $1 \times n$  vektor hitrosti po sklepih (za prvi sklep  $\dot{q}_1$ )

- $\dot{\mathbf{p}} = [\dot{p}_x, \dot{p}_y, \dot{p}_z, \omega_x, \omega_y, \omega_z]^T$ ,  $\omega_x = \frac{\theta_x}{dt}$ ,  $\omega_y = \frac{\theta_y}{dt}$ ,  $\omega_z = \frac{\theta_z}{dt}$

podaja hitrost, linearno in kotno, torej hitrost vrha manipulatorja v odvisnosti od hitrosti sklepov robota. Ob takšni rešitvi nas zanima obratna pot. Na osnovi danih hitrosti linearnega pomika in rotacije vrha manipulatorja

iščemo ustrezne hitrosti sklepov, ki vodijo do podanih hitrosti vrha manipulatorja.

Vsaj 6 prostostnih stopenj  $n = 6$  moramo imeti, da lahko poljubno predpišemo linearne in kotne hitrosti vrha manipulatorja. Torej moramo obravnavati inverzni problem za manipulator z  $n = 6$  sklepi. Tako bo  $\mathbf{J}$  matrika reda  $6 \times 6$ . Če je  $\mathbf{J}$  matrika nesingularna za dano konfiguracijo, tedaj inverzna matrika  $\mathbf{J}^{-1}$  tudi obstaja in smemo napisati

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{p}}. \quad (2.133)$$

En. (2.133) določa na podlagi podanih linearnih in rotacijskih hitrosti vrha manipulatorja ustrezne potrebne hitrosti sklepov  $\dot{\mathbf{q}} = [q_1 \dots q_n]^T$ . Če tako določamo potrebne sklepne hitrosti, ki zagotovijo podane linearne in rotacijske hitrosti vrha manipulatorja, potem lahko izračunane  $\dot{q}_i$  uporabimo za vodenje - regulacijo manipulatorja. Takšno regulacijo imenujemo vodenje preko izračunane hitrosti (resolved motion rate control). Ta način krmiljenja je predlagal že Whitney 1969 leta.

Ker se  $\mathbf{J}$  spreminja z lego manipulatorja, lahko postane singularna za dane lege (konfiguracije) in v teh legah inverzna  $\mathbf{J}^{-1}$  ne obstaja, torej tudi rešitev po en. (2.133) ni možna. Ustrezno lego manipulatorja, kjer  $\mathbf{J}^{-1}$  ne obstaja zaradi singularnosti, bomo poimenovali singularna lega ali konfiguracija manipulatorja. V singularni legi matrika  $\mathbf{J}$  nima polnega ranga torej so stolpni vektorji med seboj odvisni in ne pokrivajo celotnega 6-dimenzionalnega vektorskoga prostora  $\dot{\mathbf{p}}$ . Torej obstaja vsaj ena taka smer, v kateri ni mogoče pomakniti vrha manipulatorja.

Za nesingularno lego manipulatorja obstaja pri 6 prostostnih stopnjah vedno enoznačna rešitev za  $\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{p}}$ .

Če imamo redundantni manipulator in je  $n > 6$ , moremo najti enačbi  $\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{p}}$  neskončno rešitev. Predstavljajmo si našo roko, ki ima 7 prostostnih stopenj, če ne računamo sklepov prstov (rama 3, zapestje 3, komolec). Če prsti počivajo na mizi, moremo ramo in komolec poljubno premikati, ne da bi premaknili prste. To pomeni, da obstaja neka domena hitrosti sklepov, ki vodi do takšne rešitve, da se vrh manipulatorja sploh ne premakne. Torej ta nabor sklepnih hitrosti nič ne prispeva k rezultirajoči hitrosti vrha manipulatorja. Tudi, če tako definirane hitrosti sklepov superponiramo na druge sklepne hitrosti, bo končna sklepna hitrost vrha manipulatorja nespremenjena. Kot posledica moremo najti resnične rešitve trenutne kinematične

enačbe  $\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{p}}$  za dano hitrost vrha manipulatorja. Tukaj želimo podrobneje pogledati reševanje trenutne kinematične rešitve za primer, ko je  $n > 6$ .

Zamislimo si manipulator, ki ima  $n$ -prostostnih stopenj. Število prostostnih stopenj n ni nujno v praksi vedno 6, npr. pri varjenju lahko elektrodo ali šobo okoli vzdolžne osi poljubno zavrtimo in to nič ne vpliva na varjenje, torej je za ta primer  $n = 5$  dovolj. V splošnem opišemo vrh manipulatorja z  $m$ -neodvisnimi spremenljivkami, ki so potrebne, da opišemo želeno nalogu (npr. varjenje). Torej bomo opisali vrh manipulatorja z  $m$ -neodvisnimi spremenljivkami. Naj bo

$$d\mathbf{p} = [dp_1 \dots dp_m]^T$$

nek  $m \times 1$  vektor, ki opisuje infinitezimalni pomik vrha manipulatorja. Tedaj je trenutna kinematična enačba podana za  $n$ -prostostni manipulator kot

$$\begin{aligned} d\mathbf{q} &= [dq_1 \dots dq_n]^T \\ d\mathbf{p} &= \mathbf{J} \cdot d\mathbf{q} \end{aligned} \tag{2.134}$$

V tem primeru je  $\mathbf{J}$  matrika dimenzije  $m \times n$  ( $m$  število vrstic,  $n$  število stolpcov). Če je  $n$  večji od  $m$  in je  $\mathbf{J}$  polnega ranga, tedaj je  $n - m$  spremenljivk, ki lahko zavzemajo poljubno vrednost v rešitvi enačbe. Tedaj pravimo, da ima manipulator  $n - m$  redundantnih prostostnih stopenj glede na dano nalogu.

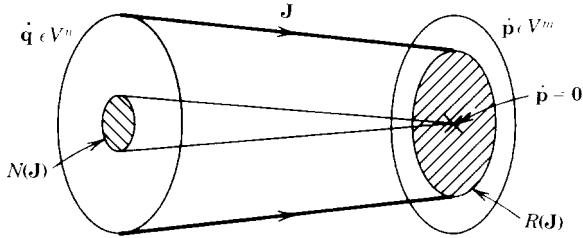
Vemo, da  $\mathbf{J}$  podaja preslikavo med hitrostjo vrha manipulatorja in hitrostmi v posameznih sklepih  $\dot{\mathbf{q}}$ .

$$\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}} \tag{2.135}$$

En. (2.135) smemo razložiti kot linearne preslikave iz  $n$ -dimenzionalnega vektorskoga prostora  $V^n$  v  $m$ -dimenzionalni vektorski prostor  $V^m$ . Da to prikažemo nazorno, uporabimo skico 2.17.

Podprostor  $R(\mathbf{J})$  je prostor rešitev, ki zajema vse tiste hitrosti vrha  $\dot{\mathbf{p}}$ , ki jih lahko generiramo z  $n$ -sklepi v dani izbrani konfiguraciji, legi manipulatorja.

Če je rang  $\mathbf{J}$  poln (vsi stolpci neodvisni), bo prostor obsega rešitev prekril celoten vektorski prostor  $V^m$ . V nasprotnem obstaja vsaj ena smer, v katero ne moremo premakniti vrha manipulatorja. Podprostor  $N(\mathbf{J})$  na sliki je



Slika 2.17: Slikovna ponazoritev linearne kinematične preslikave.

ničti prostor linearne preslikave. Poljubni element v tem ničtem prostoru  $N(\mathbf{J})$  se preslika v ničti vektor  $\dot{\mathbf{p}} = \mathbf{0}$  v prostoru  $V^m$ , ker je tedaj  $\mathbf{J} \cdot \dot{\mathbf{q}} = 0$ . Poljuben vektor hitrosti sklepov  $\dot{\mathbf{q}}$ , ki pripada ničtemu prostoru, ne bo proizvedel nobene hitrosti vrha manipulatorja (prsti na mizi - gibanje komolca in ramena). Torej, ko damo prste na mizo in premikamo ramo in komolec, bodo vsi vektorji  $\dot{\mathbf{q}}_i$  hitrosti sklepov iz ničtega prostora vodili do  $\dot{\mathbf{p}} = \mathbf{0}$  - hitrosti vrha manipulatorja. Če je  $\mathbf{J}$  polnega ranga, bo dimenzija ničtega prostora,  $N(\mathbf{J})$  enaka številu redundantnih prostostnih stopenj ( $n-m$ ). Če je  $N(\mathbf{J})$  degenerirana, to je nima polnega ranga, bo dimenzija prostora obsega rešitev dim  $R(\mathbf{J})$  zmanjšana in hkrati dimenzija ničtega prostora povečana za enak iznos. To se zgodi, ker je vsota obeh prostorov enaka  $n$

$$\dim R(\mathbf{J}) + \dim N(\mathbf{J}) = n. \quad (2.136)$$

To dokažemo tako, da predpostavimo, da je  $\dot{\mathbf{q}}^*$  rešitev enačbe  $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$  in  $\dot{\mathbf{q}}_0$  vektor iz ničtega prostora. Torej bo vektor z obliko

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}^* + k \cdot \dot{\mathbf{q}}_0$$

tudi rešitev en. (2.133).

Tu je  $k$  poljubni skalar. Velja namreč

$$\mathbf{J} \cdot \dot{\mathbf{q}} = \mathbf{J} \cdot \dot{\mathbf{q}}^* + k \cdot \mathbf{J} \cdot \dot{\mathbf{q}}_0 = \mathbf{J} \cdot \dot{\mathbf{q}}^* = \dot{\mathbf{p}}. \quad (2.137)$$

Ker člen  $k \cdot \mathbf{J} \cdot \dot{\mathbf{q}}_0$  lahko poljubno izbiramo v ničtem prostoru, imamo opravka z neskončno množico rešitev, kolikor dimenzija ničnega prostora ni 0. Torej ničti prostor vodi do poljubno mnogo rešitev. Glede na to je splošna rešitev linearne enačbe takšna, da vključuje toliko poljubnih parametrov, kot je dimenzija ničtega prostora.

## 2.9 OPTIMALNA REŠITEV, PSEVDOINVERZNA J MATRIKA

Kadar je manipulator redundanten ( $m < n$ ) ima  $\mathbf{J}$  matrika več stolpcev kot vrstic in sistem ima neskončno rešitev. Ena od možnosti je iskanje optimalne rešitve glede na izbrani kriterij preko linearne optimizacije. Linearne zato, ker je funkcija cene linearna (en.(2.140), v nadaljevanju).

Tu bomo razpravljali o optimalni rešitvi za enačbo, ki povezuje hitrosti

$$\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}} \quad (2.138)$$

Zamrznimo  $\mathbf{J}$  oziroma manipulator v neki inverzni konfiguraciji in poiščimo optimalno rešitev za linearno en. (2.138) ob predpostavki, da je  $\mathbf{J}$  matrika s polnim rangom (popolnoma neodvisni stolpci, torej poln rang, kadar ni v singularni legi). Preverjali bomo optimalnost rešitev ob uporabi kvadratične funkcije cene za vektor hitrosti sklepov po nastavku

$$G(\dot{\mathbf{q}}) = \dot{\mathbf{q}}^T \cdot \mathbf{W} \cdot \dot{\mathbf{q}} \quad (2.139)$$

Tu je  $\mathbf{W}$  neka  $n \times n$  simetrična, pozitivno definitna matrika uteži.

Problem je zastavljen na ta način, da najdemo  $\dot{\mathbf{q}}$  tako, da bo zadostil enačbo (2.138) za dani  $\dot{\mathbf{p}}$  in  $\mathbf{J}$  tako, da bo funkcija cene  $G(\dot{\mathbf{q}})$  minimizirana. To naložo bomo rešili ob uporabi Lagrangeovega multiplikatorja. Pogosto ta način poimenujemo tudi vezan ekstrem.

$$G(\dot{\mathbf{q}}, \lambda) = \dot{\mathbf{q}}^T \cdot \mathbf{W} \cdot \dot{\mathbf{q}} - \lambda^T (\mathbf{J} \cdot \dot{\mathbf{q}} - \dot{\mathbf{p}}) \quad (2.140)$$

Tu je  $\lambda$  oblike  $m \times 1$ , na začetku neznani vektor, ki ga imenujemo Lagrangeov multiplikator. Potrebna pogoja, ki jima mora zadostiti optimalna rešitev, sta

$$\begin{aligned} \left( \frac{\partial G}{\partial \dot{\mathbf{q}}} \right)^T &= 0 \rightarrow (2 \cdot \dot{\mathbf{q}}^T \cdot \mathbf{W} - \lambda^T \cdot \mathbf{J})^T = 0 \\ 2(\dot{\mathbf{q}}^T \cdot \mathbf{W})^T - (\lambda^T \cdot \mathbf{J})^T &= 0 \\ 2 \cdot \mathbf{W}^T \cdot \dot{\mathbf{q}} - \mathbf{J}^T \cdot \lambda &= 0 \\ \dot{\mathbf{q}} &= \frac{1}{2} \mathbf{W}^{-1} \cdot \mathbf{J}^T \cdot \lambda, \quad \mathbf{W} = \mathbf{W}^T \end{aligned} \quad (2.141)$$

in še

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{J} \cdot \dot{\mathbf{q}} \\ \left( \frac{\partial G}{\partial \lambda} \right)^T &= 0 \text{ ter } \dot{\mathbf{p}} = \mathbf{J} \cdot \frac{1}{2} \mathbf{W}^{-1} \cdot \mathbf{J}^T \cdot \lambda \text{ in} \\ 2 \cdot \dot{\mathbf{p}} &= (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T) \cdot \lambda. \end{aligned} \quad (2.142)$$

Ker vemo, da je matrika  $\mathbf{W}$  pozitivno definitna, jo je mogoče invertirati in glede na to lahko poiščemo rešitev v obliki iz(2.141)

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{W}^{-1} \cdot \mathbf{J}^T \cdot \lambda. \quad (2.143)$$

Če (2.143) vstavimo v (2.142) dobimo

$$2 \cdot \dot{\mathbf{p}} = (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T) \cdot \lambda. \quad (2.144)$$

Ker smo predpostavili, da ima  $\mathbf{J}$  polni rang (neodvisni stolpci), bo matični produkt  $\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T$  predstavljal kvadratno matriko s polnim rangom in jo je torej mogoče invertirati. Iz en. (2.144) izračunavamo Lagrangeov multiplikator (premultipliziramo z  $(\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1}$ ) in dobimo

$$\begin{aligned} (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1} \cdot (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T) \cdot \lambda &= 2 \cdot (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}} \\ \lambda &= 2 \cdot (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}}. \end{aligned}$$

Ta  $\lambda$  vstavimo v enačbo (2.143) ter dobimo optimalno rešitev

$$\begin{aligned} \dot{\mathbf{q}} &= \frac{1}{2} \mathbf{W}^{-1} \cdot \mathbf{J}^T \cdot 2 \cdot (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}} \\ \dot{\mathbf{q}} &= \mathbf{W}^{-1} \cdot \mathbf{J}^T (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}} \end{aligned} \quad (2.145)$$

Spoznamo, da en. (2.145) zadosti osnovni enačbi hitrosti (2.138), kar dokazemo tako, da (2.145) z  $\mathbf{J}$  premultipliziramo. Če je utežnostna matrika  $\mathbf{W}$  neka  $m \times m$  matrika identičnosti, se bo podana optimalna rešitev (2.145) zreducirala v

$$\begin{aligned} \mathbf{J} \cdot \dot{\mathbf{q}} &= \mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T \cdot (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}} \\ \mathbf{J} \cdot \dot{\mathbf{q}} &= \mathbf{J} \cdot \mathbf{J}^T \cdot (\mathbf{J} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}} \\ \mathbf{J} \cdot \dot{\mathbf{q}} &= \dot{\mathbf{p}}. \end{aligned}$$

Dobljeni  $\dot{\mathbf{q}}$  optimalno rešuje  $\mathbf{J} \cdot \dot{\mathbf{q}} = \dot{\mathbf{p}}$  glede na  $G(\dot{\mathbf{q}})$ . Torej imamo rešitev, ki nas privede nazaj na enačbo  $\mathbf{J} \cdot \dot{\mathbf{q}} = \dot{\mathbf{p}}$ . To pomeni, da je funkcija cene  $G(\dot{\mathbf{q}})$  in preko  $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$  izraz za  $\dot{\mathbf{q}} = \mathbf{W}^{-1} \cdot \mathbf{J}^T \cdot (\mathbf{J} \cdot \mathbf{W}^{-1} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}}$  ustrezan. Tak  $\dot{\mathbf{q}}$  ustreza  $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$ .

Glede na ta dokaz en. (2.145) ob upoštevanju, da je  $\mathbf{W}$  utežnostna matrika identičnosti, napišemo pri  $\mathbf{W} = \mathbf{I}$

$$\dot{\mathbf{q}} = \mathbf{J}^T \cdot (\mathbf{J} \cdot \mathbf{J}^T)^{-1} \cdot \dot{\mathbf{p}}$$

in primerjamo z

$$\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}, \text{ oz. } \dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{p}}, \text{ torej}$$

$$\mathbf{J}^\# = \mathbf{J}^T \cdot (\mathbf{J} \cdot \mathbf{J}^T)^{-1}.$$

$\mathbf{J}^\#$  označimo za psevdoinverzno matriko matrike  $\mathbf{J}$ , ker velja

$$\dot{\mathbf{q}} = \mathbf{J}^\# \cdot \dot{\mathbf{p}},$$

saj nas ta izraz spomni na

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{p}}.$$

Sklepamo torej, da je  $\mathbf{J}^\#$  na drug način podana  $\mathbf{J}^{-1}$ , torej psevdo inverzna.



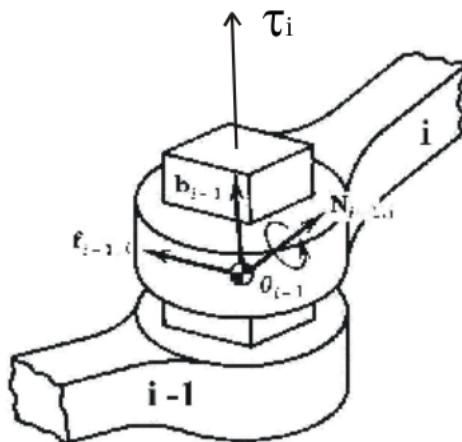
# Poglavlje 3

## STATIKA

### 3.1 EKVIVALENTNI MOMENTI SKLEPA

Tu bomo študirali funkcionalno odvisnost med vhodnimi momenti, ki jih izvajajo aktuatorji v sklepih, in silami in momenti vrha manipulatorja na okolico. Vsak sklep ima aktuator, ki izvaja pogonski moment ali silo med sosednjima segmentoma sklepa.  $\tau_i$  naj pomeni pogonski moment ali silo, ki jo izvaja  $i$ -ti sklepni aktuator.

Za translacijski sklep je pogonska sila  $\tau_i$ , ki je usmerjena vzdolž osi sklepa  $i - 1$  kot to kaže slika 3.1.



Slika 3.1: Translacijski sklep.

Predpostavimo, da v sklepu ne nastopa trenje. Tedaj lahko preprosto povežemo silo sklepa  $\tau_i$  s silo sklepa  $f_{i-1,i}$ , ki deluje med segmentoma  $i - 1$

in  $i$

$$\boldsymbol{\tau}_i = \mathbf{b}_{i-1}^T \cdot \mathbf{f}_{i-1,i} \quad (3.1)$$

$\mathbf{b}_{i-1}$  ... enotski vektor v smeri sklepa.

Poudarimo, da gre za skalarni produkt  $\mathbf{b}_{i-1}^T \cdot \mathbf{f}_{i-1,i}$ .

En. (3.1) v osnovi pravi, da bo samo tista komponenta vekorja  $\mathbf{f}_{i-1,i}$ , ki je v smeri osi sklepa, aktivno kompenzirana. Vse ostale komponente sile  $\mathbf{f}_{i-1,i}$  bodo uravnovežene s silami, ki delujejo kot reakcije v strukturi sklepa v obliki notranjih sil, ali nekje drugje v kakšnem drugem sklepu.

Te komponente sile  $\mathbf{f}_{i-1,i}$  so sile omejitev sklepa, ki se ne more gibati v teh smereh. Torej ne opravlja dela, temveč samo obremenjujejo konstrukcijo.

Za rotacijski sklep bo  $\boldsymbol{\tau}_i$  podajal moment pogonskega motorja v sklepu. Ta pogonski moment mora uravnovežiti moment sklepa  $\mathbf{N}_{i-1,i}$  in to samo tisto komponento, ki je vzporedna osi sklepa

$$\boldsymbol{\tau}_i = \mathbf{b}_{i-1}^T \cdot \mathbf{N}_{i-1,i} \quad (3.2)$$

Ostale komponente momenta  $\mathbf{N}_{i-1,i}$  prevzame konstrukcija sklepa in torej ne prispevajo k delu, temveč obremenjujejo samo konstrukcijo. Vse aktivne sile in momente  $\boldsymbol{\tau}_i$  vseh sklepov združimo v stolpni vektor  $\boldsymbol{\tau}_i$ , ki ima  $n$  vrstic in je podan z enačbo

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix}. \quad (3.3)$$

Vektor  $\boldsymbol{\tau}$  imenujemo vektor pogonskih sklepnih sil oziroma momentov. Ta vektor predstavlja pogonske sile in momente, ki jih izvajajo motorji na ustrezne segmente. Zveza med vektorjem  $\boldsymbol{\tau}$  pogonskih aktivnih sil in momentov ter vektorjem vrha manipulatorja  $\mathbf{F}$ , je podana s enačbo (3.4).

Za sklepe brez trenja bodo pogonske sile in momenti morali kompenzirati vektor sil in momentov vrha manipulatorja  $\mathbf{F}$  v obliki

$$\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F} \quad (3.4)$$

$\mathbf{J}^T$  ... transponirana Jacobijeva matrika dimenzij  $6 \times n$ .

Že od prej pa poznamo vpletenost Jacobijeve matrike v diferencialne pomike oziroma hitrosti.

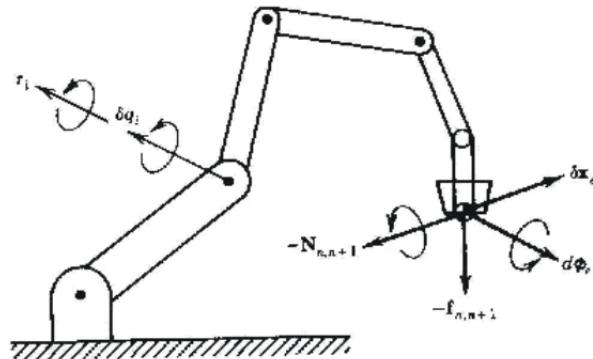
$$d\mathbf{p} = \mathbf{J} \cdot d\mathbf{q} \quad (3.5)$$

$d\mathbf{p}$  ... diferencialni vektor premikov vrha,

$d\mathbf{q}$  ... diferencialni vektor premikov sklepov.

Sprevideti moramo, da v en. (3.4) niso zajeti momenti in sile, ki jih povzroča težnost ali katerekoli druge sile in momenti, npr. zaradi trenja, vztrajnosti, itd. Torej bo  $\boldsymbol{\tau}$  podajal ekvivalentne pogonske sile in momente za silo in moment vrha manipulatorja.

Zvezo (3.4) lahko dokažemo s pomočjo principa za navidezno delo. Predstavljajmo si virtualni premik v vsakem sklepu  $\partial q_i$ , kar rezultira v virtualnem premiku vrha manipulatorja za  $\partial x_e$  in  $\partial \phi_e$  kot to kaže naslednja slika.



Slika 3.2: Sile in navori na vrhu in v sklepih manipulatorja.

Virtualni premiki so diferencialni premiki v sklepih manipulatorja, ki upoštevajo geometrične omejitve konstrukcije sistema. Virtualni premiki se razlikujejo od dejanskih premikov v tem, da morajo zadostiti samo omejitvam v geometriji in ne zadostijo ostalim zakonom mehanike in gibanja (dinamika, elastičnost sklepov in segmentov). Da jasno razločujemo virtualne pomike od dejanskih, smo uporabili znak  $\partial$  za parcialni odvod namesto  $d$ . Predpostavimo, da sklepni momenti  $\tau_i (i = 1, \dots, n)$  in končna sila vrha ter moment vrha  $\mathbf{f}_{n,n+1}$  in  $-\mathbf{N}_{n,n+1}$  delujejo na manipulator, medtem ko so sklepi in manipulator že v končni premaknjeni legi. Tedaj je virtualno delo  $D$  opravljeno zaradi sil in momentov podano s sledečo enačbo

$$\partial D = \tau_1 \cdot \partial q_1 + \dots + \tau_n \cdot \partial q_n - \mathbf{f}_{n,n+1}^T \cdot \partial x_e - \mathbf{N}_{n,n+1}^T \cdot \partial \phi_e. \quad (3.6)$$

Krajše to zapišemo z enačbo

$$\partial D = \boldsymbol{\tau}^T \partial \mathbf{q} - \mathbf{F}^T \partial \mathbf{p}. \quad (3.7)$$

Ker je manipulator v statičnem ravnotežju, pomeni, da je bilo virtualno delo opravljeno za dosego premikov in je zato virtualno delo nično. Glede na princip virtualnega dela, bo manipulator v ravnotežju samo tedaj, če bo virtualno delo enako nič, oziroma bo povzročilo virtualne premike, ki upoštevajo geometrične omejitve gibanja sklepov in omejitve okolice. Zavedati se moramo, da so virtualni premiki notranjih koordinat  $\partial \mathbf{q}$  in zunanjih  $\partial \mathbf{p}$  povezani preko manipulatorjeve Jacobijeve matrike  $\mathbf{J}$ , ki upošteva omejitve v geometriji. S pomočjo en. (3.5) in (3.7) pišemo enakost

$$\begin{aligned} \partial D &= \boldsymbol{\tau}^T \partial \mathbf{q} - \mathbf{F}^T \cdot \mathbf{J} \cdot \partial \mathbf{q} = (\boldsymbol{\tau} - \mathbf{J}^T \cdot \mathbf{F})^T \cdot \partial \mathbf{q} \\ \mathbf{J} &= (\mathbf{J}^T)^T. \end{aligned} \quad (3.8)$$

Virtualno notranje in zunanje delo daje razliko, ki pa mora biti enaka nič v statičnem ravnotežju, drugače bi se manipulator premikal.

Izraz vsebuje samo notranje premike  $\partial q_i$ , ki so medseboj neodvisni. Neodvisnost premikov  $\partial \mathbf{q}$  je veljavna samo tedaj, če bo opravljeno delo  $\partial D$  nično, torej gre za ravnotežje. Ker so premiki  $\partial q_i$  lahko poljubno izbrani, bo edino pripogoju, da je izraz v oklepaju enak nič, tudi delo  $\partial D = 0$ . Iz tega sledi

$$\boldsymbol{\tau} - \mathbf{J}^T \cdot \mathbf{F} = 0 \quad (3.9)$$

in dokaz enakosti

$$\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F}, \quad (3.10)$$

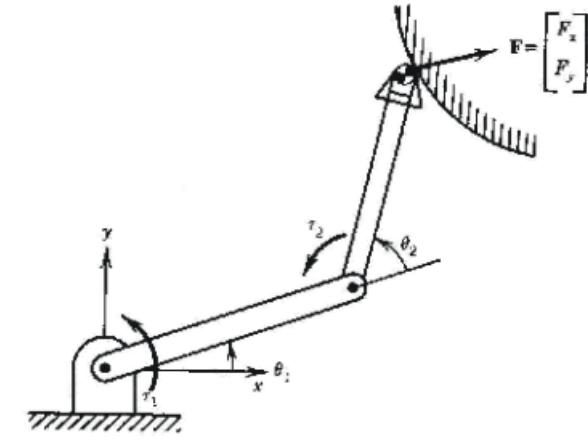
kar potrjuje pravilnost en. (3.4).

Primer:

Izračun  $\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F}$  za ravninski dvosegmentni manipulator.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \Theta_1} & \frac{\partial x}{\partial \Theta_2} \\ \frac{\partial y}{\partial \Theta_1} & \frac{\partial y}{\partial \Theta_2} \end{bmatrix}$$

$$\begin{aligned} x &= \rho_1 \cos \Theta_1 + \rho_2 \cos(\Theta_1 + \Theta_2) \\ y &= \rho_1 \sin \Theta_1 + \rho_2 \sin(\Theta_1 + \Theta_2). \end{aligned}$$



Slika 3.3: 2R dvosegmentni ravninski manipulator.

Na sliki 3.3 je podan manipulator z dvema prostostnima stopnjama v ravnini. Na vrhu robot razvije silo dotika  $\mathbf{F} = (F_x, F_y)^T$ . Najti moramo ekvivalentne momente v sklepih  $\boldsymbol{\tau} = (\tau_1, \tau_2)^T$ , ki ustrezajo sili  $\mathbf{F}$  vrha vrha. Sklepi so brez trenja. Napišemo  $\mathbf{J}$  za manipulator

$$\mathbf{J} = \begin{bmatrix} -\rho_1 \sin \Theta_1 - \rho_2 \sin(\Theta_1 + \Theta_2) & -\rho_2 \sin(\Theta_1 + \Theta_2) \\ \rho_1 \cos \Theta_1 + \rho_2 \cos(\Theta_1 + \Theta_2) & \rho_2 \cos(\Theta_1 + \Theta_2) \end{bmatrix}.$$

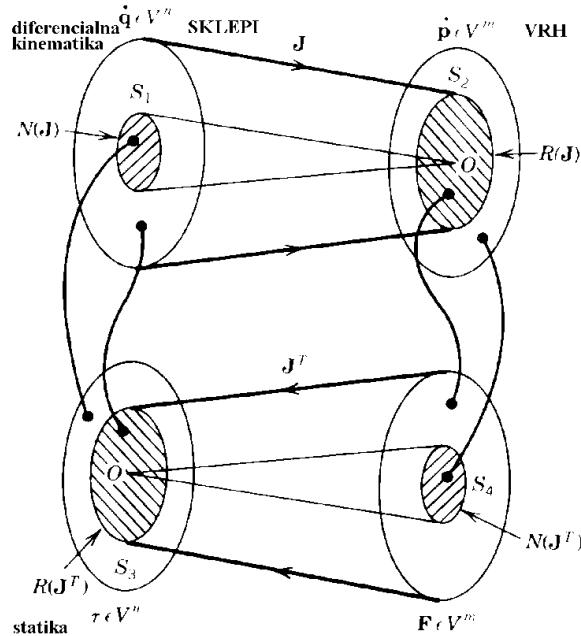
po en. (3.4)

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} -\rho_1 \sin \Theta_1 - \rho_2 \sin(\Theta_1 + \Theta_2) & \rho_1 \cos \Theta_1 + \rho_2 \cos(\Theta_1 + \Theta_2) \\ -\rho_2 \sin(\Theta_1 + \Theta_2) & \rho_2 \cos(\Theta_1 + \Theta_2) \end{bmatrix} \cdot \begin{bmatrix} F_x \\ F_y \end{bmatrix}.$$

## 3.2 DUALNOST KINEMATIKE IN STATIKE

Do sedaj smo dokazali, da so ustrezeni sklepni momenti oz. sile vezani na silo in moment vrha manipulatorja preko Jacobijeve matrike. Ta ista Jacobijeva matrika povezuje tudi infinitezimalne premike v notranjih koordinatah z ustreznimi premiki v zunanjih koordinatah. Spoznamo, da je statično razmerje sil in momentov tesno povezano s kinematiko malih premikov. Tukaj želimo pomen te povezanosti podrobnejše osvetliti.

Da nazorno pokažemo povezanost med statiko in kinematiko uporabimo sliko 3.4, ki jo bomo v nadaljevanju podrobnejše razčlenili in razložili.



Slika 3.4: Dualnost med diferencialno kinematiko in statiko.

Spomnimo se na že znani enačbi

$$\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}} \quad (3.11)$$

$$\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F} \quad (3.12)$$

$\mathbf{J}^T$  ... transponirana Jacobijeva matrika,

$\mathbf{J}$  ... Jacobijeva matrika.

Jacobijeva matrika predstavlja za dano konfiguracijo manipulatorja preslikavo med zunanjimi in notranjimi spremenljivkami, kar lepo kaže slika spondaj, glede na enačbo

$$d\mathbf{p} = \mathbf{J} \cdot d\mathbf{q} \quad (3.13)$$

$d\mathbf{p} = [dp_1 \dots dp_m]^T$  ... zunanje koordinate,

$d\mathbf{q} = [dq_1 \dots dq_n]^T$  ... notrane koordinate,

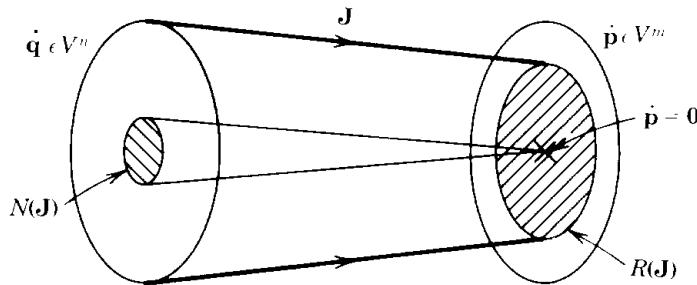
$n$  ... število sklepov,

$m$  ... število stopenj prostosti gibanja vrha manipulatorja.

Iz tega sledi dimenzija  $\mathbf{J}$ , ki je  $m \times n$ .

Če je  $n > m$  in matrika reda  $m \times n$ , bo  $(n - m)$  število poljubnih rešitev, kar pomeni, da ima manipulator  $(n - m)$  redundantnih stopenj prostosti. Toda  $\mathbf{J}$  povezuje tudi hitrosti vrha  $\dot{\mathbf{p}}$  in hitrosti v notranjih koordinatah  $\dot{\mathbf{q}}$ .

$$\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}} \quad (3.14)$$



Slika 3.5: Preslikava diferencialne kinematike.

Obe zadnji enačbi ponazorimo z zgornjo sliko. Podprostor  $R(\mathbf{J})$  podaja preslikavo prostora vseh hitrosti  $\dot{\mathbf{p}}$ , ki jih lahko proizvede vrh manipulatorja ob premikanju  $n$  sklepov za dano konfiguracijo manipulatorja. Če je rang matrike  $\mathbf{J}$  poln (vse vrstice imajo vse člene), bo področje  $R(\mathbf{J})$  prekrilo celotno področje vektorskega prostora  $V^m$ . V nasprotnem obstaja vsaj ena smer, v katero se vrh manipulatorja ne more premakniti. Podprostor  $N(\mathbf{J})$  preslikave je nični prostor te preslikave. Katerikoli element v  $N(\mathbf{J})$  prostoru se preslika v  $\mathbf{J} \cdot \dot{\mathbf{q}} = 0$ ,  $\dot{\mathbf{p}} = 0$  v prostoru  $V^m$ . Zaradi tega se katerikoli vektor hitrosti notranjih koordinat  $\dot{\mathbf{q}}$  (hitrost v sklepih), ki pripada ničnemu prostoru  $N(\mathbf{J})$  preslika v nično hitrost gibanja vrha manipulatorja, to je  $\dot{\mathbf{p}} = 0$  (ponazorimo z gibanjem komolca, dlan se ne premika, torej  $\dot{\mathbf{p}} = 0$ ).

Če ima Jacobijeva matrika poln rang, bo dimenzija ničnega prostora  $N(\mathbf{J})$  enaka številu redundantnih stopenj manipulatorja, to je  $(n - m)$ .

Če je Jacobijeva matrika degenerirana, to je nima polnega ranga, bo dimenzija prostora hitrosti  $R(\mathbf{J})$ , to je  $\dim N(\mathbf{J})$ , zmanjšana in za isto število bo dimenzija ničnega prostora  $N(\mathbf{J})$  povečana. Vsota dimenzijs obih prostorov  $N(\mathbf{J})$  in  $R(\mathbf{J})$  je vedno  $n$ , enaka številu sklepov.

$$\dim R(\mathbf{J}) + \dim N(\mathbf{J}) = n. \quad (3.15)$$

Če je  $\dot{\mathbf{q}}^*$  rešitev enačbe  $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$  in  $\dot{\mathbf{q}}_0$  vektor, ki se nahaja v ničtem prostoru hitrosti  $N(\mathbf{J})$ , potem je vektor  $\dot{\mathbf{q}} = \dot{\mathbf{q}}^* + k \cdot \dot{\mathbf{q}}_0$  tudi rešitev enačbe  $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$

. Pri tem je  $k$  poljuben skalar, kar sledi iz naslednje enačbe

$$\mathbf{J} \cdot \dot{\mathbf{q}} = \mathbf{J} \cdot \dot{\mathbf{q}}^* + k \cdot \mathbf{J} \cdot \dot{\mathbf{q}}_0 = \mathbf{J} \cdot \dot{\mathbf{q}}^* = \dot{\mathbf{p}}. \quad (3.16)$$

Ker lahko  $k \cdot \dot{\mathbf{q}}_0$  izbiramo poljubno v  $N(\mathbf{J})$  prostoru nične hitrosti vrha, pomeni, da obstaja neskončno rešitev za to podano, kolikor seveda ni dimenzija  $N(\mathbf{J})$  enaka nič. Sledi, da je število nedefiniranih rešitev zadnje enačbe, na sliki črtkani del, enako dimenziji  $N(\mathbf{J})$  prostora.

Na sliki 3.4 je zgornji del namenjen ponazoritvi preslikave kinematike, to je vektorskega prostora  $\dot{\mathbf{q}} \in V^n$ , preko  $\mathbf{J}$  v vektorski prostor  $\dot{\mathbf{p}} \in V^m$ . Spoznamo, da je prostor hitrosti vrha manipulatorja  $N(\mathbf{J})$  manjši od celotnega možnega prostora  $V^m$  ter vključuje tudi nične hitrosti, kar so preslikane v  $V^n$  prostor  $N(\mathbf{J})$ . Tu so vektorji  $\dot{\mathbf{q}}$ , ki ne proizvedejo gibanja vrha manipulatorja, čeprav se sklepi gibljejo. Če prostor  $N(\mathbf{J})$  ni prazen, bo enačba kinematike  $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$  imela neskončno možnih rešitev, pri katerih bo hitrost vrha manipulatorja blizu nič, torej gre za singularnost.

To kar smo do sedaj pokazali, je bila linearna preslikava  $\dot{\mathbf{q}}$  v  $\dot{\mathbf{p}}$  (puščice na sliki 3.4) v prostoru kinematike. Druga linearna preslikava je vezana na statiko in zvezo notranjih sil in zunanjih sil in momentov po naslednji enačbi

$$\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F} \quad (3.17)$$

Za razliko od kinematične preslikave slednja poteka od statičnih sil  $\mathbf{F}$ , torej od  $m$  dimenzionalnega vektorskega prostora  $V^m$ , ki je združen s koordinatami vrha manipulatorja, proti  $n$  dimenzionalnem vektorskem prostoru  $V^n$ , vezanem na koordinate sklepov. Zaradi tega so momenti v sklepih  $\boldsymbol{\tau}$  vedno določeni enoznačno za vsako poljubno silo vrha manipulatorja  $\mathbf{F}$  ( $\mathbf{F} \in V^m$ ). Nasprotno, za dane momente v sklepih ustrezena ravnotežna sila  $\mathbf{F}$  vrha manipulatorja ne obstaja vedno. Podobno kot pri kinematiki, bomo tudi tu definirali  $N(\mathbf{J}^T)$  nični prostor sil vrha manipulatorja, v katerem so tiste sile  $\mathbf{F}$  vrha manipulatorja, ki ne potrebujejo momentov v sklepih, da bi se manipulator lahko upiral bremenu ali sili  $\mathbf{F}$ . Tem silam se zoperstavi sama struktura manipulatorja! Prav tako definiramo prostor momentov sklepov  $R(\mathbf{J}^T)$  v prostoru momentov sklepov, kjer so vsi tisti gonalni momenti  $\boldsymbol{\tau}$ , ki lahko uravnotežijo silo vrha manipulatorja. Področji ničnih prostorov  $N(\mathbf{J}^T)$  in  $R(\mathbf{J}^T)$  sta tesno povezani. Glede na pravila linearne algebri je nični prostor  $N(\mathbf{J})$  ortogonalni komplement prostora vrednosti  $R(\mathbf{J}^T)$ . Namreč, če imamo npr. nek od 0 različen vektor, ki se nahaja v  $N(\mathbf{J})$ , potem ta ne more pripadati tudi prostoru vrednosti  $R(\mathbf{J})$  in velja tudi obratno. Če s  $S_1$  označimo ortogonalni komplement od  $N(\mathbf{J}^T)$  tedaj prostor vrednosti  $R(\mathbf{J}^T)$

postane identičen z  $S_1$ , kot je to prikazano na sliki 3.4. Tudi prostor  $S_3$ , ki je slično ortogonalni komplement  $R(\mathbf{J}^T)$ , je identičen z  $N(\mathbf{J})$ . Za ta prostor povzamemo, da se v smeri, v kateri hitrosti sklepov ne povzročijo nobene hitrosti vrha manipulatorja, sklepni momenti  $i$  ne morejo uravnovežiti sile vrha manipulatorja. Torej, da vzdržujemo stalen položaj manipulatorja in sklepov, morajo biti sklepni momenti v tem prostoru nični.

Podobna odvisnost je tudi v prostoru stanj vrha manipulatorja  $V^m$ . Področje stanj  $R(\mathbf{J})$  je ortogonalni komplement k ničnemu prostoru  $N(\mathbf{J}^T)$ . Torej prostor  $S_2$  na sliki 3.4 je identičen z  $N(\mathbf{J}^T)$  in prostor  $S_4$  je identičen prostoru  $R(\mathbf{J})$ . Zaradi tega niso potrebni sklepni gonični momenti za uravnoteženje sile vrha manipulatorja, če zunanje sile delujejo v smeri v kateri manipulatorjev vrh ne moremo premakniti s sklepi manipulatorja. Če sila  $\mathbf{F}$  deluje v smeri, v kateri lahko vrh manipulatorja premikamo, potem jo moramo uravnotežiti z momenti sklepov. Ko postane Jacobijeva matrika degenerirana, oziroma je manipulator v singularni konfiguraciji, nični prostor  $N(\mathbf{J}^T)$  ne bo prazen. To pomeni, da je lahko še vedno del zunanje sile uravnotežen preko konstrukcije manipulatorja.

Iz do sedaj povedanega lahko sklepamo, da so trenutne kinematične razmere in trenutne statične razmere tesno povezane. Temu odnosu pravimo tudi dualnost med kinematiko in statiko.

### 3.3 TRANSFORMACIJA SIL IN MOMENTOV

Pokazali smo, da moremo statične sile in momente analizirati kot trenutni kinematični problem ob uporabi dualnosti kinematika - statika in ustreznih preslikav. Ko nimamo zapisane trenutne kinematične enačbe, lahko dobimo odnose med statičnimi silami in momenti preko dualnostnih pravil, namesto da bi izpeljevali relacije preko strukture robota. Ta merila dualnosti se dajo uporabiti in razširiti za reševanje danih problemov. V ta namen bomo dualnost najprej **pospološili** in nato uporabili za reševanje problemov v robotih ( $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$ ).

Pospološitev začnemo z modifikacijo vektorja  $\mathbf{q}$ , ki definira spremenljivke sklepov. Upoštevajmo  $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$  sedaj kot nek nabor neodvisnih generaliziranih koordinat, ki določajo nek mehanski sistem v celoti. Spremenljivke sklepov so primer takšnega zapisa, saj so neodvisne in v celoti opisujejo stanje sistema. Naj bo  $\mathbf{Q} = [Q_1, Q_2, \dots, Q_n]^T$  vektor generaliziranih sil, ki pripadajo generaliziranim koordinatam  $\mathbf{q}$ . Nadalje pred-

postavimo, da obstaja dualnost generaliziranih koordinat, ki jih opišemo s  $\mathbf{p} = [p_1, p_2, \dots, p_m]^T$ . Koordinate ni potrebno, da so morejo biti kompletne (vse), ker vse prostostne stopnje sistema ni nujno, da so določene z naborom koordinat. Niti pozicija, niti orientacija nekega vrha manipulatorja ne določata celotne konfiguracije robota, če ima manipulator redundantne stopnje prostosti. V tem primeru neskončno število kombinacij ustreza isti končni orientaciji in poziciji vrha manipulatorja. Obravnavamo trenutek, ko statične sile in momenti delujejo na sistem, ki je v položaju opisan s  $\mathbf{q}$ . Predpostavimo, da so sile in momenti, ki so označeni s  $\mathbf{P} = [P_1, P_2, \dots, P_m]^T$  podani glede na  $\mathbf{p}$  generalizirane koordinate.

V nadaljevanju želimo transformirati sile in momente označene s  $\mathbf{P}$  iz koordinatnega sistema  $\mathbf{p}$  v  $\mathbf{q}$ .

V ta namen uporabimo princip virtualnih premikov  $\partial\mathbf{p}$ . Ker  $\mathbf{q}$  koordinate predstavljajo kompleten nabor generaliziranih koordinat, lahko opišemo pomike v poljubnem stanju sistema. Zaradi tega morajo biti premiki opisani s  $\mathbf{p}$  izraženi v  $\mathbf{q}$  koordinatah. Če odvajamo to funkcijo, bomo lahko pokazali virtualne pomike  $\partial\mathbf{p}$  z virtualnimi premiki  $\partial\mathbf{q}$ , tako da velja

$$d\mathbf{p} = \mathbf{J} \cdot d\mathbf{q}. \quad (3.18)$$

Tu pomeni  $\mathbf{J}$   $[m \times n]$  Jacobijevo matriko, ki je vezana na koordinatno transformacijo  $\mathbf{q} \rightarrow \mathbf{p}$ . Sedaj bomo pokazali, da se generalizirane sile  $\mathbf{P}$  transformirajo v generalizirane sile  $\mathbf{Q}$ , ki delujejo po koordinatah  $\mathbf{q}$  po obrazcu

$$\mathbf{Q} = \mathbf{J}^T \cdot \mathbf{P}. \quad (3.19)$$

Tu pomeni  $\mathbf{J}^T$  transformirano Jacobijevo matriko. Zgornjo enačbo dokažemo na podoben način, kot smo to že storili pri dokazovanju transformacije sile vrha manipulatorja v pogonske momente (preko virtualnega dela  $D$ ).

Določimo  $\mathbf{Q}$  silam ekvivalentne  $\mathbf{P}$  sile tako, da bomo obravnavali ravnotežje teh naborov sil, to je nabora  $\mathbf{P}$  in  $\mathbf{Q}$ . Tak sistem bo v ravnotežju samo, če je virtualno delo opravljeno z zunanjimi silami  $\mathbf{P}$  in ekvivalentnimi silami  $\mathbf{Q}$  enako glede na poljubno izbrane virtualne premike, ki ustrezajo geometrijski odvisnosti

$$d\mathbf{p} = \mathbf{J} \cdot d\mathbf{q}. \quad (3.20)$$

Če je tako, sledi

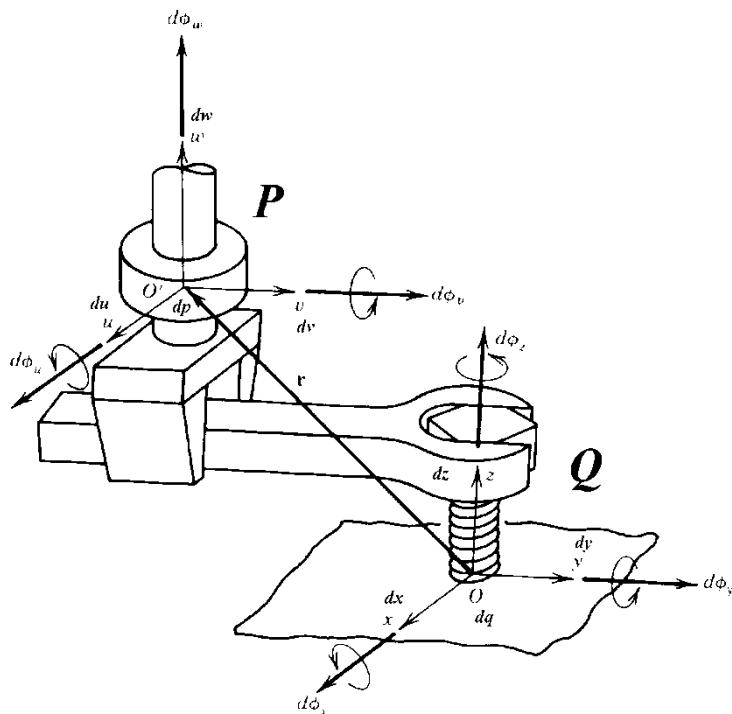
$$\partial D = \mathbf{Q}^T \cdot \partial\mathbf{q} - \mathbf{P}^T \cdot \partial\mathbf{p} = (\mathbf{Q} - \mathbf{J}^T \cdot \mathbf{P})^T \cdot \partial\mathbf{q} \quad (3.21)$$

Če naj bo delo enako nič torej  $\partial D = 0$ , bo to možno samo pri vrednosti oklepaja enaki nič. S tem je dokazana enakost

$$\mathbf{Q} = \mathbf{J}^T \cdot \mathbf{P}. \quad (3.22)$$

Primer:

Na sliki 3.6 vidimo robota, ki s ključem v prijemalu privija vijak. Zato mora izvajati silo in moment na vijak in seveda ti veličini meriti. Robot ima v ta namen vgrajen v zapestju senzor, ki meri tri sile in tri momente glede na koordinatni sistem zapestja. Zanima nas, kako izračunati sile in momente na mestu vijaka, če poznamo geometrijo naloge in izmerjene vrednosti velikosti sil in momentov na vrhu robota.



Slika 3.6: Robot s ključem in koordinatni sistemi.

Sile in momente lahko podamo na dva načina: glede na koordinatni sistem  $\mathbf{o}$ , osi  $x, y, z$ , oziroma glede na koordinate sistema  $\mathbf{o}'$ , osi  $u, v, w$ . Ta dva  $\sigma$  in  $\sigma'$  koordinatna sistema sta na sliki 3.6 v trenutku paralelna in središče koordinatnega sistema  $\mathbf{o}'$  je podano s  $3 \times 1$  pozicijskim vektorjem  $\mathbf{r} = [r_x, r_y, r_z]^T$ , ki je definiran glede na  $\mathbf{o}$  koordinatni sistem.

V primeru moremo smatrati zapestje, prijemo, ključ in vijak kot dele nekega enotnega togega sistema. Infinitezimalne translacije in rotacije togega telesa so opisane s 6-dimenzionalnim vektorjem  $d\mathbf{q} = [dx, dy, dz, d\phi_x, d\phi_y, d\phi_z]^T$  glede na koordinatni sistem vijaka  $\mathbf{o}$ , osi  $x, y, z$  in z drugim vektorjem  $d\mathbf{p} = [du, dv, dw, d\phi_u, d\phi_v, d\phi_w]^T$  glede na  $\mathbf{o}'$ , osi  $u, v, w$ , koordinatni sistem. Za sliko ugotovimo koeficiente transformacije, ki preslikajo  $d\mathbf{q}$  v  $d\mathbf{p}$

$$d\mathbf{p} = \begin{bmatrix} du \\ dv \\ dw \\ d\phi_u \\ d\phi_v \\ d\phi_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & r_z & -r_y \\ 0 & 1 & 0 & -r_z & 0 & r_x \\ 0 & 0 & 1 & r_y & -r_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \\ d\phi_x \\ d\phi_y \\ d\phi_z \end{bmatrix}.$$

V skladu z infinitezimalnimi premiki vektorjev bomo označili tudi sile in momente s 6-dimenzionalnim vektorjem  $\mathbf{Q} = [F_x, F_y, F_z, M_x, M_y, M_z]^T$ . To so sile in momenti glede na  $\mathbf{o}, x, y, z$  koordinatni sistem. S  $\mathbf{P} = [F_u, F_v, F_w, M_u, M_v, M_w]^T$  označimo sile in momente podane v  $\mathbf{o}', u, v, w$  koordinatnem sistemu. Če sedaj uporabimo relacijo za transformacijo sil in momentov, torej enačbo  $\mathbf{Q} = \mathbf{J}^T \cdot \mathbf{P}$  ( $\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F}$ ) pišemo

$$\boldsymbol{\tau} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -r_z & r_y & 1 & 0 & 0 \\ r_z & 0 & -r_x & 0 & 1 & 0 \\ -r_y & r_x & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_u \\ F_v \\ F_w \\ M_u \\ M_v \\ M_w \end{bmatrix}.$$

Ta izraz podaja želeno transformacijo, torej način, kako se preslikajo sile in momenti v zapestju v sile in momente v vrhu orodja.

## 3.4 TOGOST

V nadaljevanju želimo pobliže spoznati togost robotskega mehanizma. Predstavljammo si odprto kinematično verigo, ki jo tvori robotski manipulator. Če na vrhu robota delujemo z neko silo  $\mathbf{F} = (F_x, F_y, F_z)$ , se bo manipulator zaredi zunanje sile podal ali odklonil za nek  $\Delta\mathbf{p} = (\Delta x, \Delta y, \Delta z, \Delta\phi_x, \Delta\phi_y, \Delta\phi_z)$ . Velikost  $\Delta\mathbf{p}$  bo proporcionalna togosti in velikosti sile  $\mathbf{F}$ . Torej, čim večja bo sila in čim manjša bo togost, tem večji bo  $\mathbf{p}$ . Podajanje robota pod bremenom odloča o točnosti pozicioniranja, pomembno pa je tudi v nalogah,

kjer robotski manipulator deluje v dotiku z okoljem. Zaradi tega je podajanje robotskega mehanizma pomembna lastnost manipulatorja. Vsekakor je možnost regulacije togosti manipulatorja pomembna odlika, ker s tem lahko spremojamo lastnosti manipulatorja, tako da se ujemajo z zahtevami pri opravljanju kompleksnih robotskih nalog. Pri vsakem robotskem opravilu so sila in moment dotika, zahtevana točnost ter dopustno podajanje pomembni parametri in določeni v predpisanim področju (primeri: nanašanje lepila, brušenje, precizna montaža, paletno zlaganje). V tem razdelku bomo razložili osnovne pojme in opise, ki zadevajo togost manipulatorja.

Robot sestavlja motorni pogon z regulatorji ter predležje, ki prenaša navor ali silo motorja na segmente. Elastičnost (elastično podajanje in majhna togost) nastopa na različnih mestih robotskega mehanizma, kot na primer v motorju, elementih predležja, sklepu in vzdolž segmenta. V nadaljevanju bomo predpostavili, da so segmenti popolnoma togii. Vso preostalo elastičnost, ki definira togost manipulatorja, to je pogonski sistem in predležje sklepa, bomo združili v konstanti elastičnosti sklepa  $k_i$  i-tega sklepa. Definiramo vzmeti analogno odvisnost ( $M_i$  je moment v sklepu, ki povzroči podajanje notranje spremenljivke  $\Delta q_i$ ).

$$M_i = k_i \cdot \Delta q_i \quad (3.23)$$

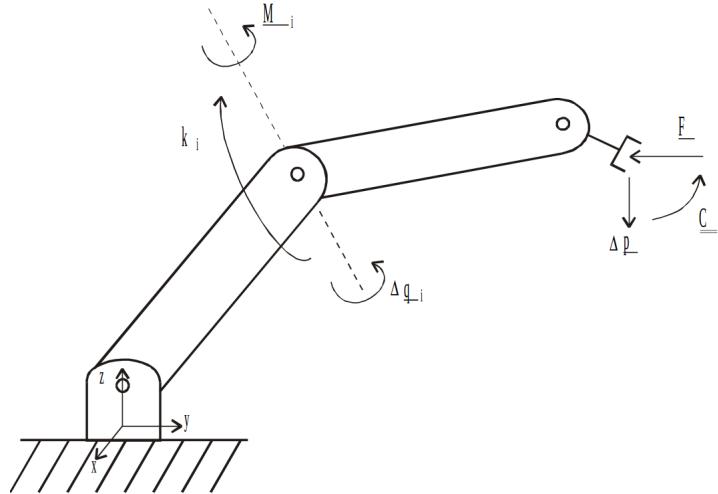
Pri določanju togosti vrha manipulatorja privzamemo, da je togost sklepa definirana po en. (3.23). Pri tem smo predpostavili, da so segmenti popolnoma togii. Teorija manipulatorjev z elastičnimi segmenti zahteva posebno obravnavo in presega obseg učbenika.

Po en. (3.23) smo definirali elastičnost za vsak posamezen sklep. Ob predpostavki, da so segmenti togii, bodo elastičnosti sklepov definirale podajnost vrha robota. Vrh robota se bo elastično podal za  $\Delta p$  in tako popustil zunanji sili  $\Delta F$ . To pa pomeni, da se sumirajo podajanja sklepov  $q_i$  in se glede na strukturo robotskega mehanizma kažejo v podajanju  $\Delta p$ . Razmere za lažjo predstavo prikažemo s pomočjo slike 3.7. Ker zaradi nazornosti in neposrednosti zanemarimo gravitacijske sile, kakor tudi trenja, bomo izpeljavo elastičnega podajanja in definicijo podajnosti robota najlaže navezali na povezavo med silo vrha robota in momenti v sklepih robota

$$M = J^T \cdot F \quad (3.24)$$

V vsakem sklepu je moment/sila vezan preko enačbe (3.23) na podajanje sklepa. Torej za celoten manipulator en. (3.23) zapišemo v naslednji obliki

$$M = K \cdot \Delta q \quad (3.25)$$



Slika 3.7: Podajnost vrha robotskega mehanizma.

Tu je  $\mathbf{K}$  diagonalna matrika elastičnosti sklepov in razsežnosti  $n \times n$  ( $n$  predstavlja število sklepov)

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 & \dots & 0 \\ 0 & k_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_n \end{bmatrix}. \quad (3.26)$$

Iz prejšnjega razdelka povzamemo enačbo (3.14), ki povezuje premike sklepov  $\Delta \mathbf{q}$  s premiki vrha manipulatorja  $\Delta \mathbf{p}$

$$\Delta \mathbf{p} = \mathbf{J} \cdot \Delta \mathbf{q}. \quad (3.27)$$

Izračunati želimo premike vrha manipulatorja  $\Delta \mathbf{p}$  v odvisnosti od  $\mathbf{F}$  ob upoštevanju razmer, podanih z enačbama (3.24) in (3.25). V ta namen izenačimo (3.24) in (3.25) ter izrazimo  $\Delta \mathbf{q}$

$$\mathbf{J}^T \cdot \mathbf{F} = \mathbf{K} \cdot \Delta \mathbf{q} \quad (3.28)$$

$$\mathbf{K}^{-1} \cdot \mathbf{J}^T \cdot \mathbf{F} = \Delta \mathbf{q}. \quad (3.29)$$

Sklepov brez elastičnosti fizikalno ni moč izdelati in zaradi tega so v matriki  $\mathbf{K}$  vsi diagonalni členi  $k_i \neq 0$ . Torej je vedno mogoče izračunati inverzno matriko  $\mathbf{K}^{-1}$ . Izraz za  $\Delta \mathbf{q}$  (3.29) vstavimo v izraz (3.27)

$$\Delta \mathbf{p} = \mathbf{J} \cdot \mathbf{K}^{-1} \cdot \mathbf{J}^T \cdot \mathbf{F} \quad (3.30)$$

$$\Delta \mathbf{p} = \mathbf{C} \cdot \mathbf{F}. \quad (3.31)$$

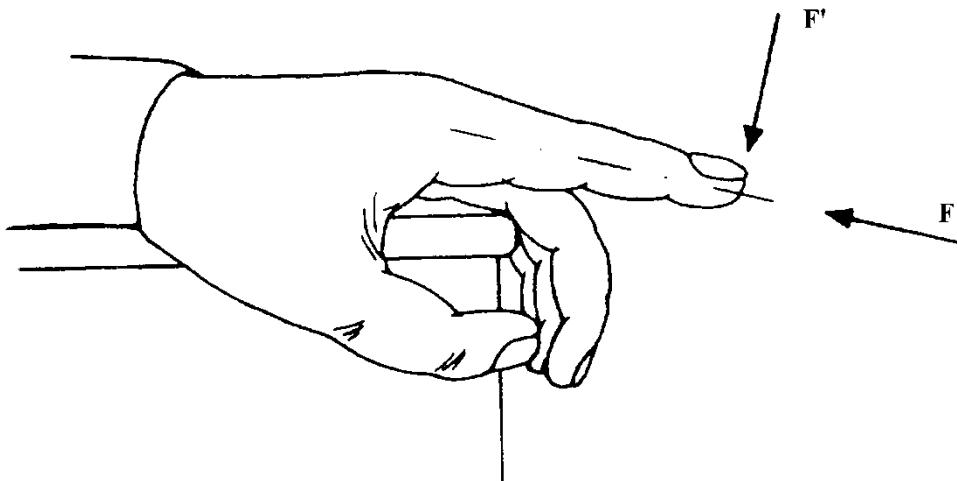
Zanimivo je, da ta enačba kaže linearno odvisnost  $\Delta\mathbf{p}$  od  $\mathbf{F}$ .  $\mathbf{C}$  je tu označen kot produkt

$$\mathbf{C} = \mathbf{J} \cdot \mathbf{K}^{-1} \cdot \mathbf{J}^T. \quad (3.32)$$

$\mathbf{C}$  poimenujemo matriko podajnosti (compliance) vrha manipulatorja. Iz enačb, ki definirajo matriko podajnosti izhaja, da bo matrika podajnosti  $\mathbf{C}$  vedno simetrična. Če je Jacobijeva matrika kvadratna in polnega ranga, moremo matriko podajnosti invertirati in tako izračunati vektor sile vrha manipulatorja  $\mathbf{F}$  za podano podajanje  $\Delta\mathbf{p}$

$$\mathbf{F} = \mathbf{C}^{-1} \cdot \Delta\mathbf{p}. \quad (3.33)$$

Inverzno matriko matrike podajnosti  $\mathbf{C}$ , to je  $\mathbf{C}^{-1}$ , označimo za matriko togosti vrha manipulatorja. Tako matrika podajnosti  $\mathbf{C}$  kot matrika togosti  $\mathbf{C}^{-1}$  vrha manipulatorja po en. (3.31) ozziroma (3.33) vključujeta Jacobijevu matriko, ki se spreminja s položajem robotskega mehanizma. Tako sta obe matriki odvisni od položaja robota. Posledično ugotavljamo tudi, da se bo po enačbi (3.31) za konstantno silo  $\mathbf{F}$  (nespremenljivo breme) pri različnih položajih robotskega mehanizma spremenjalo podajanje  $\Delta\mathbf{p}$  (odstopanje in točnost robota). Analogno se bo po enačbi (3.33) za dani konstantni vektor  $\Delta\mathbf{p}$  pri različnih položajih robota spremenjala tudi sila.



Slika 3.8: Singularen položaj kazalca.

Opozoriti velja še na primere, ko manipulator zavzame singularno lego. (Primer: popolnoma iztegnjen kazalec, roka v zapestju ob robu mize in sila  $\mathbf{F}$  deluje tlačno v osni smeri, slika 3.8) Tedaj sile na vrhu robota ne prestrežajo sklepni motorji in lahko velja  $\mathbf{M} = 0$ . V takšni singularni legi postane Jacobijeva matrika degenerirana. Seveda silo vrha tedaj prestreže struktura

mehanizma in če je ta popolnoma toga, bo podajanje nično oziroma togost neskončna. Dosedanja razprava nakazuje, da bo podajanje odvisno tudi od smeri sile na vrhu manipulatorja in ne samo od položaja robotskega mehanizma. Dokaz za to je na dlani. Če v zgornjem primeru iztegnjeni kazalec obremenimo na vrhu s silo  $\mathbf{F}$ , ki bo pravokotna na smer iztegnjenega kazalca, bodo sklepni »motorji« morali v celoti prevzeti obremenitev ter bo togost minimalna in zaradi tega podajnost ter podajanje maksimalno. Smeri, ki ju tvorita maksimalna in minimalna podajnost, bomo imenovali glavne smeri. Zanima nas, kako najti za dani manipulator in njegov položaj glavni smeri, ki sta si očitno med seboj pravokotni. Poglejmo preprost primer robotskega manipulatorja z dvema rotacijskima sklepoma (slika 3.7).

Po en. (3.32) izračunamo matriko podajnosti  $\mathbf{C}$  v obliki

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \quad (3.34)$$

$$C_{11} = \frac{(a_1 S_1 + a_2 S_{12})^2}{k_1} + \frac{a_2^2 S_{12}^2}{k_2} \quad (3.35)$$

$$C_{12} = C_{21} = \frac{-(a_1 C_1 + a_2 C_{12})(a_1 S_1 + a_2 S_{12})}{k_1} - \frac{a_2^2 C_{12} S_{12}}{k_2} \quad (3.36)$$

$$C_{22} = \frac{(a_1 C_1 + a_2 C_{12})^2}{k_1} + \frac{a_2^2 C_{12}^2}{k_2} \quad (3.37)$$

$a_1$  in  $a_2$  predstavlja dolžini prvega ter drugega segmenta,  $k_1$  in  $k_2$  pa elastičnosti v prvem in drugem sklepu. Opozorimo naj, da velja v primeru avninskega robotskega manipulatorja  $\Delta\mathbf{p} = [\Delta x, \Delta y]^T$  in  $\mathbf{F} = [F_x, F_y]^T$ , tako lahko v enačbi za  $\mathbf{C}$  uporabimo  $\mathbf{J}$  matriko reda  $2 \times 2$ .

Za dano matriko podajnosti (manipulator v zamrznjenem položaju) želimo najti maksimalno in minimalno podajanje. To bomo določili tako, da izberemo enotsko sil

$$|\mathbf{F}|^2 = 1 = \mathbf{F}^T \cdot \mathbf{F} = 1. \quad (3.38)$$

Ne glede na predznak  $\Delta p$  želimo poiskati minimalno in maksimalno vrednost podajanja in bomo zaradi tega računali s kvadratom izraza (3.31)

$$|\Delta\mathbf{p}|^2 = \Delta\mathbf{p}^T \cdot \Delta\mathbf{p} = \mathbf{F}^T \cdot \mathbf{C}^T \cdot \mathbf{C} \cdot \mathbf{F} = \mathbf{F}^T \cdot \mathbf{C}^2 \cdot \mathbf{F}. \quad (3.39)$$

Vemo, da je  $\mathbf{C}$  simetrična matrika. Da najdemo minimalne in maksimalne vrednosti izraza (3.39), bomo računali vezani ekstrem  $\lambda$  in definirali naslednji izraz

$$L = \mathbf{F}^T \cdot \mathbf{C}^2 \cdot \mathbf{F} - \lambda(\mathbf{F}^T \cdot \mathbf{F} - 1). \quad (3.40)$$

Potrebni pogoj, da zavzame absolutna kvadratna vrednost podajanja vrha manipulatorja minimalno ali maksimalno vrednost, je dan z naslednjim izrazom

$$\frac{\partial L}{\partial \lambda} = 0 : -\mathbf{F}^T \cdot \mathbf{F} + 1 = 0. \quad (3.41)$$

Ta pogoj je izpolnjen, saj to predpostavlja izraz (3.38). Drugi pogoj, ki mora biti zadovoljen, podaja naslednja enačba

$$\frac{\partial L}{\partial F} = 0 : \mathbf{C}^2 \cdot \mathbf{F} - \lambda \cdot \mathbf{F} = 0. \quad (3.42)$$

To enačbo moremo pisati tudi v obliki

$$\mathbf{C}^2 \cdot \mathbf{F} = \lambda \cdot \mathbf{F}, \quad (3.43)$$

ki pravi, da je linearna preslikava vektorja  $\mathbf{F}$  za matriko  $\mathbf{C}^2$  kar enaka vektorju  $\mathbf{F}$ , ki je pomnožen s konstanto  $\lambda$ . Torej je vektor  $\mathbf{F}$  invariantna preslikava  $\mathbf{C}^2$ , to pa pomeni, da je  $\lambda$  lastna vrednost matrike  $\mathbf{C}^2$ . Iz tega sledi, da je problem za minimalno in maksimalno vrednost podajanja, vezan na izračun lastnih vrednosti matrike  $\mathbf{C}^2$

$$\frac{\lambda_{max}}{\lambda_{min}} = \frac{1}{2} \left[ b_1 + b_2 \pm \sqrt{(b_1 + b_2)^2 + 4(b_3^2 - b_1 b_2)} \right], \quad (3.44)$$

kjer je kvadrat matrike podajnosti podan ko

$$\mathbf{C}^2 = \begin{bmatrix} b_1 & b_3 \\ b_3 & b_2 \end{bmatrix}. \quad (3.45)$$

Zavedati se moramo, da so lastne vrednosti matrike  $\mathbf{C}$  vedno pozitivne, kajti posamezne podajnosti sklepov so vselej pozitivne. Ko poznamo lastne vrednosti in ob upoštevanju en. (3.43), lahko en. (3.39) napišemo v naslednji obliki

$$|\Delta \mathbf{p}|^2 = \mathbf{F}^T \cdot \mathbf{C}^2 \cdot \mathbf{F} = \mathbf{F}^T \cdot \lambda \cdot \mathbf{F} = \lambda \quad (3.46)$$

$$|\Delta \mathbf{p}|^2 = \lambda \quad (3.47)$$

Iz en. (3.46) sledi, da so maksimalna in minimalna podajanja enaka  $\sqrt{\lambda_{max}}$  in  $\sqrt{\lambda_{min}}$ . Smeri maksimalnega in minimalnega podajanja sta podani kot smeri  $e_{max}$  in  $e_{min}$

$$\Delta \mathbf{p}_{max} = |\lambda_{max}| \cdot e_1 = |\lambda_{max}| \cdot e_{max} \quad (3.48)$$

$$\Delta \mathbf{p}_{min} = |\lambda_{min}| \cdot e_2 = |\lambda_{min}| \cdot e_{min}. \quad (3.49)$$

Ti usmeritvi sta, kot smo že ugotovili, pravokotni. Smeri  $e_{max}$  in  $e_{min}$  smo imenovali glavni smeri in vzporedno z obema vektorjema izberemo glavni koordinatni sistem, ki ima glavne koordinatne osi. Če matriko podajnosti izrazimo v glavnem koordinatnem sistemu, postane ta diagonalna. S pomočjo vektorjev glavnega koordinatnega sistema  $e_1$  in  $e_2$  definiramo matriko  $2 \times 2$  (za večdimenzijski primer bo to matrika  $n \times n$ )

$$\mathbf{E} = |e_1, e_2|. \quad (3.50)$$

Matriko podajnosti transformiramo v diagonalno matriko  $\mathbf{C}^*$  tako, da jo množimo po naslednjem pravilu

$$\mathbf{C}^* = \mathbf{E}^T \cdot \mathbf{C} \cdot \mathbf{E} = \begin{bmatrix} \sqrt{\lambda_{max}} & 0 \\ 0 & \sqrt{\lambda_{min}} \end{bmatrix}. \quad (3.51)$$

Pri tem tudi velja, da je  $\mathbf{E}^T = \mathbf{E}$ , ker je  $\mathbf{E}$  ortogonalna matrika. Pravkar pokazano transformacijo, kjer transformiramo matriko glede na glavne koordinate, imenujemo kanonična transformacija. Treba je tudi razumeti, da se bo, če posredujemo obremenitev na vrhu robota v smeri glavne osi, podajanje izvajalo v smeri te glavne osi in ima velikost  $|\lambda| \cdot \| \mathbf{F} \|$ .

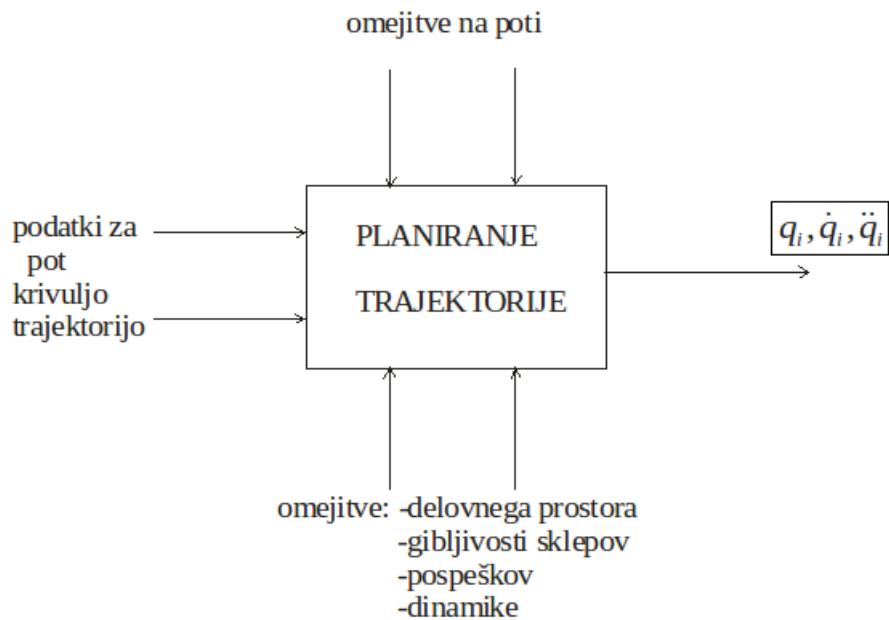
## Poglavlje 4

# GENERIRANJE TRAJEKTORIJE

### 4.1 INTERPOLACIJA

V tem poglavju želimo spoznati, kako izračunamo za robot v prostoru delovno trajektorijo, ki bo opisala naše zahteve. Pod trajektorijo razumemo časovno zaporedje pozicij in orientacij vrha manipulatorja in s tem sta podana tudi hitrost in pospešek v vsaki prostostni stopnji. Problem podajanja trajektorije zahteva povezavo človek operater in robot. Naš namen je določiti trajektorijo v prostoru. Da to opravilo za operaterja poenostavimo, mora biti zapis takšen, da ne zahteva od operaterja pisanja komplikiranih funkcij ali preračunavanja enačb, temveč lahko le preprosto opiše zaporedne lege v prostoru in zatem robot sam določi vse ostale zahtevane parametre. Npr., uporabnik predpiše samo nekaj leg za vrh prijemala v prostoru in robot sam določi vse ostale trajektorije glede na želeno hitrost oziroma čas giba. Zanima nas tudi, kako bodo tako podane trajektorije zapisane in predstavljene v računalniku, potem ko smo jih načrtovali. Tu se tudi srečamo s problemom, kako trajektorijo izračunati na podlagi podatkov, ki so vstavljeni v računalniku (generiranje trajektorije). To generiranje, izračunavanje se naj opravi sproti v realnem času, torej se morajo sproti izračunavati elementi pozicije, orientacije, hitrosti, pospeška, eventualnih sil in momentov. Seveda se to odvija iz točke v točko z neko dano vzorčno frekvenco (path update rate), ki je v realnih sistemih med 20 in 200 Hz. Ker so v računalniku vstavljenе le določene točke poti, oziroma kakšen drug način opisa poti, ki gre od točke A do B, potem mora vse vmesne točke sistem sam izračunati in s tem pridemo do problema interpolacije oziroma sestave interpolatorja.

Planiranje trajektorije lahko v blokovni obliki prikažemo v obliki, ki jo kaže slika 4.1.

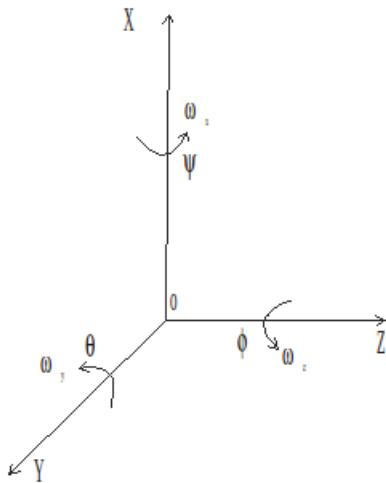


Slika 4.1: Vhodi v planiranje trajektorije.

Največkrat sta funkciji generiranje trajektorije in interpolacija združeni v enem bloku znotraj krmilnika robota.

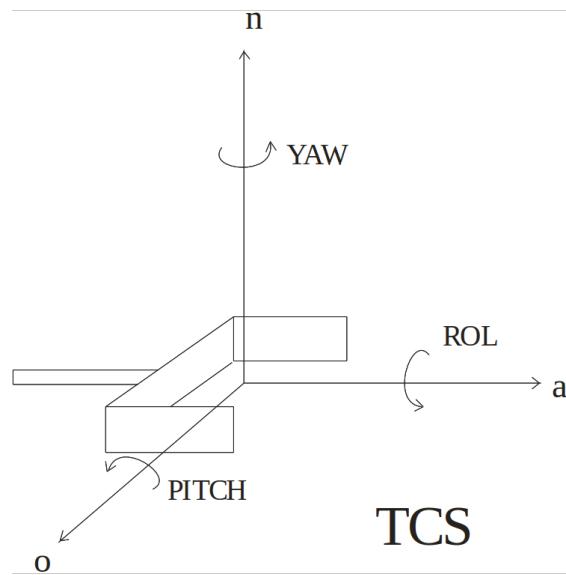
#### 4.1.1 Pomen interpolacije

Interpolacija je v osnovi nujna tako v fazi učenja robota, kot v fazi izvajanja giba. Podajanje notranjih koordinat  $q_i$ , ko robot ročno učimo ali programiramo v notranjih koordinatah je nepraktično, zamudno ter neuporabno. Zato podajamo bolj nazorno podatke na vhodu ali v osnovnem koordinatnem sistemu ali v koordinatnem sistemu orodja po naslednjih slikah. Predstavljammo si nanašanje lepila ali varjenje, kjer senzor na orodju opisuje režo vara in predpisuje trajektorijo tako da robot sledi verno poti za lepilo. Interpolator mora v fazi izvajanja giba znati generirati trajektorijo, na podlagi: 1. začetne in končne točke, 2. včasih interpolator, npr. pri brušenju in raziglanju, po pravilih umetne inteligence in informacije senzorja izbere optimalno pot izmed vseh možnih. Osnova pri tem je podatek taktilnega senzorja, na podlagi katerega se izračuna naslednja točka orientacije in pozicije, da odbrusimo iglo uspešno.

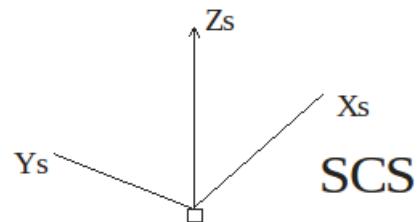


Slika 4.2: Osnovni koordinatni sistem (WCS, World Coordinate System).

- a) Torej ločimo interpolacijo v fazi učenja, ko interpolator določa samo vmesne točke, začetno, končno in specifične vmesne (via) točke, ki se vnesejo v računalnik.
- b) V fazi izvajanja interpolator določa vmesne točke med onimi, ki so vstavljeni v spomin, kar je prva možnost, ali na podlagi informacije senzorjev in umetne inteligence izračuna naslednjo točko na poti do končne točke.

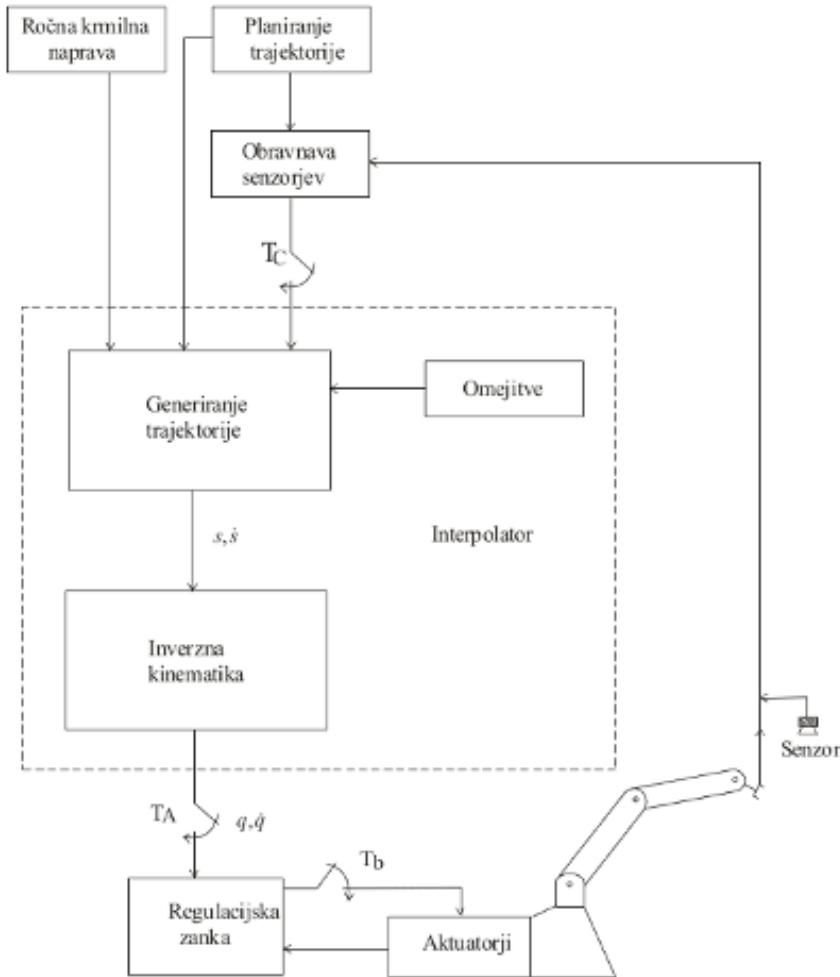


Slika 4.3: Koordinatni sistem orodja (TCS, Tool Coordinate System).



Slika 4.4: Koordinatni sistem senzorja (SCS, Sensor Coordinate System).

Na sliki 4.5 je podan osnovni prikaz interpolacije in izvedba poenostavljenega robotskega sistema.



Slika 4.5: Osnovne funkcije robotskega krmilnika.

Na sliki ločimo tri nivoje za sekvenčno obnavljanje informacije. Najvišji nivo je  $T_c$ , ki specificira ciljno točko in s tem celoten gib. Definira drugi nivo, ki pomeni posredovanje zaporedja točk iz interpolatorja (potem, ko so preko inverzne kinematike preračunane v zaporedje vrednosti, točk notranjih koordinat) kot referenčno veličino na vhode servosistemov po sklepih v vzorčni frekvenci  $T_a$ . Zadnji nivo je  $T_b$ , ki predstavlja diskretni servosistem na nivoju sklepa in njegovo vzorčenje. Velja, da so vzorčni intervali vedno krajsi na poti k nižnjim nivojem. Zaradi tega velja

$$T_b \leq T_a \leq T_t = T_c \quad (4.1)$$

Za krajše  $T_b$  bo trajektorija podana podrobnejše in doseženo manjše odstopanje. Važno je, da so ti nivoji  $T_c$ ,  $T_a$ ,  $T_b$  neodvisni. Težave pri izračunavanju

veličin bomo poskušali odpraviti tako, da bomo uporabili ločene mikroračunalnike in tako dosegli čim krajše čase in boljše sledenje.

#### 4.1.2 Izbira interpolacije

Za neko dano delovno krivuljo, trajektorijo, ki smo jo predpisali in sestavili (programer robota) glede na delovno nalogu, bo algoritom za planiranje trajektorij razdelil celotno pot na segmente in točke začenši s startno točko (pozicijo in orientacijo) ter vmesnimi točkami. Točke, v katerih se spremeni smer, so vmesne točke (via points). Vmesne točke in končno ciljno točko, shranimo v spomin po učenju in planiraju. Vse točke bo algoritom za planiranje trajektorije skladno upošteval in določil potrebne vmesne točke. Te točke z vzorčnim intervalom  $T_a$  interpolator posreduje osnim modulom.

Ciljna točka oziroma  $T_c$  se določa lahko na dva načina:

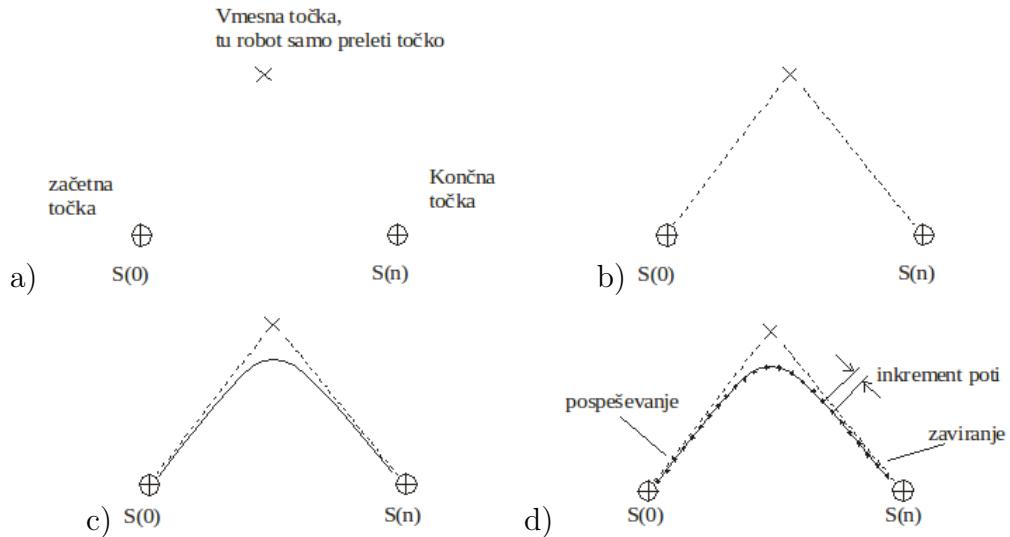
1. S končno točko trajektorije, ki je določena v programu ob učenju robota ali z vrednostjo odčitka senzorja, ki zaustavi izvajanje robota. Ta način podajanja  $T_c$ , s programom ali z vrednostjo odčitka senzorja, je najbolj razširjen način in je v rabi skoraj pri vseh konvencionalnih robotih. Za ta primer se  $T_c$  podaja enkrat ob začetku izvajanja trajektorije.
2. V adaptivnih robotih sproti glede na informacijo senzorjev program določi  $T_c$  glede na opravilo npr. raziglanje z brusom. Program traja tako dolgo, dokler delo ni opravljeno v celoti na delovnem kosu, nakar se program vrne na začetek za delo na novem kosu. Na ta način je robot na podlagi senzorne informacije optimalno voden po trajektoriji, ki se sproti izračunava in z orodji umetne inteligence optimizira.

V programu za delo robota želimo imeti čim manj točk (hitro in lahko delo). Za izvajanje pa rabimo mnogo točk. Točno predpisano trajektorijo določi generator trajektorije, vmesne točke pa interpolator. Naša naloga je, da pogledamo principe in kriterije za oba algoritma. Grafično generiranje trajektorije in interpolacijo predstavimo na sliki 4.6 za nazornejšo predstavitev.

Vmesna (via) točka – tu robot samo preleti točko.

- a) Te točke so bile podane v programu in so bile določene v fazi učenja ali programiranja naloge.
- b) Črtkasta črta določa  $n$  variant delovne trajektorije.

- c) Realna delovna trajektorija, ki upošteva pospeške in druge omejitve.
  - d) Interpolacija, ki upošteva pospeševanje, zaviranje in inkrement poti, ki zadošča za želeno točnost dela.



Slika 4.6:

Veljajo naslednje preproste relacije

$$\Delta S(0) = 0 \quad (4.2)$$

$$\Delta S(i) = S(i)' \cdot T_a \quad (4.3)$$

$$S(i) = S(i-1) + \Delta S(i) \quad a_1 = \text{konst} = S'' \quad (4.4)$$

$$S(i) = S(i-1) + \frac{a_1 T_a^2}{2} \quad (4.5)$$

Gre poudariti, da začetna točka, vsaka vmesna in končna točka na trajektoriji podajajo pozicijo in orientacijo orodja oziroma vrha manipulatorja. Zaradi tega moramo pri izvajanju interpolacije za vmesne točke paziti, da upoštevamo omejitve tako v smeri  $x, y, z, \Theta, \Phi, \Psi$ , kakor tudi omejitve v hitrosti, pospešku in to preko inverzne kinematike za vsako os. Torej, ko interpoliramo, bodo pravila podprograma in algoritmi izvajanja podani za vsako os, tako za vsako koordinato zunanjega kot notranjega prostora. Zaradi tega bomo podali probleme interpolacije za eno os npr.  $x$ -os in sestavili algoritem, ki ga bo možno zapored uporabljati smiselno za poljubno koordinatno os.

Ločimo dve vrsti interpoliranja:

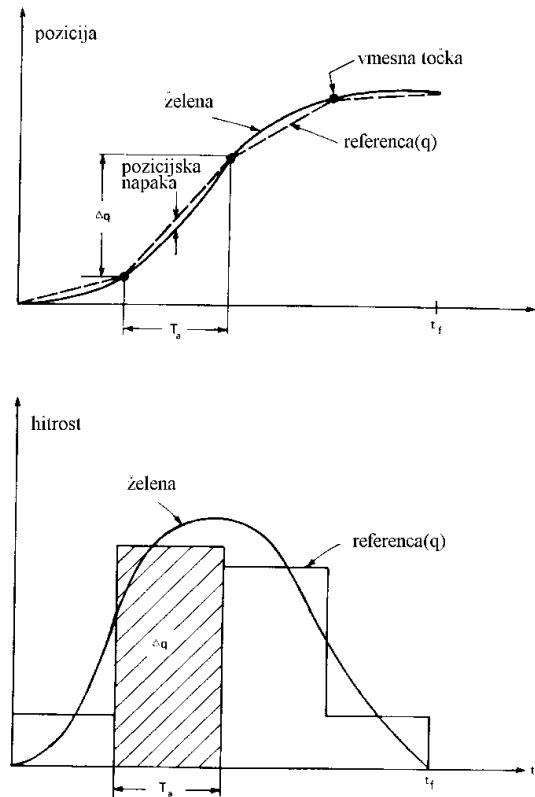
1. ABSOLUTNE INTERPOLATORJE
2. INKREMENTALNE INTERPOLATORJE

Absolutni interpolatorji hranijo podatke o poti (referenca je pot), inkrementalni podatki o hitrosti (referenca je hitrost).

#### 4.1.3 Absolutni interpolator

Vhodna veličina v interpolator je izračunana trajektorija s pripadajočo krivuljo hitrosti. Te veličine se v interpolatorju preračunajo v ustrezne vrednosti za notranje koordinate  $q(t)$  in  $q'(t)$  za vsako os, prostostno stopnjo posebej. Te veličine  $q$  in  $q'$  so referenčne veličine za regulatorje osi.

Absolutni interpolator razdeli pot  $q$  na vzorčne intervale  $T_a$  in določi tako vse interpolacijske točke  $q(i)$  (glej sliko 4.7).



Slika 4.7: Absolutni interpolator.

Vrednosti  $q(i)$  so referenca za pozicijo in med temi točkami je trajektorija poti linearна. Zaradi tega pride do največje napake na sredini vsakega intervala. Hitrost je izbrana glede na ta linearni del in je ves čas med  $q_i$  in  $q_{i+1}$  konstantna. V vsakem vozlu dosežemo dokaj dobro ujemanje med referenčno in dejansko pozicijo. Čas je določen s

$$t = (i + 1) \cdot T_a \quad (4.6)$$

Iz slike napišemo

$$\Delta q(i) = q(i) - q(i - 1) \quad (4.7)$$

in iz tega srednjo absolutno hitrost

$$\dot{q}(i) = \frac{\Delta q(i)}{T_a}. \quad (4.8)$$

Razumljivo bo napaka med dejansko in referenčno vrednostjo naraščala z večjim  $T_a$  in višjo hitrostjo. Prisotna bo še dodatna napaka zaradi diferenciranja. Važno je uvideti, da izračun referenčnih pozicij ni odvisen od prejšnje pozicije, ker so točke  $q_i$  dane, torej se napake ne sumirajo. Pri absolutnih interpolatorjih imamo lahko težavo pri preračunavanju danih referenčnih točk na krivulji poti v zunanjih koordinatah preko inverzne kinematike v vrednosti  $q_i$  za posamezne osi. Nastane lahko napaka v singularnih točkah (v singularni točki sta tako  $+\varphi$  kot  $-\varphi$  rešitev in seveda samo ena rešitev je prava). To moramo upoštevati v algoritmu, tako da zna izbrati pravilno rešitev.

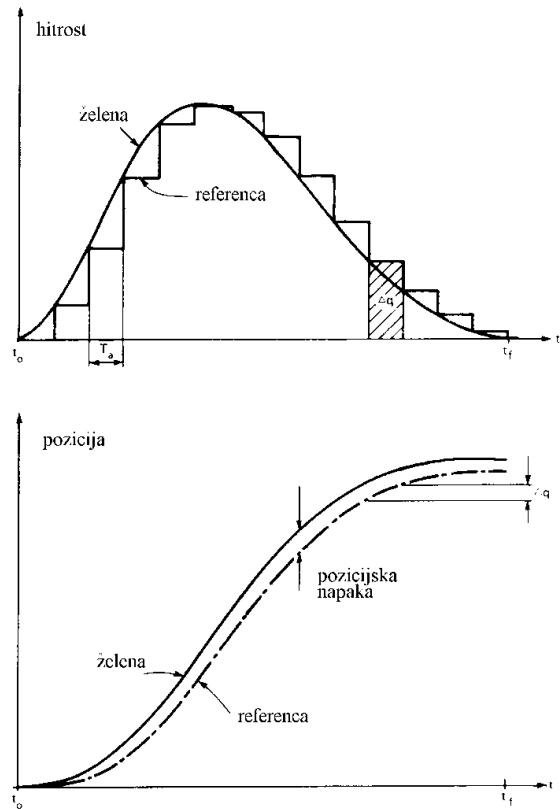
#### 4.1.4 Inkrementalni interpolator

Iz znanih vrednosti  $s(t)$  in  $\dot{s}(t)$  izračunamo  $\dot{q}(i)$  in krivuljo za hitrost posamezne osi aproksimiramo v interpolatorju s stopničasto krivuljo na sliki 4.8), ki predstavlja referenčno veličino za osne regulatorje.

Samo na začetku in koncu intervala  $T_a$  sta prava in referenčna hitrost enaki. Hitrost na intervalu  $T_a$  je konstantna. Da bi dobili pozicijo, moramo integrirati hitrosti  $\dot{q}(i)$  preko intervala  $T_a$ , torej seštevamo pravokotnike pod krivuljo na sliki in tako dobimo referenčno pozicijo, ki se razlikuje od želene. Enoti, ki jih integriramo, so  $\Delta q$

$$\Delta q(i) = \dot{q}(i) \cdot T_a \quad (4.9)$$

$$q(n) = \sum_{i=1}^n \Delta q(i) \quad (4.10)$$



Slika 4.8: Inkrementalni interpolator.

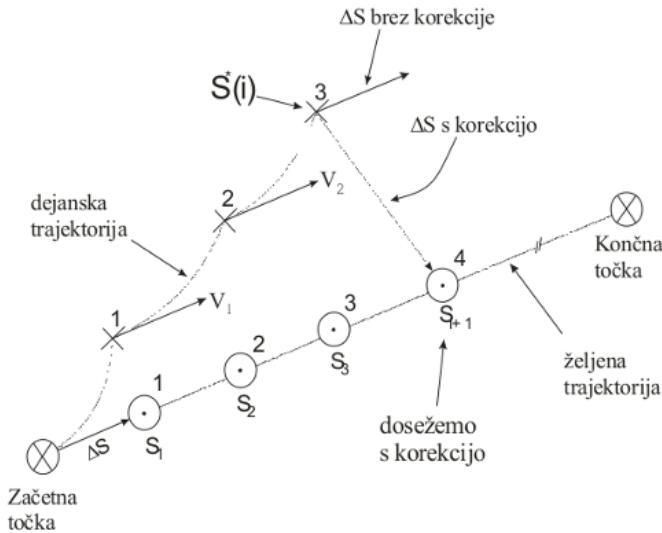
Vsaka vrednost  $q(i)$  je referenčna vrednost za regulator osi in se razlikuje od želene referenčne vrednosti, ki je dana s časovnim integralom preko časa  $t$

$$q(t) = \int_0^t \dot{q}(t) dt \quad (4.11)$$

Vrednost  $q(t)$  je na sliki podana s polno črto. Storjena napaka  $q(t) - q(n)$  je nastala, ker smo hitrost preko intervala imeli konstantno. Vse napake po oseh se seštevajo in dobimo dejansko napako v prostorski trajektoriji. Ta napaka se akumulira in veča z vsakim interpolacijskim intervalom in bo zaradi tega končna pozicija hudo narobe. Sam inkrementalni (hitrostna interpolacija) interpolator te napake ne more preprečiti.

Ker se pozicijske napake seštevajo, bodo tem večje, čim večje bodo napake v integraciji, to je,  $T_a$  intervali. Toda tudi z zelo zmanjšanimi intervali  $T_a$  bo napaka ostala, bo pa dokaj manjša. Grafično moremo razmere ponazoriti

zsliko 4.9.

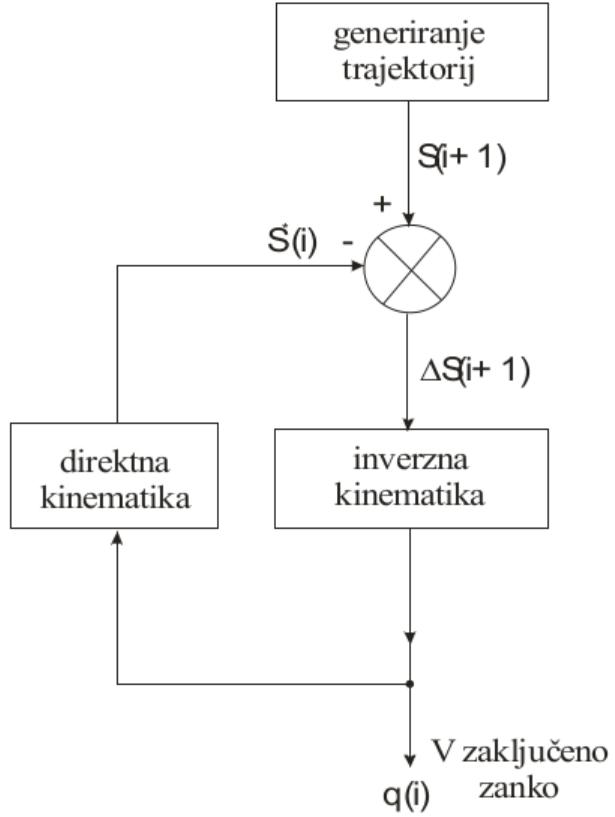


Slika 4.9: Razmere pri inkrementalnem interpolatorju.

Pri inkrementalnem interpolatorju je premica poti interpolirana in predstavlja želeno trajektorijo. Dejanska trajektorija s časom vedno bolj odstopa, ker smo uporabili inkrementalni interpolator (hitrostna interpolacija), saj se napake akumulirajo. To napako lahko odpravimo na nivoju generiranja trajektorije, če uporabimo korekcijo preko direktne kinematike (slika 4.10).

$$\Delta S(i+1) = S(i+1) - S(i) \quad (4.12)$$

Torej reguliramo preko poti in hitrosti, obe točki shranimo v pomnilniku.



Slika 4.10: Korekcija napake nastale pri inkrementalni interpolaciji.

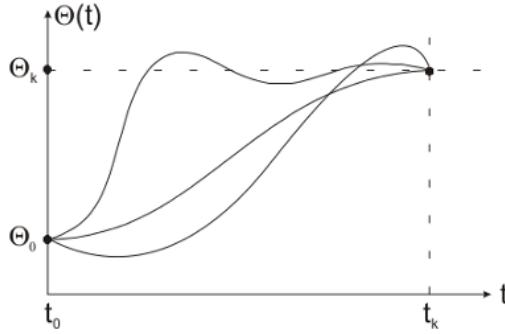
## 4.2 APROKSIMACIJA Z OSNOVNIMI FUNKCIJAMI

V robotiki moramo pogosto predmet prestaviti iz točke  $T_1$  v  $T_2$  v predpisanim času. Preko inverzne kinematike določimo ustrezno začetno in končno vrednost za ta premik in to za vsako sklepno spremenljivko. Problem grafično prikažemo za naslednjo sliko, kjer tudi vidimo, da je možnih veliko rešitev, poti. Za gladko trajektorijo poti mora biti hitrost zvezna funkcija. Ker v začetni in končni točki predmet miruje, imamo naslednje robne pogoje

$$\begin{aligned} \Theta(0) &= \Theta_0, & \Theta(t_k) &= \Theta_k \\ \dot{\Theta}(0) &= 0, & \dot{\Theta}(t_k) &= 0. \end{aligned} \quad (4.13)$$

Iz matematike vemo, da štirim robnim pogojem lahko zadostimo s polinomom, ki ima štiri konstante, torej s polinomom tretjega reda

$$\Theta(t) = a_3t^3 + a_2t^2 + a_1t + a_0. \quad (4.14)$$



Slika 4.11: Nekatere možne poti med dvema točkama.

Določiti moramo konstante hitrosti oziroma pospeška.

$$\dot{\Theta}(t) = 3a_3t^2 + 2a_2t + a_1 \quad (4.15)$$

$$\ddot{\Theta}(t) = 6a_3t + 2a_2 \quad (4.16)$$

Konstante določimo tako, da vstavimo robne pogoje in dobimo štiri enačbe.

$$\Theta_0 = a_0 \quad (4.17)$$

$$\Theta_k = a_3t^3 + a_2t^2 + a_1t + a_0 \quad (4.18)$$

$$\dot{\Theta}(0) = 0 = a_1 \quad (4.19)$$

$$\dot{\Theta}(t_k) = 0 = 3a_3t_k^2 + 2a_2t_k + a_1 \quad (4.20)$$

$$\begin{aligned} \Theta_k &= a_3t_k^3 + a_2t_k^2 + a_0 \\ 0 &= 3a_3t_k^2 + 2a_2t_k \quad / \cdot t_k/3 \\ \Theta_k &= a_2t_k^2 - 2/3a_2t_k^2 + \Theta_0 \\ \Theta_k &= a_2t_k^2/3 + \Theta_0 \\ a_2 &= (3/t_k^2)(\Theta_k - \Theta_0) \end{aligned} \quad (4.21)$$

$$\begin{aligned} 3a_3t_k^2 + (2 \cdot 3/t_k)(\Theta_k - \Theta_0) &= 0 \quad / \cdot t_k^{-2} \\ 3a_3 + (6/t_k^3)(\Theta_k - \Theta_0) &= 0 \\ a_3 &= -2(\Theta_k - \Theta_0)/t_k^3 \end{aligned} \quad (4.22)$$

Če te izraze vstavimo v izraz za  $\Theta(t)$ , dobimo

$$\Theta(t) = (-2/t_k^3)(\Theta_k - \Theta_0)t^3 + (3/t_k^2)(\Theta_k - \Theta_0)t^2 + \Theta_0 \quad (4.23)$$

$$\dot{\Theta}(t) = (-6/t_k^3)(\Theta_k - \Theta_0)t^2 + (6/t_k^2)(\Theta_k - \Theta_0)t \quad (4.24)$$

S tem dvojico enačbama lahko za vsak čas  $T$  izračunamo pot in hitrost. Če interval  $[0, t_k]$  razdelimo na enake podintervale, npr.  $i$ , bomo tako izvedli

interpolacijo.

Prikazani primer je veljal, ko je hitrost v začetni in končni točki bila enaka nič (mirovanje). Takšen primer je v robotiki dokaj pogost.

Primer:

$$\begin{aligned}\Theta(0) &= 15, \quad \Theta(3) = 75 \\ \dot{\Theta}(0) &= 0, \quad \dot{\Theta}(3) = 0\end{aligned}$$

$$a_0 = 15 \quad a_1 = 0$$

$$a_2 = (3/t_k^2)(\Theta_k - \Theta_0) = 3/3^2(75 - 15) = 1/3 \cdot 60 = 20$$

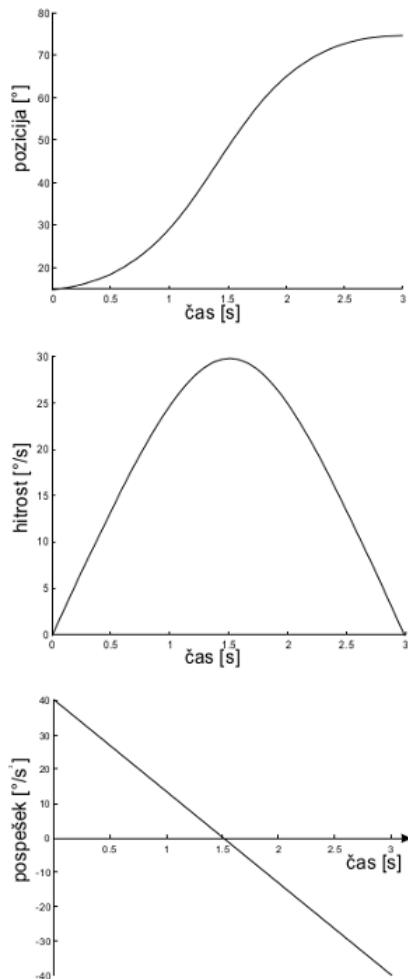
$$a_3 = (-2/t_k^3)(\Theta_k - \Theta_0) = -2/3^3 \cdot 60 = -120/27 = -4.44$$

$$\Theta(t) = -4.44t^3 + 20t^2 + 15$$

$$\dot{\Theta}(t) = -13.32t^2 + 40t$$

$$\ddot{\Theta}(t) = -26.64t + 40$$

Tako lahko, narišemo točke za  $\Theta(t)$ ,  $\dot{\Theta}(t)$ ,  $\ddot{\Theta}(t)$ , ki jih povežemo v krivuljo. Za npr. vzorčno frekvenco 20 Hz dobimo krivulje na sliki 4.12. Pomni, da je hitrost za kubično pot (polinom) parabola in pospešek premica. V robotiki pogosto vzorčimo v območju med 40 in 200 Hz.



Slika 4.12: Pozicija, hitrost in pospešek za prikazani primer.

#### 4.2.1 Polinomaska aproksimacija z kubičnim polinomom za interpolacijo poti z vmesno točko

Do sedaj smo spoznali trajektorije, kjer je bila v začetni in končni točki hitrost enaka nič. Torej sta bila podana samo razdalja in čas, v katerem moramo razdaljo preleteti. V splošnem želimo takšne krivulje poti, kjer na poti lahko predpišemo vmesne točke. Če se v vsaki vmesni točki manipulator ustavi, lahko zopet uporabimo kubični polinom za izračun zveznosti generiranja trajektorije. Običajno želimo preleteti vmesne točke, ne da bi se ustavili. Zaradi tega moramo najti način, kako bi z novo nastalimi robnimi pogoji priredili že izdelano generiranje trajektorije s kubičnim polinomom.

Vsaka vmesna točka je predpisana z začetno pozicijo in orientacijo orodja. Torej lahko preko inverzne kinematike določimo želene sklepne, notranje koordinate  $q$ . Ko imamo za vse vmesne točke določene  $q$  notranje koordinate, se lotimo za vsak sklep posebej povezovati te vmesne točke tako, da ne pride do trzanja, toda ohraniti moramo zvezno hitrost. Če imamo želeno hitrost v vsaki vmesni točki, potem določimo po kubičnem polinomu trajektorije. Ko to izračunamo vstavljamo v robne pogoje ustrezne hitrosti na začetku in koncu poti. V naših enačbah se torej spreminjajo robni pogoji v obeh vmesnih točkah.

$$\Theta(0) = \Theta_0 \quad \Theta(t_k) = \Theta_k \quad (4.25)$$

$$\dot{\Theta}(0) = \dot{\Theta}_0 \quad \dot{\Theta}(t_k) = \dot{\Theta}_k \quad (4.26)$$

Enačbe za opis sistema so

$$\Theta_0 = a_0 \quad (4.27)$$

$$\Theta_k = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (4.28)$$

$$\dot{\Theta}_0 = a_1 \quad (4.29)$$

$$\dot{\Theta}_k = 3a_3 t^2 + 2a_2 t + a_1 \quad (4.30)$$

Rešitev sistema enačb je

$$a_0 = \Theta_0 \quad (4.31)$$

$$a_1 = \dot{\Theta}_0 \quad (4.32)$$

$$a_2 = (3/t_k^2)(\Theta_k - \Theta_0) - (2/t_k)\dot{\Theta}_0 - (1/t_k)\dot{\Theta}_k \quad (4.33)$$

$$a_3 = (-2/t_k^3)(\Theta_k - \Theta_0) + (1/t_k^2)(\dot{\Theta}_k + \dot{\Theta}_0) \quad (4.34)$$

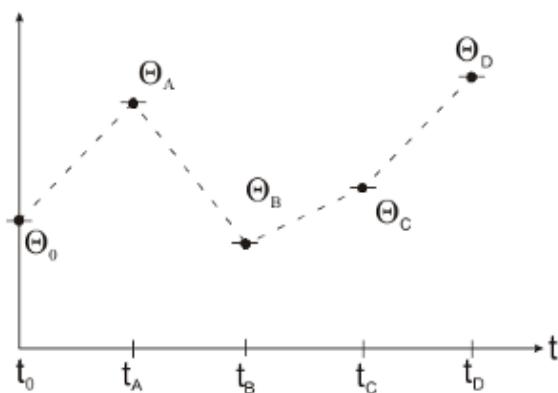
Tako lahko zvezno povežemo dve točki, ki imata poleg lege predpisane tudi hitrosti od začetka v prvi točki do konca v drugi točki. Če imamo za vsak sklep podane hitrosti v vmesni točki, potem iz sistema enačb izračunamo konstante in lahko sestavimo polinom za interpolacijo.

Hitrosti v vmesnih točkah so lahko podane na več načinov:

1. Programer predpiše v kartezijevem prostoru za vsako vmesno točko želeno linearno in kotno hitrost.
2. Za dve vmesni točki robotski regulator sam določi primerne hitrosti glede na specificirane hitrosti na posameznih segmentih poti.

3. Robotski sistem vodenja avtomatično izbere hitrost v vmesnih točkah glede na predpisani čas celotne trajektorije, tako da je zagotovljena zveznost pospeška v vsaki vmesni točki. V tem primeru potrebujemo novo metodo, ki bo v robnih pogojih določala ne samo hitrost temveč tudi pospeške.

Za drugo točko je seveda potrebno najti neko hevristično preprosto metodo, ki lahko hitro določi hitrost v vsaki vmesni točki na podlagi hitrosti vstopnega in izstopnega segmenta. Preprosto pravilo pravi: če hitrost v vmesni točki spremeni predznak, potem izberemo hitrost v vmesni točki enako nič, v nasprotnem pa vzamemo srednjo vrednost (4.13). Hitrost  $v(t)$  v vmesni točki podamo s strmino premice, ki je dobljena po predlaganem pravilu.



Slika 4.13: Prikaz končnih in vmesnih točk.

#### 4.2.2 Polinomi višjega reda

Če želimo na nekem segmentu poti predpisati na začetku in na koncu segmenta vse tri parametre vključujuč pozicijo, hitrost in pospešek, tedaj moremo zaradi šestih robnih podatkov uporabiti polinom 5. reda

$$\Theta(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0. \quad (4.35)$$

Tu so robni pogoji dani z (0, začetek in  $k$ , konec segmenta)

$$\Theta_0 = a_0 \quad (4.36)$$

$$\Theta_k = a_5 t_k^5 + a_4 t_k^4 + a_3 t_k^3 + a_2 t_k^2 + a_1 t_k + a_0 \quad (4.37)$$

$$\dot{\Theta}_0 = a_1 \quad (4.38)$$

$$\dot{\Theta}_k = 5a_5 t_k^4 + 4a_4 t_k^3 + 3a_3 t_k^2 + 2a_2 t_k + a_1 \quad (4.39)$$

$$\ddot{\Theta}_0 = 2a_2 \quad (4.40)$$

$$\ddot{\Theta}_k = 20a_5 t_k^3 + 12a_4 t_k^2 + 6a_3 t_k + 2a_2 \quad (4.41)$$

Ti robni pogoji predpisujejo sistem šestih polinomskih enačb, ki imajo naslednjo rešitev

$$a_0 = \Theta_0 \quad (4.42)$$

$$a_1 = \dot{\Theta}_0 \quad (4.43)$$

$$a_2 = \ddot{\Theta}_0 / 2 \quad (4.44)$$

$$a_3 = (20\Theta_k - 20\Theta_0 - (8\dot{\Theta}_k + 12\dot{\Theta}_0)t_k - (3\ddot{\Theta}_0 - \ddot{\Theta}_k)t_k^2) / (2t_k^3) \quad (4.45)$$

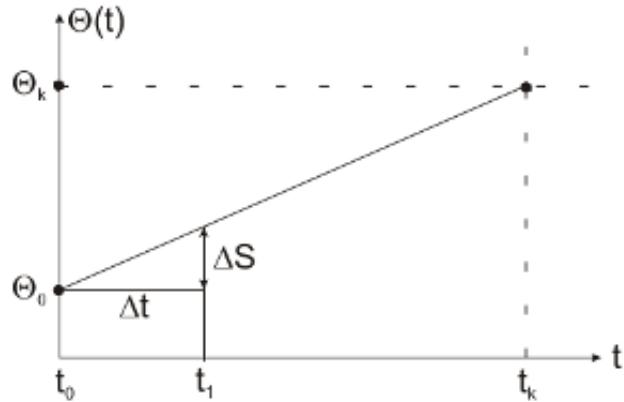
$$a_4 = (30\Theta_0 - 30\Theta_k + (14\dot{\Theta}_k + 16\dot{\Theta}_0)t_k + (3\ddot{\Theta}_0 - 2\ddot{\Theta}_k)t_k^2) / (2t_k^4) \quad (4.46)$$

$$a_5 = (12\Theta_k - 12\Theta_0 - (6\dot{\Theta}_k + 6\dot{\Theta}_0)t_k - (\ddot{\Theta}_0 - \ddot{\Theta}_k t_k^2)) / (2t_k^5) \quad (4.47)$$

Pomni, da so možni za zvezno povezavo šestih robnih pogojev še drugi algoritmi.

### 4.2.3 Linearni segmenti s paraboličnimi nastavki

Zelo priljubljena zaradi geometrijsko-matematične nazornosti in enostavnosti (narava ne pozna idealiziranih razmer iz geometrije in matematike) je linearna premica kot segment poti. Linearna interpolacija med dvema segmentoma poti je prikazana na sliki 4.14. Prikazan je tak segment poti, ki seveda v realnosti ne bo vodil do linearnega gibanja vrha manipulatorja.



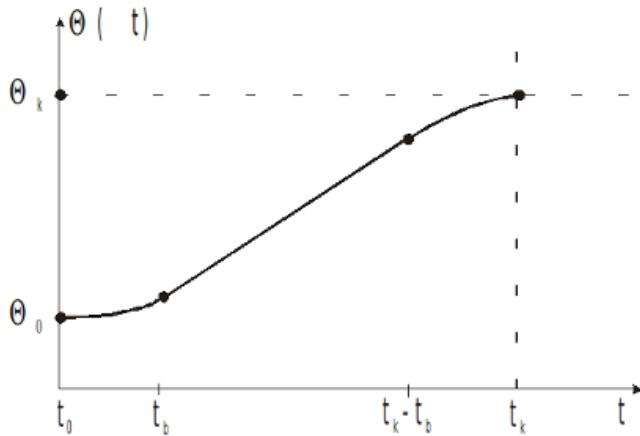
Slika 4.14: Linearni segment poti.

$\dot{\Theta}(0) = k$ ,  $\dot{\Theta}_k(t_k) = k'$ , v obeh primerih je pospešek  $\ddot{\Theta}$  nezvezen.

Take razmere vodijo do nezveznosti v hitrosti na začetku in koncu poti.

$$v = (\Theta_1 - \Theta_0)/\Delta t_1, \\ \text{za } t_0 = 0, v = \lim(\Theta_1 - \Theta_0)/\Delta t_1 \rightarrow \infty, \lim v \rightarrow \infty. \quad (4.48)$$

Da bi lahko dosegli zvezno hitrost in pot bomo začeli s segmentom v obliki parabole, ki preide v linearno pot in na koncu iz linearnega dela nazaj na parabolo v končno točko (slika 4.15).



Slika 4.15: Linearni segment z dvema paraboličnima nastavkoma.

Če sta  $\dot{\Theta}$  in  $\ddot{\Theta}$  hitrost ter pospešek, potem pri  $t$  med  $t_o$  do  $t_b$  velja

$$\Theta = \Theta_0 + \frac{1}{2}\ddot{\Theta}t^2. \quad (4.49)$$

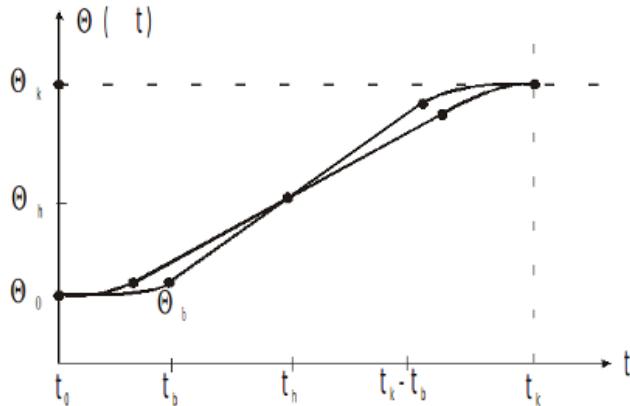
Od  $t_b$  do  $t_k - t_b$  je

$$\Theta = \dot{\Theta}t \quad (4.50)$$

in pri  $t$  od  $t_k - t_b$  do  $t_k$  je

$$\Theta = \Theta_{t_k-t_b} - \frac{1}{2}\ddot{\Theta}t^2. \quad (4.51)$$

V teh prehodnih delih je uporabljeno konstantno pospeševanje. Tako dobimo s časom zvezne hitrost in kot. Oba parabolična dela in del linearne poti tvorijo en segment poti.



Slika 4.16: Dve možni poti z ustreznimi oznakami.

Če upoštevamo oznake na sliki 4.16, potem bosta hitrost in pot ob času  $t_b$  zapisani kot

$$\begin{aligned} \ddot{\Theta} \cdot t_b &= (\Theta_h - \Theta_b)/(t_h - t_b), & t_k &= 2t_h, \quad t_h = t_k/2 \\ \Theta_b &= \Theta_0 + \frac{1}{2}\ddot{\Theta} \cdot t_b^2. \end{aligned}$$

Prvo enačbo preoblikujemo in združimo z drugo

$$\begin{aligned} \ddot{\Theta} \cdot t_b \cdot (t_h - t_b) &= (\Theta_h - \Theta_b), \\ \Theta_b &= \Theta_h - \ddot{\Theta} \cdot t_b \cdot (t_h - t_b) = \Theta_0 + \frac{1}{2}\ddot{\Theta} \cdot t_b^2. \end{aligned}$$

Zapišemo skupaj, vstavimo še  $t_k/2$

$$\begin{aligned}\frac{1}{2}\ddot{\Theta} \cdot t_b^2 + \ddot{\Theta} \cdot t_b \cdot (t_h - t_b) - \Theta_h + \Theta_0 &= 0, \\ \frac{1}{2}\ddot{\Theta} \cdot t_b^2 + \ddot{\Theta} \cdot t_b \cdot \frac{t_k}{2} - \ddot{\Theta} \cdot t_b^2 - \Theta_h + \Theta_0 &= 0,\end{aligned}$$

pomnožimo z -2

$$\begin{aligned}-\frac{1}{2}\ddot{\Theta} \cdot t_b^2 + \frac{1}{2}\ddot{\Theta} \cdot t_k \cdot t_b - \Theta_h + \Theta_0 &= 0, \\ \ddot{\Theta} \cdot t_b^2 - \ddot{\Theta} \cdot t_k \cdot t_b + 2(\Theta_h - \Theta_0) &= 0,\end{aligned}$$

ter upoštevamo še

$$2(\Theta_h - \Theta_0) = \Theta_k - \Theta_0.$$

Končno dobimo kvadratično enačbo, preko katere lahko izračunamo čase pospeševanja in zaviranja

$$\ddot{\Theta} \cdot t_b^2 - \ddot{\Theta} \cdot t_k \cdot t_b + (\Theta_k - \Theta_0) = 0. \quad (4.52)$$

Običajno bomo izbrali v obeh, začetnem in končnem nastavku enako trajanje in isti konstantni pospešek. Vsekakor ima tako sestavljena naloga več rešitev glede na izbrani čas in pospešek. Toda rešitev je vedno simetrična glede na polovico poti (slika 4.16).

Še enkrat ponovno bistveno

$$t_b \leq t_h, \quad (4.53)$$

velja pa tudi  $2t_h = t_k$ , in

$$t_b \leq \frac{t_k}{2}. \quad (4.54)$$

Hitrost na koncu prvega in na začetku drugega nastavka se mora ujeti s hitrostjo linearne poti.

$$\dot{\Theta} \cdot t_b = (\Theta_h - \Theta_b)/(t_h - t_b). \quad (4.55)$$

$\ddot{\Theta}$  je pospešek v prvem in zadnjem nastavku.

Vrednost  $\Theta_b$  je dana z

$$\Theta_b = \Theta_0 + \frac{1}{2}\ddot{\Theta} \cdot t_b^2. \quad (4.56)$$

S pomočjo zadnjih obeh enačb ter za čas  $t = 2t_h$  dobimo

$$\ddot{\Theta} \cdot t_b^2 - \ddot{\Theta} \cdot t_k \cdot t_b + (\Theta_k - \Theta_0) = 0, \quad (4.57)$$

$t_k$  predstavlja skupen potreben čas za gibanje. Če so dani  $\Theta_k$ ,  $\Theta_0$  in izbrani  $t_k$ , potem lahko vedno najdemo ustrezen  $\ddot{\Theta}$  in  $t_b$  čas, ki ustreza pogojem zveznosti. Običajno izbiramo pospešek  $\ddot{\Theta}$  in iščemo čas  $t_b$ . Seveda mora biti pospešek primerno velik, drugače en. (4.57) nima rešitve. V skladu s to enačbo lahko  $t_b$  izračunamo

$$t_b = \frac{t_k}{2} - \frac{\sqrt{\ddot{\Theta}^2 \cdot t_k^2 - 4\ddot{\Theta}(\Theta_k - \Theta_0)}}{2\ddot{\Theta}}. \quad (4.58)$$

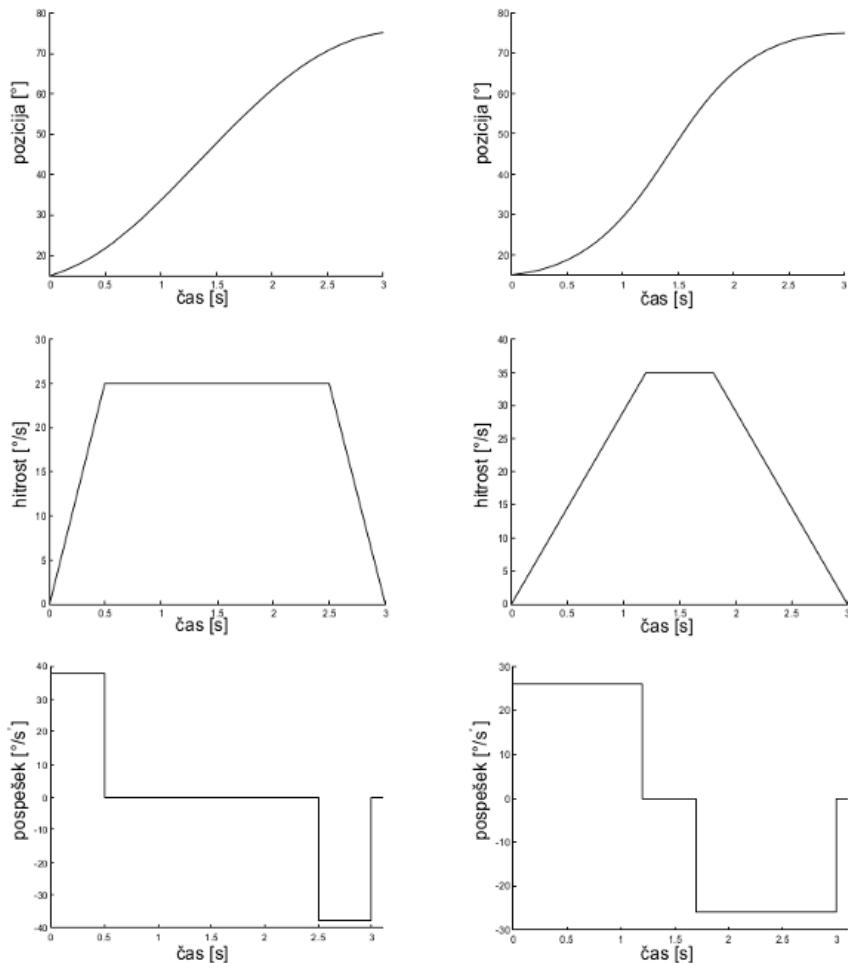
Rešitev korena bo obstojala pri pozitivnem argumentu, torej mora biti pospešek večji ali enak po vrednosti

$$\ddot{\Theta} \geq \frac{4(\Theta_k - \Theta_0)}{t_k^2}. \quad (4.59)$$

Ko nastopi enakost po tej enačbi, tedaj v poti ni več linearne usmerjenega dela in sta se oba parabolična nastavka razširila do polovice poti. Z večanjem pospeška postaja linearni del večji in nastavka krajša, in za  $d^2\Theta/dt^2 \rightarrow \infty$  bomo dobili nazaj čisto linearne pot, premico.

Primer:

$\Theta_0 = 15$ ,  $\Theta_k = 75$ ,  $t_k = 3s$ . Na sliki 4.17 so izračunani pot, hitrost in pospešek levo za uporabo velikega pospeška in desno za bistveno manjši pospešek.



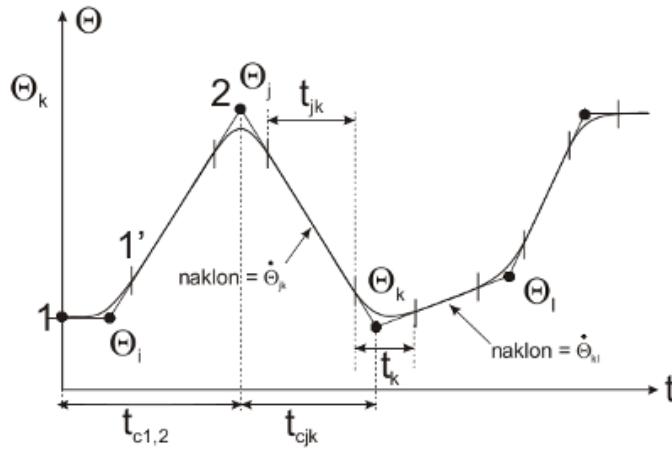
Slika 4.17: Pot, hitrost in pospešek za primer z  $\Theta_0 = 15$ ,  $\Theta_k = 75$ ,  $t_k = 3s$ .

#### 4.2.4 Vmesne točke in premične povezave

V nadaljevanju so opisane trajektorije poljubne oblike, ki jih sestavljajo vmesne točke in premične linearne povezave.

Imamo trajektorijo, na kateri je med začetno in končno točko poljubno vmesnih točk. Točke so povezane s premicami ( $v = \text{konst.}$ ). Npr., slika 4.18 podaja trajektorije za sklep  $\Theta$ . Točke so povezane z premicami, kjer smo že upoštevali na začetku in koncu premičnega segmeta parabolični prehod.

Za reševanje bomo uporabili naslednje označevanje: tri zaporedne točke  $j$ ,  $k$  in  $l$  povezujejo premice s paraboličnimi prehodi, časi preostalih linearnih delov, označimo med  $j$  in  $k$  za  $t_{jk}$  in celotni čas med  $j$  in  $k$  za  $t_{cjk}$  ter čas



Slika 4.18: Vmesne točke in premične linearne povezave.

paraboličnega prehoda npr. v  $k$  točki  $t_k$ .

Hitrost na linearinem delu med  $j$  in  $k$  je  $\dot{\theta}_{jk}$  in pospešek npr. v paraboličnem delu točke  $j$  je  $\ddot{\theta}_j$ .

Pri reševanju prehodov imamo neskončno možnosti, pač glede na iskani možni pospešek v bližini vsake točke vmesne točke. Če imamo podane za vse vmesne točke  $\Theta_k$ , trajanje  $t_{cjk}$  in velikost pospeška, ki ga smemo uporabiti v vsaki točki  $|\ddot{\theta}_k|$ , lahko izračunamo potrebeni čas za paraboličen prehod  $t_k$ . Za točke na notranjosti trajektorije velja (niso mejne, začetna ali končna)

$$\Theta_k - \Theta_j = \dot{\theta}_{jk} \cdot t_{cjk} \rightarrow \dot{\theta}_{jk} = \frac{\Theta_k - \Theta_j}{t_{cjk}} \quad (4.60)$$

$$\ddot{\theta}_k = SGN(\dot{\theta}_{kl} - \dot{\theta}_{jk}) \cdot |\ddot{\theta}| \quad (4.61)$$

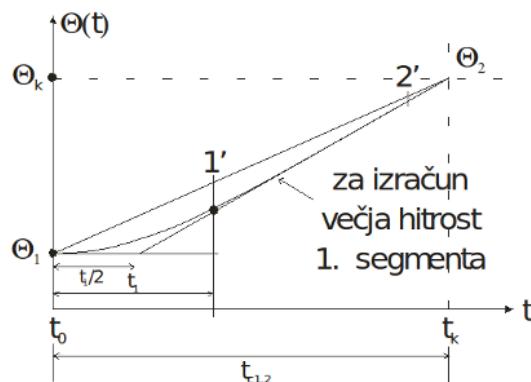
$$\dot{\theta}_{kl} - \dot{\theta}_{jk} = \ddot{\theta}_k \cdot t_k \rightarrow t_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k} \quad \leftarrow \ddot{\theta}_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{t_k} \quad (4.62)$$

$$t_{jk} = t_{cjk} - \frac{1}{2}t_j - \frac{1}{2}t_k. \quad (4.63)$$

Za prvi in zadnji segment moramo drugače računati prehode, ker moramo celoten čas prehoda všteti v prvi segmet.

**Za prvi segment** moramo najti  $t_1$  na podlagi hitrosti linearnega dela prvega segmenta. Po izteku časa  $t_1$  morata biti hitrosti prehodnega dela in linearnega segmenta enaki in pospešek predzadnjega dela nič.

Ker pospešujemo od nič, bo v prvem delu  $1/2t_1$  hitrost počasi naraščala, toda opravljena pot bo zanemarljiva, zaradi tega ta del ne upoštevamo v poti. Zaradi storjene napake bomo le to kompenzirali tako, da bomo hitrost linearnega dela prvega segmenta minimalno povečali za doprinos, ki ga prinese računanje s krajšim časom torej  $(t_{c1,2} - \frac{1}{2}t_1)$ . Glej naslednjo sliko 4.19.



Slika 4.19: Prvi segment poti.

Hitrost po izteku časa  $t_1$  mora biti za linearni del enaka oni ob koncu predhodnega dela, torej velja sledeča enačba.

$$\frac{\Theta_2 - \Theta_1}{(t_{c1,2} - \frac{1}{2}t_1)} = \ddot{\Theta}_1 \cdot t_1 \quad (4.64)$$

Le  $\frac{1}{2}t_1$  zato, ker se prvo polovico  $t_1$  pozicija spremeni le malo.

Iz te enačbe dobimo  $t_1$  ob pogoju za  $\ddot{\Theta}_1$

$$\ddot{\Theta}_1 = sign(\dot{\Theta}_2 - \dot{\Theta}_1) \cdot |\ddot{\Theta}_1| \quad (4.65)$$

Iz en. (4.64) izračunamo  $t_1$

$$(\Theta_2 - \Theta_1) = \ddot{\Theta}_1 \cdot t_1 \left( t_{c1,2} - \frac{1}{2}t_1 \right) \quad (4.66)$$

$$- \frac{1}{2}\ddot{\Theta}_1 \cdot t_1^2 + \ddot{\Theta}_1 \cdot t_1 \cdot t_{c1,2} - (\Theta_2 - \Theta_1) = 0 \quad (4.67)$$

$$t_1 = \frac{-\ddot{\Theta}_1 \cdot t_{c1,2} \pm \sqrt{\ddot{\Theta}_1^2 \cdot t_{c1,2}^2 - 4(\frac{1}{2}\ddot{\Theta}_1)(\Theta_2 - \Theta_1)}}{\ddot{\Theta}_1} \quad (4.68)$$

$$t_1 = t_{c1,2} - \sqrt{t_{c1,2}^2 - \frac{2(\Theta_2 - \Theta_1)}{\ddot{\Theta}_1}} \quad (4.69)$$

$$\dot{\Theta}_{12} = \frac{\Theta_2 - \Theta_1}{t_{c1,2} - \frac{1}{2}t_1} \quad (4.70)$$

$$t_{1,2} = t_{c1,2} - t_1 - \frac{1}{2}t_2. \quad (4.71)$$

Podobno velja tudi **za zadnji segment**, ki povezuje točko  $n-1$  z zadnjim točkom  $n$ , kjer izračunamo vse veličine (kot za 1. točko do 2. točke)

$$\frac{\Theta_{n-1} - \Theta_n}{t_{c(n-1)n} - \frac{1}{2}t_n} = \ddot{\Theta}_n \cdot t_n \quad (4.72)$$

Pri zadnjem segmentu mora veljati

$$\ddot{\Theta}_n = sign(\dot{\Theta}_{n-1} - \dot{\Theta}_n) \cdot |\ddot{\Theta}_n| \quad (4.73)$$

$$t_n = t_{c(n-1)n} - \sqrt{t_{c(n-1)n}^2 + \frac{2(\Theta_n - \Theta_{n-1})}{\ddot{\Theta}_n}} \quad (4.74)$$

$$\dot{\Theta}_{(n-1)n} = \frac{\Theta_n - \Theta_{n-1}}{t_{c(n-1)n} - \frac{1}{2}t_n} \quad (4.75)$$

$$t_{(n-1)n} = t_{c(n-1)n} - t_n - \frac{1}{2}t_{n-1} \quad (4.76)$$

S tako podanimi enačbami lahko določimo za multisegmentno trajektorijo ustrezne čase za prehode in hitrosti linearnih delov. Uporabnik ponavadi izbere vmesne točke in želeni čas za trajanje giba. Sistem vodenja nam nato določi ustrezne pospeške na vsakem prehodu. Včasih sistem sam določa hitrosti in jih uporabnik »potrdi«, določi na tej podlagi čase med točkami. Važno je, da moramo v prehodih uporabiti (in imeti na voljo) dovolj pospeška, da zadostimo kriteriju, v nasprotnem mora sistem javiti prekoračenje in je zahtevano nemogoče.

Primer:

Trajektorija sklepa je definirana z točkami  $T_1, T_2, T_3, T_4$ , ki so  $\Theta(10, 35, 25, 10)$  cm ali .

Časi za prehod med točkami so  $2s, 1s, 3s$ .

Na razpolago nudimo pospešek  $50/s^2$ . Določi hitrosti segmentov in prehodne segmente na trajektoriji prehodnih delov.

$$\ddot{\Theta}_1 = 50/s^2$$

$$\text{računamo } t_1 = 2 - \sqrt{4 - \frac{2(35 - 10)}{50}} = 0,27s$$

$$\dot{\Theta}_{12} = \frac{35 - 10}{2 - 0.5(0.27)} = 13,50/s$$

$$\dot{\Theta}_{23} = \frac{25 - 35}{1} = -10/s$$

Uporabimo  $\ddot{\Theta}_1 = 50/s^2$  in računamo  $t_1$  (vmesne točke)

$$t_2 = \frac{-10,0 - 13,5}{-50} = 0,47s.$$

Linearni del prvega segmenta bo trajal

$$t_{1,2} = 2 - 0,27 - \frac{1}{2}0,47 = 1,5s$$

$$\ddot{\Theta}_4 = 50 \rightarrow t_4 = 3 - \sqrt{9 + \frac{2(10 - 25)}{50}} = 0,102s$$

$$\dot{\Theta}_{3,4} = \frac{10 - 25}{3 - 0,050} = -5,1/s$$

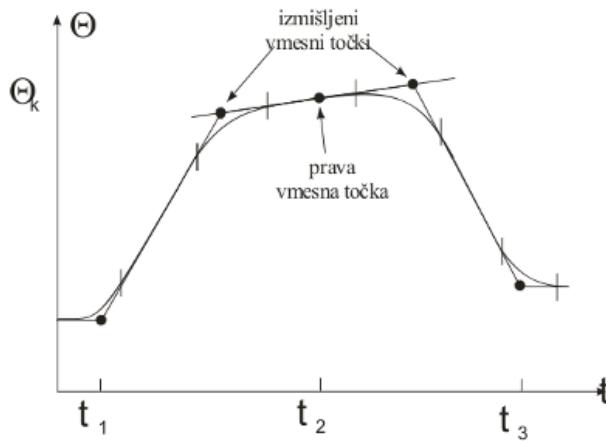
$$\ddot{\Theta}_3 = 50 \rightarrow t_3 = \frac{-5,1 - (-10,0)}{50} = 0,098s$$

$$t_{23} = 1 - \frac{1}{2}0,47 - \frac{1}{2}0,098 = 0,716s$$

$$t_{3,4} = 3 - \frac{1}{2}0,098 - 0,102 = 2,849s$$

Izračunani rezultati so načrt trajektorije za prehod in časi za izvajanje prehodov in ustrezne vrednosti  $\dot{\Theta}, \ddot{\Theta}$ .

V primeru, da želimo, da robot preleti vmesno točko z željeno hitrostjo, tedaj to točko postavimo na linearni del trajektorije in pred njo postavimo dve novi, dodatni vmesni točki ter račun izvedemo enako kot do sedaj. Glej sliko 4.20.



Slika 4.20: Primer z vmesno točko in želeno hitrostjo.

Če ne predpišemo hitrosti preleta v tej točki, bo sistem določil neko srednjo vrednost, glede na vstopni in izstopni segment. Podobno postopamo tudi če želimo zagotoviti, da bo sistem šel točno skozi to točko. Kot vemo, vmesni točki med dvema linearima deloma, kjer se hitrost lomi, robot obide točke po neki krivulji 3. reda in je zato storjena napaka.

#### 4.2.5 Trajektorija podana v zunanjih koordinatah(WCS)

Te trajektorije sestavljamo s podajanjem točk, ki jih lahko povezujejo premice, krožnice, sinusoide itd. Za vsako točko je podana pozicija in orientacija. Interpolacijo lahko oblikujemo v zunanjih koordinatah, kar je dokaj računalniško zahtevno, če pomislimo, da podamo še hitrost na poti, saj mora sistem preko inverzne kinematike določiti trajektorijo v sklepnih (notranjih) koordinatah ter preveriti, ali nismo prekoračili omejitve. Ves izračun je zahteven, kajti opravljati se mora v času vzorčne frekvence (path update rate), ki je lahko 20 – 400 Hz. Obstaja več metod za generiranje trajektorij v zunanjih koordinatah.

Problemi, ki otežujejo planiranje in interpolacijo v zunanjih koordinatah so standardni: omejitve v delovnem prostoru, singularnosti in hitrostne omejitve, kajti v nekaterih točkah, čeprav izvajamo linearen potek, to lahko pomeni za nek sklep zahtevo za neskončno hitrost. Komercialni roboti v fazi planiranja opomnijo, da je nekaj nemogoče, oziroma v izvajjanju ustavijo robot. Včasih zaradi hitrostnih zahtev znižamo v bližini takšnih točk hitrost gibanja, tako da prestavimo hitrost v prizadetem sklepu na sprejemljivo vrednost. Seveda je zaradi tega ohranjen prostorski potek, ne pa tudi časovni

potek.

Običajno se med sprotnim računanjem trajektorije in interpolacijo (path generation at real time) iz spomina čitajo končne in vmesne vrednosti točk med dvema skrajnjima točkama podanima v zunanjih koordinatah, zatem se določijo vrednosti  $\ddot{\Theta}$ ,  $\dot{\Theta}$ ,  $\Theta$ . Te točke služijo kot robni pogoji, ki so uporabljeni za interpolacijo vmesnih točk, kot smo že pokazali. Ko je ta izračun končan, se postavi interpolacijski čas na  $t = 0$  in prevzame nova točka ter ustrezni podprogram se ponovi.

Pri naših predpostavkah za pospešek  $\ddot{\Theta}$  smo naredili zelo veliko poenostavitev. Doseženi pospešek je bistveno odvisen od dinamike aktuatorjev in seveda dinamike robota z bremenom, to je celotna dinamika robota. Problem se rešuje na ta način, da za dano prostorsko trajektorijo iščemo časovni potek, tako da bo pot opravil manipulator v minimalnem času. To je mejna rešitev, vse počasnejše so možne, rešujemo numerično.

Pri planiranju trajektorij na visokem nivoju, lahko bi rekli inteligentnem planiranju povemo trajektorijo robotu, ki samodejno določi vmesne točke in to tako, da se ne zadanemo v kakšno oviro. Robot na podlagi notranjega shranjenega modela in delovnega okolja določi trajektorijo. Planiranje postane še bolj zanimivo, če so v istem okolju še tekoči trakovi, kjer se predmeti premikajo in so drugi roboti na delu. Problem sodelovanja dveh robotov, bimanualni roboti in integracija vidne informacije za gibanje tukaj ni obravnavano podrobno.

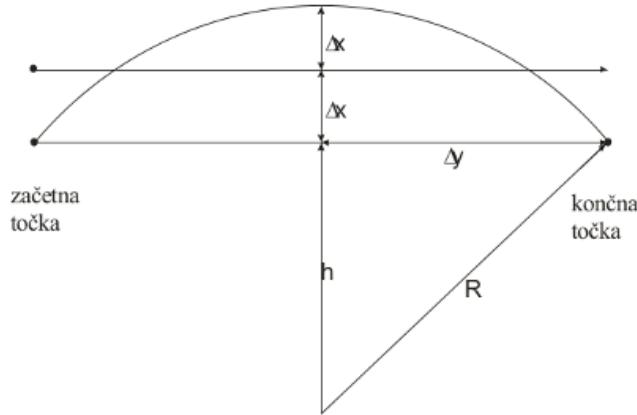
#### 4.2.6 Kakšno interpolacijsko razdaljo potrebujemo za želeno točnost

Pri interpolaciji nismo do sedaj specificirali števila točk, ki ga potrebujemo, da dosežemo želeno točnost pozicioniranje gibanja. V ta namen poglejmo sliko 4.21, ki prikazuje premico zaradi napake s krožnico, ki ima radij  $R$ .

Na sliki  $\Delta x$  pomeni maksimalno pozitivno in negativno dovoljeno odstopanje od premice. Torej bo  $2\Delta y$  maksimalna distanca na premici, preko katere smemo gibanje po premici nadomestiti z gibanjem po krožnici, da bo maksimalna storjena napaka ravno  $\pm\Delta x$ .

Če rešujemo desni trikotnik v sliki imamo

$$R^2 = \Delta y^2 + h^2 \quad (4.77)$$



Slika 4.21: Začetna in končna točka z dodatnimi spremenljivkami.

oziroma

$$h = R - 2\Delta x \quad (4.78)$$

in

$$R^2 = \Delta y^2 + (R - 2\Delta x)^2 \quad (4.79)$$

kar da

$$\Delta y^2 = 4R \cdot \Delta x - 4\Delta x^2. \quad (4.80)$$

Primer:

Za tipičen robot znaša  $R$  npr.  $1,5m$ . Če robot sledi tekoči trak (premično gibanje), je dopuščena napaka, da predmet na traku še zagrabimo npr.  $1mm$  (napaka  $\pm 1mm$ ). Torej

$$\begin{aligned} \Delta y^2 &= 4 \cdot 1.5 \cdot 10^{-3} - 4 \cdot 10^{-3} \cdot 10^{-3} m^2 = \\ &= 6 \cdot 10^{-3} - 4 \cdot 10^{-6} m^2 = 6 \cdot 10^{-3} m^2 = 6 \cdot 10^3 mm^2 \\ \Delta y &= \sqrt{60} \cdot 10 = 7,75 \cdot 10 mm = 7,75 cm \\ 2\Delta y &= 2x \cdot 7,75 cm = 15,5 cm \end{aligned}$$

Torej moramo postaviti interpolacijske točke narazen za največ  $15,5cm$ .

#### 4.2.7 Bezier krivulje

Bézierjeve krivulje srečamo v računalniški grafični CAD (tudi Photoshop, Flash, Illustrator, Inkscape), kot tudi lahko v 3D trajektorijah gibanja robotov, v splošnem pa za 2D interpolacijo. P. Bézier je v zgodnjih šestdesetih letih prejšnjega stoletja uporabljal podoben zapis v Renaultovem sistemu

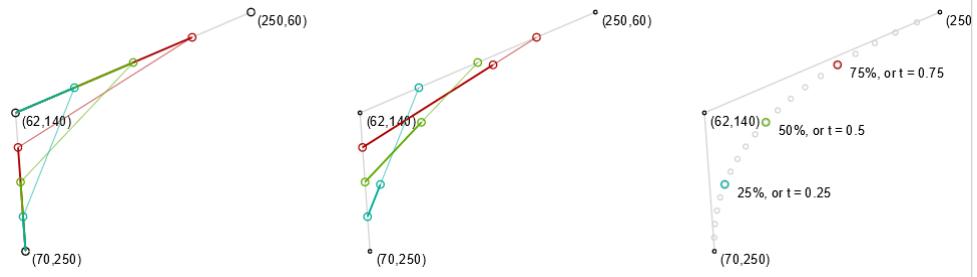
UNISURF CAD. Bézier krivulje in Bézier površine je patentiral in jih populariziral, vendar ni bil prvi. Matematična osnova za Bézier krivulje, Bernstein polinomi so znani od leta 1912. Študijo o teh krivuljah je kasneje leta 1959 opravil P. de Casteljau pri Citroënu, vendar mu objave niso dovolili. Uporabnost Bézierjevih krivulj izhaja iz številnih geometrijskih in analitičnih lastnosti. Obstajajo elegantni in učinkoviti algoritmi za vrednotenje, diferenciacijo, razdelitev krivulj in pretvorbo v druge uporabne predstavitev. Poleg tega Bézierjeve krivulje interpolirajo svoje končne točke ( $P_0$  in  $P_n$ ), imajo odvode v končnih točkah, odvisne le od prvih nekaj ali zadnjih nekaj kontrolnih točk (npr.  $B'(0) = n(P_1 - P_0)$ ), so invariantne (transformirana krivulja je enaka krivulji, ki jo dobimo s transformacijo točk in zatem računanjem Bézier krivulje), ležijo v konveksnem trupu njihovih kontrolnih točk in nihajo ne več kot delno linearne interpolant do njihovih kontrolnih točk. Bézierjeve krivulje se na naraven način razširijo tudi na površine (ki so še bolj pomembne za oblikovanje, prim. Bézierjevo površino), prostornine itd. Obstaja tudi racionalna razširitev Bézierjevih krivulj, ki omogoča natančen prikaz stožčastih odsekov. Zaradi teh lastnosti in algoritmov so krivulje priljubljene na področjih, kjer je oblika pomembna. Primeri takšnih področij poleg CAD vključujejo tudi računalniško animacijo ter oblikovanje pisav. Zato so danes zelo pogoste pri urejanju gibov ter animacij. True Type pisava uporablja kvadratne, postScript in SVG pa kubične Bézier krivulje v poznih petdesetih in zgodnjih šestdesetih letih.

Bézier krivulje so posledica linearne interpolacije. Začnemo z dvema končnima točkama, ki jima želimo dodati vmesno točko. Poznamo prvo točko in seveda razdaljo med obema končnima točkama, vmesno točko želimo dodati 20% razdalje stran od prve točke: Ob

$$\begin{pmatrix} p_1 = \text{prva tocka} \\ p_2 = \text{druga tocka} \\ razdalja = (p_2 - p_1) \\ razmerje = \frac{\text{procenci}}{100} \end{pmatrix} \text{ nova tocka} = p_1 + razdalja \cdot razmerje \quad (4.81)$$

Primer z eno več, s tremi točkami definira dve premici, linearna interpolacija na vsaki premici definira dve točki, med katerima je spet ena premica. Ponovna linearna interpolacija na tej premici definira eno točko. In ta točka definira Bézier krivuljo.

Bézier krivulje so oblika parametrične funkcije. Parametrične funkcije pravzaprav preslikajo eden ali več vhodov v samo eno izhodno vrednost. Bézier krivulje so samo eden od mnogih oblik parametričnih funkcij in so značilne po uporabi ene same osnovne funkcije za vse izhodne vrednosti. Lahko bi za os  $x$  in os  $y$  uporabljali različne funkcije (npr.  $\cos, \sin$ ), vendar Bézier krivulje uporabljajo binomski polinom za  $x$  in  $y$  izhoda (koordinati).



Slika 4.22: Linarna interpolacija povzroči nastanek Bézier krivulje

Kaj so binomski polinomi? Preprost polinom tretje stopnje ima obliko

$$f(x) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d \quad (4.82)$$

Bézier krivulje uporabljajo neodvisni parameter  $t$  namesto  $x$ ,  $0 \leq t \leq 1$ , medtem ko koeficienti  $a, b$ , itd. zavzamejo binomsko formo.

$$\text{linearna} = (1 - t) + t \quad (4.83)$$

$$\text{kvadratna} = (1 - t)^2 + 2 \cdot (1 - t) \cdot t + t^2 \quad (4.84)$$

$$\text{kubicna} = (1 - t)^3 + 3 \cdot (1 - t)^2 \cdot t + 3 \cdot (1 - t) \cdot t^2 + t^3 \quad (4.85)$$

V osnovni algebri binomski izrek (ali binomska ekspanzija) opisuje algebrično razširitev potence binoma. V skladu z izrekom je mogoče polinom  $(x + y)^n$  razširiti v vsoto, ki vključuje izraze z eksponenti  $ax^b y^c$ , kjer sta eksponenta pri  $x$  in  $y$  nenegativni celi števili ter je vsota  $a + b = n$ , koeficient  $a$  je specifično pozitivno celo število, odvisno od  $n$  in  $b$ . Koeficient  $a$  v  $ax^b y^c$  je poimenovan kot binomski koeficient  $\binom{n}{b}$  ali  $\binom{n}{c}$ . Te koeficiente je mogoče razvrstiti v Pascalov trikotnik.

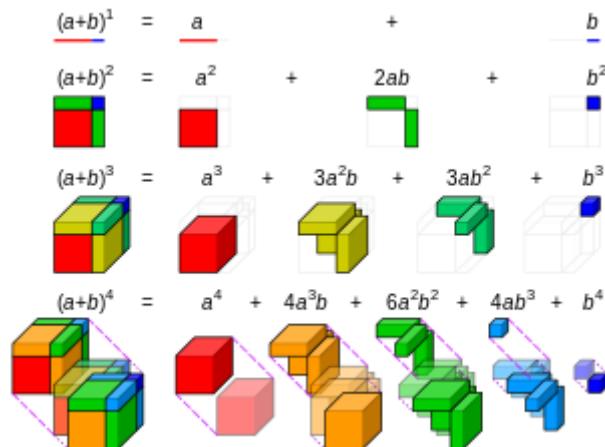
Za pozitivne vrednosti  $a$  in  $b$ , je pri binomskem teoremu  $n = 2$  mogoče ponazoriti situacijo geometrijsko na zelo transparenten način. Kvadrat s stranico  $a + b$  je mogoče izrezati najprej v kvadrat s stranico  $a$ , kvadrat s stranico  $b$  in dva pravokotnika s stranicama  $a$  in  $b$ . Pri  $n = 3$  je kocka s stranico  $a + b$  sestavljena iz kocke s stranico  $a$ , treh prostornin z dimenzijami  $a \cdot a \cdot b$  in treh  $a \cdot b \cdot b$  škatelj.

Ključ:  $2 = 1 + 1, 3 = 2 + 1$  in  $1 + 2, 6 = 3 + 3$ . V vsaki vrstici začnemo z 1, vse vmesne vrednosti so samo vsota dveh števil nad tem mestom.

Koeficient pred členom  $x^{n-k} y^k$  je tako podan s formulo

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1

Slika 4.23: Pascalov trikotnik z vrsticami 0 do 7



Slika 4.24: Vizualizacija binomske razširitve do četrtega reda (n=4)

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}, \quad (4.86)$$

$$\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k(k-1)\dots 1} = \prod_{l=1}^k \frac{n-l+1}{l} = \prod_{l=0}^{k-1} \frac{n-l}{k-l} \quad (4.87)$$

Čeprav je v izrazu deljenje, je binomski koeficient  $\binom{n}{k}$  na koncu celo število (integer).

Podobno preprosto je mogoče razložiti tudi polinomske člene. Če za trenutek uporabimo  $(1-t) = a$  in  $t = b$  ter umaknemo koeficiente dobimo

$$\text{linearna} = a + b \quad (4.88)$$

$$\text{kvadratna} = a.a + a.b + b.b \quad (4.89)$$

$$\text{kubicna} = a.a.a + a.a.b + a.b.b + b.b.b \quad (4.90)$$

V tem zapisu so samo med seboj sešteji produkti  $a$  in  $b$ . To je v splošni obliki mogoče zapisati kot

$$\text{Bezier}(n, t) = \sum_{i=0}^n \binom{n}{i} \cdot (1-t)^{n-i} \cdot t^i \quad (4.91)$$

kar predstavlja Bézier krivuljo reda  $n$  s parametrom  $0 \leq t \leq 1$ . Sestavljena je iz binomskega dela in polinomskega člena.

Ta način zapisa je poimenovan kot eksplisitna definicija. Morda je nekoličko bolj nazoren ekvivalenten daljši zapis

$$\begin{aligned} B(t) &= \sum_{i=0}^n \binom{n}{i} \cdot (1-t)^{n-i} \cdot t^i P_i, \\ &= (1-t)^n P_0 + \binom{n}{1} (1-t)^{n-1} t P_1 + \dots + \binom{n}{n-1} (1-t) t^{n-1} P_{n-1} + t^n P_n, \\ &\quad 0 \leq t \leq 1 \end{aligned} \quad (4.92)$$

$B_{P_0 P_1 \dots P_n}$  je Bézier krivulja določena s točkami  $P_0, P_1, \dots, P_n$ .

Linearna Bézier krivulja, torej  $n = 1$  z začetno in končno točko ima zapis

$$B_{P_0 P_1}(t) = (1-t)P_0 + tP_1, \quad 0 \leq t \leq 1 \quad (4.93)$$

Kvadratna Bézier krivulja torej  $n = 2$ , s tremi točkami, ima zapis

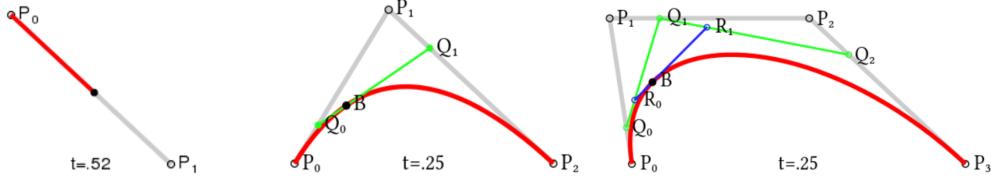
$$\begin{aligned} B_{P_0 P_1 P_2}(t) &= (1-t)B_{P_0 P_1}(t) + tB_{P_1 P_2}(t), \\ &= (1-t)((1-t)P_0 + tP_1) + t((1-t)P_1 + tP_2), \\ &= (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, \\ &\quad 0 \leq t \leq 1 \end{aligned} \quad (4.94)$$

Kubična Bézier krivulja, s štirimi točkami, torej  $n = 3$  ima zapis

$$\begin{aligned} B_{P_0 P_1 P_2 P_3}(t) &= (1-t)B_{P_0 P_1 P_2}(t) + tB_{P_1 P_2 P_3}(t), \\ &= (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, \\ &\quad 0 \leq t \leq 1 \end{aligned} \quad (4.95)$$

Kubična Bézier krivulja z začetno točko  $P_0$  (120, 160), kontrolno točko  $P_1$  (35, 200) in  $P_2$  (220, 260) ter končno točko  $P_3$  (220, 40) ima obliko

$$\begin{aligned} x &= 120.(1-t)^3 + 35.3.(1-t)^2 t + 220.3.(1-t)t^2 + 220.t^3, \\ y &= 160.(1-t)^3 + 200.3.(1-t)^2 t + 260.3.(1-t)t^2 + 40.t^3 \end{aligned} \quad (4.96)$$



Slika 4.25: Prikaz konstrukcije linearne, kvadratične in kubične Bézier krivulje

Pri  $n = 5$  ima obliko

$$B(t) = (1-t)^5 P_0 + 5t(1-t)^4 P_1 + 10t^2(1-t)^3 P_2 + 10t^3(1-t)^2 P_3 + 5t^4(1-t)P_4 + t^5 P_5, \quad 0 \leq t \leq 1 \quad (4.97)$$

Iz zgornjih zapisov za linearno, kvadratno in kubično krivuljo lahko prepoznamo tudi rekurzivno obliko Bézier krivulj. Rekurzivno lahko za Bézier krivuljo reda  $n$  uporabimo linearno kombinacijo para spremljajočih točk, ki pa sta posledica dveh Bézier krivulj reda  $n - 1$ . Bralec v zapisu druge vrstice lahko opazi indeks do  $n - 1$  v prvi in indeks do  $n$  v drugi funkciji. Rekursivnost je vidna tudi na grafični predstavitev kvadratične in kubične Bézier krivulje.

$$\begin{aligned} B_{P_0}(t) &= P_0, \\ B(t) &= B_{P_0 P_1 \dots P_n}(t) = (1-t)B_{P_0 P_1 \dots P_{n-1}}(t) + tB_{P_1 P_2 \dots P_n}(t) \end{aligned} \quad (4.98)$$

Prav enak zapis za Bézier krivuljo reda  $n$  je v obliki

$$B(t) = \sum_{i=0}^n b_{i,n}(t) P_i, \quad 0 \leq t \leq 1 \quad (4.99)$$

kjer je del tega zapisa

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n \quad (4.100)$$

poimenovan kot Bernstein polinom.  $P_0, P_1, \dots, P_n$  so kontrolne točke. Najpreprostejša krivulja je že samo med točkama  $P_0$  in  $P_1$  pri  $n = 1$ .  $t^0 = 1, (1-t)^0$ , binomski koeficienti se izračunajo kot

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (4.101)$$

torej skupno

$$B(t) = \sum_{i=0}^n \frac{n!}{i!(n-i)!} (1-t)^{n-i} t^i P_i \quad (4.102)$$

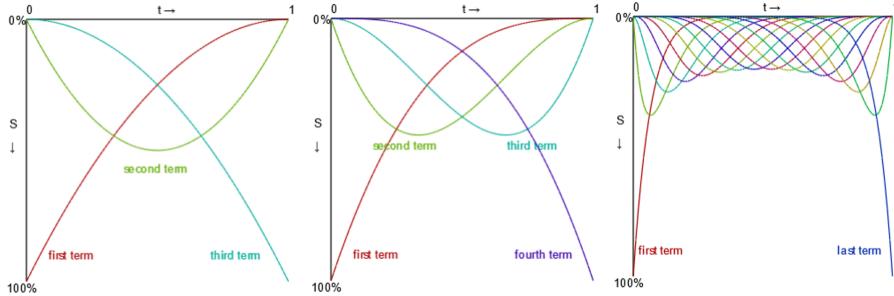
$$\begin{array}{c}
 \begin{matrix} & & 1 \\ & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 3 & 3 & 1 \\ \boxed{1} & 4 & 6 & 4 & 1 \\ \hline 1 & 5 & 10 & 10 & 5 & 1 \end{matrix}
 \end{array}
 \quad \mathbf{B}(t) = \boxed{\begin{aligned}
 & (1-t)^4 \mathbf{P}_0 \\
 & + 4t (1-t)^3 \mathbf{P}_1 \\
 & + 6t^2 (1-t)^2 \mathbf{P}_2 \\
 & + 4t^3 (1-t) \mathbf{P}_3 \\
 & + t^4 \mathbf{P}_4
 \end{aligned}}
 \quad \mathbf{B}(t) = \boxed{\begin{aligned}
 & (1-t)^5 \mathbf{P}_0 \\
 & + 5t (1-t)^4 \mathbf{P}_1 \\
 & + 10t^2 (1-t)^3 \mathbf{P}_2 \\
 & + 10t^3 (1-t)^2 \mathbf{P}_3 \\
 & + 5t^4 (1-t) \mathbf{P}_4 \\
 & + t^5 \mathbf{P}_5
 \end{aligned}}$$

Slika 4.26: Bézier krivulji za n=4 in n=5

Mogoči so še drugi nazorni načini zapisa Bézier krivulj

Bézierjeve krivulje so interpolacijske funkcije, tako kot vsi zlepki. To pomeni, da vzamejo niz točk in ustvarijo vrednosti nekje med temi točkami. Ena od posledic tega je, da nikoli ni mogoče ustvariti točke, ki leži zunaj obrisa za kontrolne točke, ta obris se imenuje trup krivulje. Zanimivi so prispevki posameznih točk k vrednosti opazovane funkcije. To prikazuje pomembnost posameznih točk na različnih odsekih krivulje.

Naslednji grafi prikazujejo interpolacijske funkcije za kvadratne in kubične krivulje, pri čemer je spremenljivka  $S$  na vertikalni osi velikost prispevka točke k skupni vsoti Bézierjeve funkcije.

Slika 4.27: Interpolacijske funkcije za kvadratne in kubične krivulje ter  $n = 15$ 

Prikazana je tudi funkcija interpolacije za Bézierjevo funkcijo 15. reda. Kot je razvidno z grafa, začetna in končna točka bistveno bolj prispevata k obliku krivulje kot katera koli druga točka v naboru kontrolnih točk.

Če želimo spremeniti krivuljo, moramo spremeniti uteži vsake točke in tako učinkovito vplivati na interpolacijo. Način za to je enostaven: vsako točko pomnožimo z vrednostjo, ki spremeni njen prispevek. Te vrednosti običajno poimenujemo uteži  $\omega_i$  in jih lahko dodamo prvotni Bézierjevi funkciji:

$$\sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \omega_i, \quad (4.103)$$

ki ima tako poleg binomskega člena še polinomski člen ter na koncu utež. Uteži so pravzaprav samo koordinatne vrednosti, ki jih želimo od naše krivulje.

Še nadalje lahko vplivamo na Bézierjeve krivulje tako, da jih racionaliziramo. To pomeni, da poleg vrednosti uteži, dodamo še vrednost razmerja, to razmerje definira kako močno vsaka koordinata vpliva na krivuljo.

$$\text{Racionaliziran Bezier}(n, t) = \left( \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \omega_i \cdot \text{razmerje}_i \right) \cdot \frac{1}{\text{Osnova}_{(n,t)}}, \quad (4.104)$$

Prvi nov koeficient predstavlja razmerje, to je nov faktor množenja za vsako koordinato. Če so vrednosti razmerja  $[1, 0,5, 0,5, 1]$ , potem je  $\text{razmerje}_0 = 1$ ,  $\text{razmerje}_1 = 0,5$  itd. Nato drugi člen normalizira vsa ta množenja tako, da vrednost deli z osnovno vrednostjo Bézierjeve krivulje. To vrednost dobimo z izračunavanjem krivulje brez kakršnihkoli uteži, vendar z razmerji:

$$\text{Osnova}_{(n,t)} = \left( \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \right). \quad (4.105)$$

Skupno bi torej zapisali kot

$$B(t) = \frac{\sum_{i=0}^n b_{i,n}(t) P_i \omega_i}{\sum_{i=0}^n b_{i,n}(t) \omega_i} \quad (4.106)$$

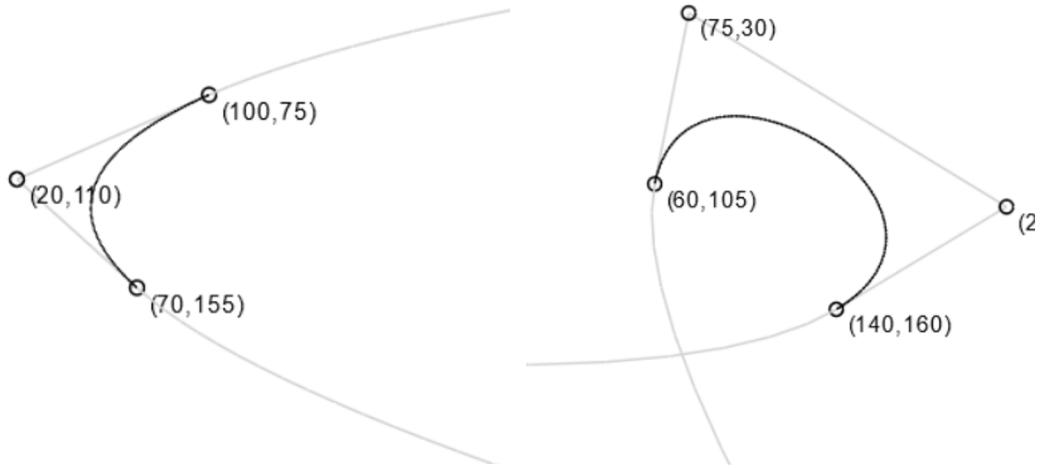
ozziroma

$$B(t) = \frac{\sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i P_i \omega_i}{\sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \omega_i} \quad (4.107)$$

Vrednost razmerja vpliva šibkeje pri vrednostih bližje 0 ter ima večji vpliv na pripadajočo koordinato pri večjih vrednostih. Vrednosti razmerja lahko ponazorimo kot gravitacijo vsake koordinate: večja je gravitacija, bližje tej koordinati bo želela biti krivulja. Nobene spremembe ne opazimo, če preprosto povečamo ali zmanjšamo vsa razmerja za enak znesek. Podobno je pri gravitaciji.

V primeru Bézierjevih krivulj podaljšanje siceršnjega intervala  $0 \leq t \leq 1$  preprosto povzroči nadaljevanje krivulje na prej ne videno področje. Bézierjeve krivulje so preprosto odseki neke polinomske funkcije, kot posledica pa pri širšem intervalu preprosto vidimo več krivulje.

Obe grafiki prikazujeta Bézier krivulji, narisani na običajen način, pa tudi krivulje, na katerih ležijo. Te vrednosti bi dobili pri povečanju vrednosti  $t$ . Kot vidimo, se v preostalem delu krivulje skriva veliko več oblike, ki jo lahko



Slika 4.28: Kvadratna in kubična Bézier krivulji podaljšani proti neskončnemu intervalu

modeliramo s premikanjem točk krivulje.

Bézierjeve krivulje lahko predstavimo tudi kot matrične operacije z zapisom Bézierjeve formule v obliki polinomske osnovne funkcije in matrike koeficientov ter dejanskih koordinat kot matrike.  $P \dots$  se lahko nanaša na koordinate z eno ali več dimenzijami:

$$B(t) = P_1 \cdot (1-t)^3 + P_2 \cdot 3(1-t)^2 \cdot t + P_3 \cdot 3 \cdot (1-t)t^2 + P_4 \cdot t^3 \quad (4.108)$$

Začasno zanemarimo točke, torej imamo

$$B(t) = (1-t)^3 + 3(1-t)^2 \cdot t + 3 \cdot (1-t)t^2 + t^3 \quad (4.109)$$

To pa samo preoblikujemo v zapis s štirimi vrsticami

$$\begin{aligned} \dots &= (1-t)^3 \\ &+ 3 \cdot (1-t)^2 \cdot t \\ &+ 3 \cdot (1-t)t^2 \\ &+ t^3 \end{aligned} \quad (4.110)$$

Razširimo na posamezne oklepaje in pomnožimo

$$\begin{aligned} \dots &= (1-t) \cdot (1-t) \cdot (1-t) = -t^3 + 3t^2 - 3t + 1 \\ &+ 3 \cdot (1-t) \cdot (1-t) \cdot t = 3t^3 - 6t^2 + 3t \\ &+ 3 \cdot (1-t) \cdot t \cdot t = -3t^3 + 3t^2 \\ &+ t \cdot t \cdot t = +t^3 \end{aligned} \quad (4.111)$$

Manjkajoče člene nadomestimo z 0 in 1 koeficienti:

$$\begin{aligned} \dots &= -1 \cdot t^3 + 3 \cdot t^2 - 3 \cdot t + 1 \\ &+ 3 \cdot t^3 - 6 \cdot t^2 + 3 \cdot t + 0 \\ &- 3 \cdot t^3 + 3 \cdot t^2 + 0 \cdot t + 0 \\ &+ 1 \cdot t^3 + 0 \cdot t^2 + 0 \cdot t + 0 \end{aligned} \quad (4.112)$$

Te štiri vrstice lahko zapišemo kot štiri skalarne produkte

$$[t^3 \ t^2 \ t \ 1] \cdot \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} + [t^3 \ t^2 \ t \ 1] \cdot \begin{bmatrix} 3 \\ -6 \\ 3 \\ 0 \end{bmatrix} + [t^3 \ t^2 \ t \ 1] \cdot \begin{bmatrix} -3 \\ 3 \\ 0 \\ 0 \end{bmatrix} + [t^3 \ t^2 \ t \ 1] \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (4.113)$$

kar po združitvi v eno matriko vodi do

$$[t^3 \ t^2 \ t \ 1] \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (4.114)$$

Tovrsten zapis s polinomsko osnovo je običajno zabeležen z osnovami v naraščajočem vrstnem redu, kar pomeni, da moramo matriko t obrniti vodoravno in veliko mešalno matriko na glavo:

$$[1 \ t \ t^2 \ t^3] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}. \quad (4.115)$$

Končno lahko dodamo še točke za končno obliko reda 3.

$$B(t) = [1 \ t \ t^2 \ t^3] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}. \quad (4.116)$$

Kvadratična krivulja ima sledeči zapis

$$B(t) = [1 \ t \ t^2] \cdot \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}. \quad (4.117)$$

Po vstavitvi vrednosti za  $t$  in ustremnem množenju dobimo popolnoma enake vrednosti kot v originalni funkciji.

Matrične predstavitve omogočajo odkrivanje stvari o funkcijah, ki bi jih sicer težko prepoznali. Izkazalo se je, da krivulje tvorijo trikotne matrike in imajo determinanto, enako zmnožku dejanskih koordinat, ki jih uporabljam za krivuljo. Obstaja tudi inverzna matrika, kar povzroči veliko lepih lastnosti.

### 4.2.8 Lokalno utežena regresija (LWR)

#### 4.2.8.1 Metoda najmanjših kvadratov

V številnih aplikacijah sodelujočih robotov in humanoidne robotike se metoda najmanjših kvadratov ter utežena metoda najmanjših kvadratov oz. linearna regresija pojavlja v takšni ali drugačni obliki, od preprostih do kompleksnih problemov. To je predvsem zato, ker lahko problem nelinearne regresije rešimo učinkovito in na praktičen način lokalno z nelinearno metodo. V nadaljevanju je najprej omenjena linearna regresija in njena utežena varianta (Local weighted Regression, LWR), ki je osnova še za druge metode.

Število vhodov in izhodov v robotske aplikacije je le zelo redko majhno. Komplet večdimenzionalnih vhodnih podatkov lahko zapišemo kot  $\mathbf{X}^I \in \mathbb{R}^{N \times D^I}$ , večdimenzionalne izhodne podatke pa kot  $\mathbf{X}^O \in \mathbb{R}^{N \times D^O}$ .  $N$  je število podatkovnih točk,  $D^I$  je dimenzija vhodnih podatkov, in  $D^O$  je dimenzija izhodnih podatkov. Podatki so običajno sestavljeni iz več poskusov, posnetkov ali demonstracij. Z združitvijo  $M$  posnetkov z  $T_m$  podatkovnimi točkami dobimo skupno

$$N = \sum_{m=1}^M T_m. \quad (4.118)$$

Linearna regresija išče vrednost  $\mathbf{A} \in \mathbb{R}^{D^I \times D^O}$ , tako da je  $\mathbf{X}^O = \mathbf{X}^I \mathbf{A}$ . Rešitev je mogoče poiskati z minimizacijo  $L_2$  norme vrednosti napak (residuals)

$$\begin{aligned} \hat{\mathbf{A}} &= \arg \min_{\mathbf{A}} \|\mathbf{X}^O - \mathbf{X}^I \mathbf{A}\|_F^2 \\ &= \arg \min_{\mathbf{A}} \text{tr}((\mathbf{X}^O - \mathbf{X}^I \mathbf{A})^T (\mathbf{X}^O - \mathbf{X}^I \mathbf{A})) \\ &= \arg \min_{\mathbf{A}} (\mathbf{X}^{OT} \mathbf{X}^O - 2\mathbf{A}^T \mathbf{X}^{IT} \mathbf{X}^O + \mathbf{A}^T \mathbf{X}^{IT} \mathbf{X}^I \mathbf{A}), \end{aligned} \quad (4.119)$$

ki jo razrešimo z odvajanjem po  $A$  in iskanjem ekstrema z enačenjem z ničlo

$$\begin{aligned} -2\mathbf{X}^{IT} \mathbf{X}^O + 2\mathbf{X}^{IT} \mathbf{X}^I \mathbf{A} &= 0, \\ \hat{\mathbf{A}} &= (\mathbf{X}^{IT} \mathbf{X}^I)^{-1} \mathbf{X}^{IT} \mathbf{X}^O. \end{aligned} \quad (4.120)$$

Preostala napaka parametrov je dana kot

$$\hat{\mathbf{A}} = \mathbf{X}^{IT} (\mathbf{X}^{IT} \mathbf{X}^I)^{-1}. \quad (4.121)$$

Obstaja še alternativna rešitev v obliki

$$\hat{\mathbf{A}} = \mathbf{X}^{IT} (\mathbf{X}^{IT} \mathbf{X}^I)^{-1} \mathbf{X}^O. \quad (4.122)$$

Rešitev pri  $D^I = D^O = 1$  je mogoče ponazoriti v 2D kot prilagajanje premice na množico podatkovnih točk. Za vhodne podatke ni potrebno, da so linearni, to pričakujemo samo od parametrov, ki definirajo najboljše prileganje. Na primer, pri prilagajanju trajektorij, kjer  $\mathbf{X}^I$  predstavlja časovno serijo, lahko v  $\mathbf{X}^O = \mathbf{X}^I \mathbf{A}$  uporabimo različne oblike vhodov, kot na primer  $\mathbf{X}^I = [t_1, t_2, \dots, t_N]^T$  ali  $\mathbf{X}^I = [t_1^2, t_2^2, \dots, t_N^2]^T$ , pri čemer je model še vedno linearen v  $\mathbf{X}$  parametrih.

Kot tipičen primer je prilaganje s polinomi  $[t_1, t_2, \dots]$  kot neodvisnimi spremenljivkami v večdimenzionalnem regresijskem primeru, kjer so vhodni podatki zapolnjeni z  $D^I - 1$  odvodom (vključno z  $t^0 = 1$ , ki je povezan z odvodi). Za polinom tretje stopnje ( $D^I = 4$ ) imamo

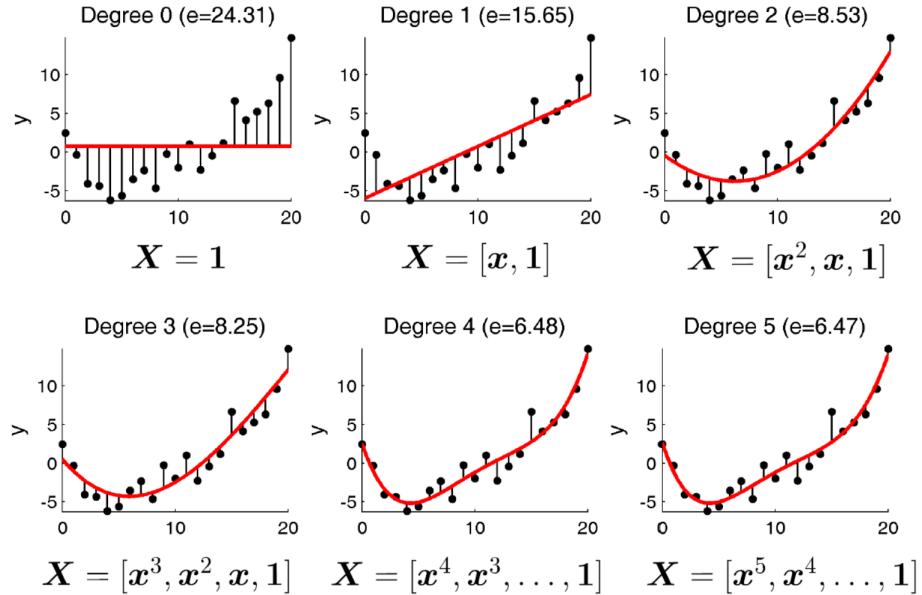
$$\mathbf{X}^I = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & t_N & t_N^2 & t_N^3 \end{bmatrix}. \quad (4.123)$$

Polinomska regresija je primer uporabe regresije, kjer so uporabljeni polinomske osnovne funkcije za modeliranje razmerja med dvema veličinama. Pomanjkljivost polinomskih prilagoditev je nelokalnost funkcij. Prav zaradi tega so polinomske funkcije pogosto uporabljene v učenju algoritmov, skupaj z ostalimi baznimi funkcijami kot so spline, valovne funkcije ali radialne bazne funkcije (Radial Basis Function, RBF). Posebej nazadnje omenjene RBF so zelo popularne v robotiki, saj so preproste, sicer pa omogočajo dekompozicijo robotskih gibov v odseke z parcialnimi gibalnimi primitivi, ki so med seboj gladko povezani. Da je to mogoče, je zapis najprej nekoliko spremenjen v uteženo obliko najmanjših kvadratov (weighted least square)

$$\hat{\mathbf{A}} = \arg \min_{\mathbf{A}} \text{tr}((\mathbf{X}^O - \mathbf{X}^I \mathbf{A})^T (\mathbf{X}^O - \mathbf{X}^I \mathbf{A})) \quad (4.124)$$

in

$$\hat{\mathbf{A}} = (\mathbf{X}^{IT} \mathbf{W} \mathbf{X}^I)^{-1} \mathbf{X}^{IT} \mathbf{W} \mathbf{X}^O. \quad (4.125)$$



Slika 4.29: Prilagajanje polinomov različnih stopenj

Pri tem je matrika uteži enaka  $\mathbf{W} \in \mathbb{R}^{N \times N}$  ter residual (napaka parametrov) podana kot

$$\hat{\mathbf{A}} = (\mathbf{X}^{IT} \mathbf{W} \mathbf{X}^I)^{-1}. \quad (4.126)$$

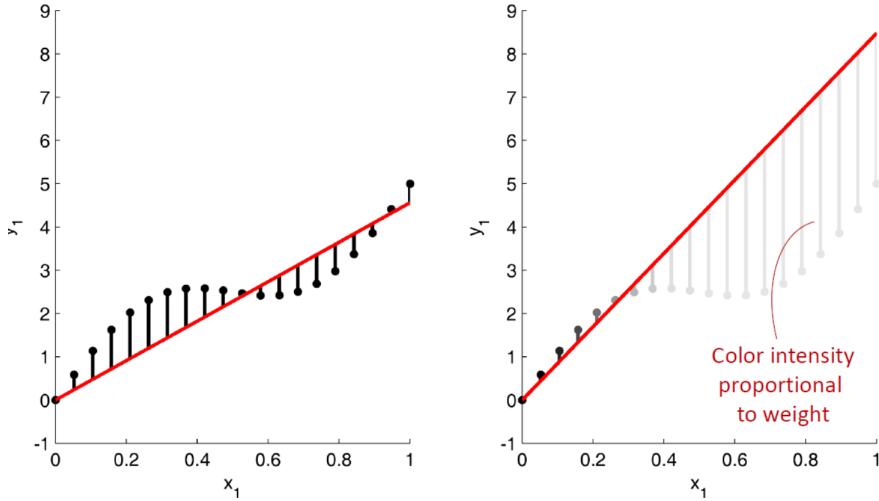
Alternativna rešitev z metodo najmanjših kvadratov je podana v obliki

$$\hat{\mathbf{A}} = \tilde{\mathbf{W}} \mathbf{X}^{IT} (\mathbf{X}^I \tilde{\mathbf{W}} \mathbf{X}^{IT})^{-1} \mathbf{X}^O. \quad (4.127)$$

Zgornji pristopi so prisotni pri učenju in vodenju mnogo robotskih, posebej humanoidnih sistemih.

#### 4.2.8.2 Lokalno utežena regresija

Lokalno utežena regresija (Locally Weighted Regression, LWR) je direktna razširitev prej omenjene utežene oblike najmanjših kvadratov, kjer se izvede  $K$  uteženih regresij nad isto množico podatkov  $\{\mathbf{X}^I, \mathbf{X}^O\}$ . Pravzaprav rešuje nelinearni problem z uporabo več linearnih regresij lokalno. LWR izračuna  $K$  ocen za  $\hat{\mathbf{A}}_k$ , vsako z drugačno funkcijo uteži  $\phi_k(\mathbf{x}_n^I)$ , ki jo pogosto definiramo kot radialno bazno funkcijo (Radial Basis Function, RBF)

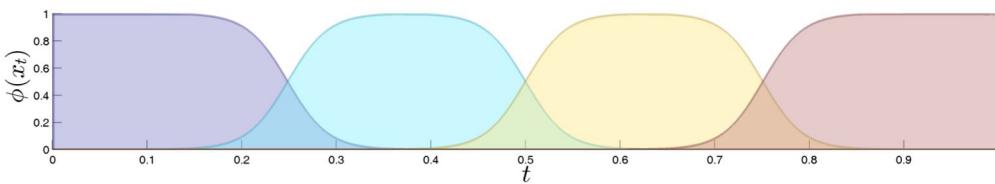


Slika 4.30: Običajna metoda najmanjih kvadratov in utežena metoda najmanjih kvadratov

$$\tilde{\phi}_k(\mathbf{x}_n^I) = \exp\left(-\frac{1}{2}(\mathbf{x}_n^I - \mu_k^I)^T \sum_{k=1}^{I-1} (\mathbf{x}_n^I - \mu_k^I)\right), \quad (4.128)$$

ali v njeni spremenjeni obliki kot

$$\phi_k(\mathbf{x}_n^I) = \frac{\tilde{\phi}_k(\mathbf{x}_n^I)}{\sum_{i=1}^K \tilde{\phi}_k(\mathbf{x}_n^I)}. \quad (4.129)$$



Slika 4.31: Vrednosti  $\phi_k(\mathbf{x}_n^I)$

Ta po velikosti spremenjena oblika je pogosto potrebna pri nekaterih uporabah, vendar to ni nujno pri LWR.  $\mu_k^I$  in  $\Sigma_k^I$  so parametri k-te RBF. Pripravljajoča diagonalna matrika

$$\mathbf{W}_k = \text{diag}(\phi_k(\mathbf{x}_1^I), \phi_k(\mathbf{x}_2^I), \dots, \phi_k(\mathbf{x}_N^I)) \quad (4.130)$$

je potem lahko uporabljena za izračun  $\hat{\mathbf{A}}_k$ .

$$\hat{\mathbf{A}}_k = (\mathbf{X}^{IT} \mathbf{W}_k \mathbf{X}^I)^{-1} \mathbf{X}^{IT} \mathbf{W}_k \mathbf{X}^O. \quad (4.131)$$

Končno je mogoče zapisati rezultat v obliki

$$\mathbf{X}^O = \sum_{k=1}^K \mathbf{W}_k \mathbf{X}^I \hat{\mathbf{A}}_k. \quad (4.132)$$

Pogosto so centroide  $\mu_k^I$  določene primerno za pokrivanje celotnega vhoodnega območja, medtem ko je  $\Sigma_k^I = \mathbf{I}\sigma^2$  uporabljeni kot območje skupno vsem baznim funkcijam.

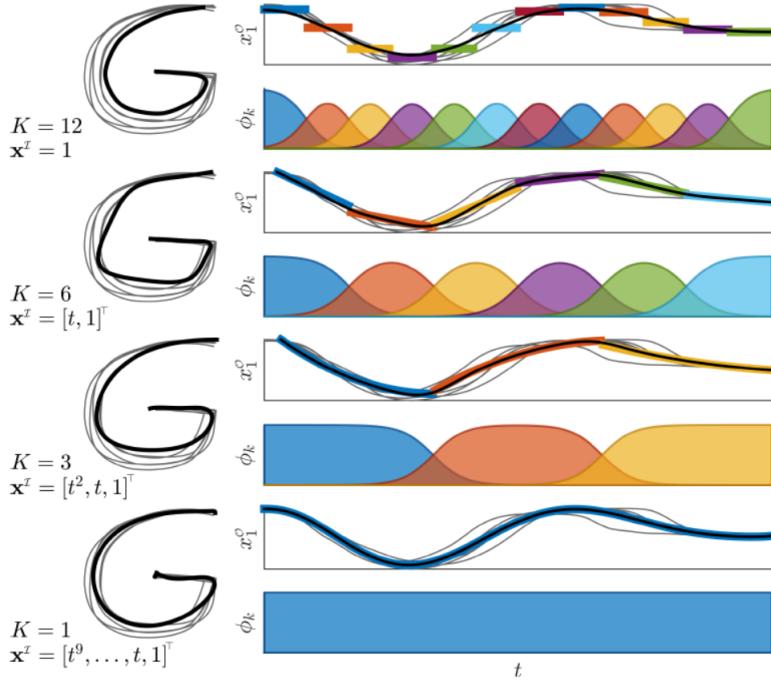
LWR je mogoče neposredno razširiti na metodo lokalnih najmanjših kvadratov ter prilagajanjem polinomov. Sledenča slika kaže primere LWR z različnimi bazičnimi funkcijami in številom komponent. Obstajajo različni zapisi, kot tudi izvedbe, vključno z rekurzivnim pristopom, Bayes obliko LWR, ali razširitve v obliki lokalno utežene regresije s projekcijo (Locally Weighted Projection Regression, LWPR). LWPR izkorišča parcialne odseke z najmanjšimi kvadrati, kar vpliva na redundantne ali irrelevantne vhode ob sprotni oceni vrednosti parametrov modela, na inkrementalnen način brez zadrževanja velike količine podatkov v pomnilniku.

Aplikacije v robotiki so različne, od humanoidnih robotov s številnimi prostostnimi stopnjami ter zapisom inverzne dinamike za celo telo, do kompleksne dvoročne manipulacije.

#### 4.2.9 Gaussova mešana regresija (GMR)

Gaussova mešana regresija (Gaussian Mixture Regression, GMR) je naslednja popularna metoda, ki jo je mogoče uporabiti samostojno ali v povezavi z dinamičnimi primitivi gibanja (Dynamic Movement Primitives, DMP). Uporablja linearne transformacije in lastnosti večdimenzionalnih normalnih porazdelitev. GMR predstavlja način za izračun izhodnih porazdelitev pri sprotnem izračunavanju, pri tem pa je računski čas neodvisen od števila podatkovnih točk uporabljenih za učenje modela. Značilnost GMR je, da ne modelira direktno regresijske funkcije. Namesto tega najprej oblikuje skupno gostoto verjetnosti podatkov v obliki Gaussovega mešanega modela (Gaussian mixture model, GMM), ki je npr. lahko ocenjen s postopkom maksimiranja pričakovanj (Expectation-maximization, EM). Zatem lahko iz naučenega modela gostote z le malo računanja določi regresijsko funkcijo.

V GMR so lahko vhodne in izhodne spremenljivke večdimenzionalne. Mogoče je uporabiti katerokoli kombinacijo vhodno-izhodnih dimenzij podatkov, celo se lahko v primeru potrebe spremenijo v vsaki iteraciji. Pri robotih je to mogoče izkoristiti za obdelavo različnih virov z manjkajočimi podatki, saj je



Slika 4.32: Prileganje polinomov z LWR, z upoštevanjem različnih stopenj polinomov in uporabo različnega števila baznih funkcij. Najbolj zgornja krvulja na sliki prikazuje potek gibanja s preprostimi vzorci, to posledično zahteva veliko baznih funkcij. Najbolj spodnja vrstica kaže nasprotno eks tremno situacijo z globalnim prilagajanjem polinoma na isti potek gibanja, potreben je polinom višjega reda.

med reprodukcijo mogoče upoštevati katerokoli kombinacijo vhodno-izhodnih preslikav, pri tem pa preostale dimenzijske mogoče izračunati iz multidimenzionalne distribucije.

V nadaljevanju bomo označili podatkovne točke v trenutku  $\mathbf{x}_t$  in z centrom  $\mu_i$  ter kovarianco  $\Sigma_i$  kot

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_t^I \\ \mathbf{x}_t^O \end{bmatrix}, \quad \mu_i = \begin{bmatrix} \mu_i^I \\ \mu_i^O \end{bmatrix}, \quad \Sigma_i = \begin{bmatrix} \Sigma_i^I \Sigma_i^{IO} \\ \Sigma_i^{OI} \Sigma_i^O \end{bmatrix}. \quad (4.133)$$

Z namenom povezave s prej predstavljenimi metodami, bo najprej obravnavan primer iskanja časovne trajektorije. V vsakem iteracijskem koraku  $t$  med iskanjem, je  $\mathcal{P}(\mathbf{x}_t^O \mid \mathbf{x}_t^I)$  lahko izračunana kot multidimenzionalna pogojna distribucija

$$\mathcal{P}(\mathbf{x}_t^O \mid \mathbf{x}_t^I) = \sum_{i=1}^K h_i(\mathbf{x}_t^I) \mathcal{N}(\hat{\mu}_i^O(\mathbf{x}_t^I), \hat{\Sigma}_i^O), \quad (4.134)$$

z

$$\begin{aligned} \hat{\mu}_i^O(\mathbf{x}_t^I) &= \mu_i^O + \Sigma_i^{IO} \Sigma_i^{I^{-1}} (\mathbf{x}_t^I - \mu_i^I) \\ \hat{\Sigma}_i^O &= \Sigma_i^O - \Sigma_i^{IO} \Sigma_i^{I^{-1}} \Sigma_i^{OI} \end{aligned} \quad (4.135)$$

in

$$h_i(\mathbf{x}_t^I) = \frac{\pi_i \mathcal{N}(\mathbf{x}_t^I \mid \mu_i^I, \Sigma_i^I)}{\sum_k^K \pi_k \mathcal{N}(\mathbf{x}_t^I \mid \mu_k^I, \Sigma_k^I)}, \quad (4.136)$$

izračunanim s pomočjo

$$\mathcal{N}(\mathbf{x}_t^I \mid \mu_i^I, \Sigma_i^I) = (2\pi_i)^{-\frac{D}{2}} |\Sigma_i^I|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}_t^I - \mu_i^I)^T \Sigma_i^{I^{-1}} (\mathbf{x}_t^I - \mu_i^I)). \quad (4.137)$$

Ko je zahtevana samo ena izhodna porazdelitev, se pravkar zapisani izrazi poenostavijo v Gaussovo distribucijo

$$\mathcal{P}(\mathbf{x}_t^O \mid \mathbf{x}_t^I) = \mathcal{N}(\mathbf{x}_t^O \mid \hat{\mu}_t^O, \hat{\Sigma}_t^O), \quad (4.138)$$

z

$$\hat{\mu}_t^O = \sum_{i=1}^K h_i(\mathbf{x}_t^I) \hat{\mu}_i^O(\mathbf{x}_t^I), \quad (4.139)$$

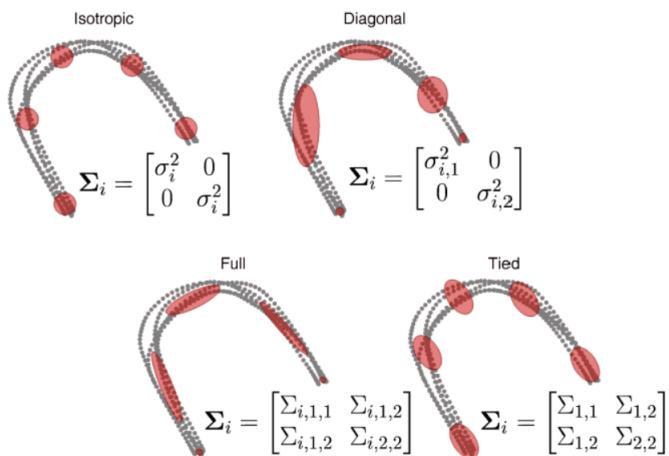
$$\hat{\Sigma}_t^O = \sum_{i=1}^K h_i(\mathbf{x}_t^I) (\hat{\Sigma}_i^O + \hat{\mu}_i^O(\mathbf{x}_t^I) \hat{\mu}_i^O(\mathbf{x}_t^I))^T - \hat{\mu}_t^O \hat{\mu}_t^{OT}. \quad (4.140)$$

Spodnja slika prikazuje več tipov kovariančnih omejitev, ki so tipično uporabljeni v GMM. Še naslednja slika prikazuje GMR proces z 1D vhodom in 1D izhodom. Še nadaljnja slika prikazuje primer DMP izračuna preko GMR. Pri zapisu s pomočjo GMR reprezentacije, standardni DMP ustrezajo GMM z diagonalnimi kovariancami. Z formulacijo DMP kot GMR, je DMP razširjen tako, da

1. dopušča zapis lokalnih korelacij med samimi spremenljivkami gibanja s pomočjo razširitve diagonalnih kovarianc v polne kovariance;
2. zagotavlja načelni pristop k ocenitvi parametrov v RBF, to je podobno GMM modelu naleganja;

3. omogoča znatno zmanjšanje števila zahtevanih RBF, saj je položaj in širino vsakega RBF mogoče samodejno prilagoditi iz demonstracij; in
4. sprotna ocena DMP parametrov in način izbire modela (samodejno ocena števila RBF) lahko zlahka izkorišča tehnike združljive z GMM (Bayesian nonparametrics with Dirichlet processes, spectral clustering, small variance asymptotics, etc.).

GMR je bil uporabljen pri robotih za učenje različnih nalog, vključno s premikanjem predmetov, točenju pijač v kozarec pri humanoidnih robotih, korekciji dotika v primeru kretenj rok, kuhanju riža ali razvaljanju testa za pico.

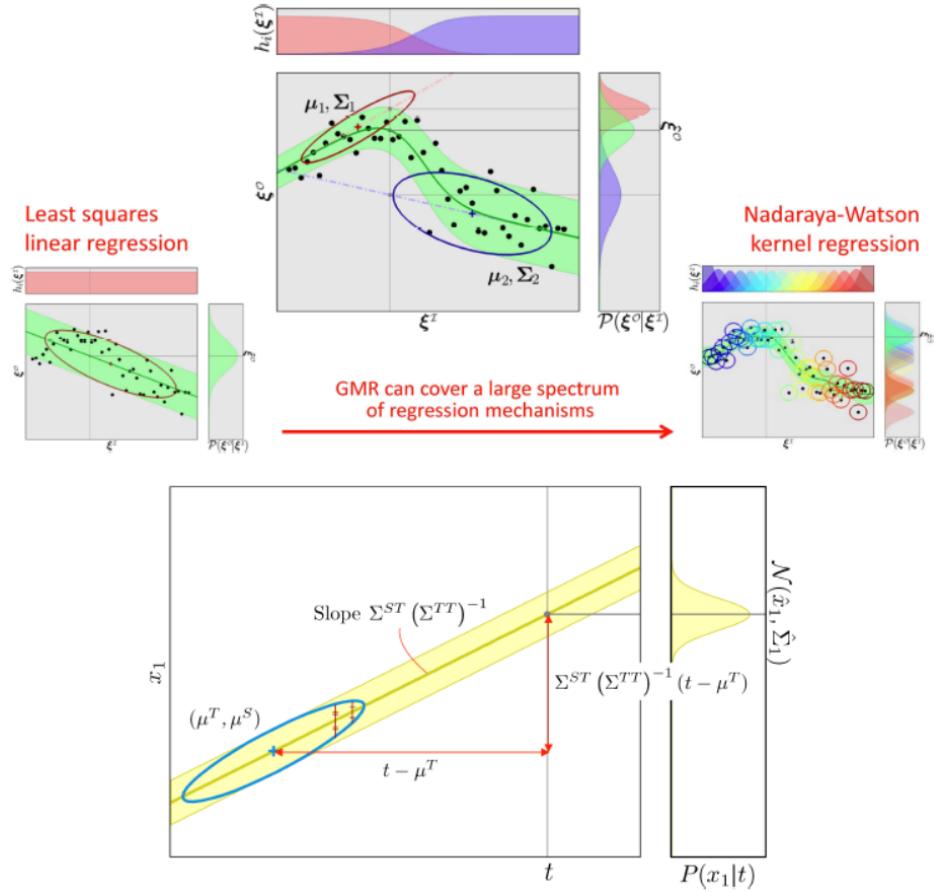


Slika 4.33: Različne varijante kovariančnih matrik v Gaussovih mešanih modelih (GMM)

#### 4.2.10 Gaussian Process Regression (GPR)

Želimo poiskati rešitev regresijskega problema  $\mathbf{x}^O = f(\mathbf{x}^I) + \eta$ , kjer je  $f$  neznana funkcija, medtem ko je  $\eta$  dodani šum procesa. Če predpostavimo, da imamo poln komplet vhodno-izhodnih spremenljivk  $\{\mathbf{x}_t^I, \mathbf{x}_t^O\}_{t=1}^N$ , potem je cilj izračun zapisa funkcije  $f$  ter pripadajoče izhodne porazdelitve  $\mathbf{x}^O$ , pri čemer je podana prej nevidna  $\mathbf{x}^{I*}$ , torej  $\mathbf{x}^{O*} \sim \mathcal{P}(\mathbf{x}^O | \mathbf{x}^{I*})$ .

V prejšnjih podnaslovih je prikazano polinomsko prileganje, pri tem smo podali stopnjo polinoma. Veliko je scenarijev, kjer imamo malo ali nič predznanja o morebitnem primernem modelu, vendar imamo morda nekaj specifičnega znanja, ki bi ga radi izrazili v bolj priročni obliki. Na primer, lahko vemo, da opazovani vzorci prihajajo iz zveznega procesa, ali da spremembe

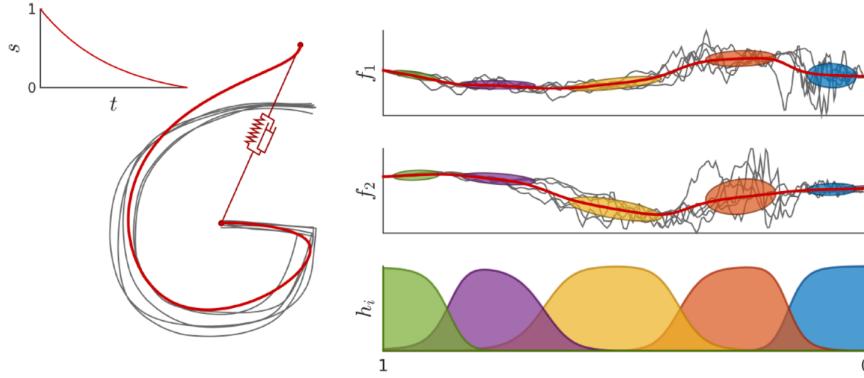


Slika 4.34: Gaussova mešana regresija (GMR) ima preprosto formulacijo, ki lahko pokrije široko paletu regresijskih metod, od multivariabilne linearne regresije (enjni Gaussian) do podatkovno osnovanih regresij v jedru (Gaussian centred on each datapoint).

v funkciji pripadajo nekemu časovnemu intervalu. Gassovi procesi so lahko uporabljeni kot pristop, ki odraža razne oblike prejšnjega znanja o opazovanem fizičnem procesu.

GPR temelji na dejstvu, da je vsaka opazovana točka v naboru podatkov lahko predstavljena kot podatkovna točka v multivariabilni Gaussovi distribuciji. Neskončna množica vseh možnih spremeljivk potem predstavlja ekvivalent distribucije celega funkcijskoga prostora. Hiperparametre v modelu je še vedno potrebno določiti, vendar hiperparametri popisujejo bolj splošne karakteristike, kot je velikost distribucije ter ne vplivajo na strukturo ali funkcijsko obliko signalov.

Kovarianca se nahaja v samem jedru Gaussovih procesov. Definirana je



Slika 4.35: DMP z GMM z zapisom sklepnih spremeljivk v obliki vplivnih členov  $f(s)$  in fazne spremeljivke  $s$ .

skozi uporabo jedrni funkcijs  $k(\mathbf{x}_i^I \mathbf{x}_j^I)$ , te opisujejo kovariančne elemente med dvema vzorcema  $\mathbf{x}_i^I$  in  $\mathbf{x}_j^I$ . Za množico vhodov  $\mathbf{x}^I = \{\mathbf{x}_1^I, \mathbf{x}_2^I, \dots, \mathbf{x}_N^I\}$  je kovariančna matrika (znana tudi kot Gram-ova matrika) zatem definirana kot

$$\mathbf{K}(\mathbf{x}^I, \mathbf{x}^I) = \begin{bmatrix} k(\mathbf{x}_1^I, \mathbf{x}_1^I) & k(\mathbf{x}_1^I, \mathbf{x}_2^I) & \dots & k(\mathbf{x}_1^I, \mathbf{x}_N^I) \\ k(\mathbf{x}_2^I, \mathbf{x}_1^I) & k(\mathbf{x}_2^I, \mathbf{x}_2^I) & \dots & k(\mathbf{x}_2^I, \mathbf{x}_N^I) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N^I, \mathbf{x}_1^I) & k(\mathbf{x}_N^I, \mathbf{x}_2^I) & \dots & k(\mathbf{x}_N^I, \mathbf{x}_N^I) \end{bmatrix}. \quad (4.141)$$

To pomeni, da izračun celotne funkcije  $f(\mathbf{x}^I)$  povezane s skupino vhodov  $\mathbf{x}^I$  izhaja iz multivariantne Gaussove distribucije  $\mathbf{x}^O \sim \mathcal{N}(\mu(\mathbf{x}^I), \mathbf{K}(\mathbf{x}^I, \mathbf{x}^I))$ . Torej Gassov proces definira distribucijo funkcij.

Poleg tega lahko dodatno upoštevamo tudi šum prisoten v vrednostih  $\mathbf{x}_t^O$ . Vzorce pogosto privzamemo kot neodvisne in enakomerno porazdeljene (independent and identically distributed, IID), kar pomeni samo prištevanje v diagonali  $K$ . Tako je spremenjena kovarianca za pošumljene opazovane vzorce v obliki

$$\tilde{\mathbf{K}}(\mathbf{x}^I, \mathbf{x}^I) = \mathbf{K}(\mathbf{x}^I, \mathbf{x}^I) + \Theta^{GP} \mathbf{I}, \quad (4.142)$$

kjer je  $\mathbf{I}$  enotina matrika in  $\Theta^{GP}$  varianca šuma kot eden od hiperparametrov Gassovega procesa.

Pri računanju regresije smo zainteresirani za kasnejšo distribucijo  $\mathbf{x}^{O*}$ , torej na izhodu, pri nekih podatkovnih točkah  $\mathbf{x}^{I*}$ . Skupna porazdelitev obravnavanega para  $\mathbf{x}^I$  in  $\mathbf{x}^O$ , razširjenima v  $\mathbf{x}^{I*}$  in  $\mathbf{x}^{O*}$  je

$$\begin{bmatrix} \mathbf{x}^O \\ \mathbf{x}^{O*} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\mathbf{x}^I) \\ \mu(\mathbf{x}^{I*}) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{x}^I, \mathbf{x}^I) & \mathbf{K}(\mathbf{x}^I, \mathbf{x}^{I*}) \\ \mathbf{K}(\mathbf{x}^{I*}, \mathbf{x}^I) & \mathbf{K}(\mathbf{x}^{I*}, \mathbf{x}^{I*}) \end{bmatrix}\right). \quad (4.143)$$

Zatem lahko uporabimo lastnost pogojne verjetnosti Gaussovih distribucij pri izračunu kasnejše porazdelitve preko  $\mathbf{x}^{O*}$

$$\mathbf{x}^{O*} \mid \mathbf{x}^O \sim \mathcal{N}(\mu^*, \Sigma^*) \quad (4.144)$$

s srednjo vrednostjo in kovarianco

$$\begin{aligned} \mu^* &= \mu(\mathbf{x}^{I*}) + \mathbf{K}(\mathbf{x}^{I*}, \mathbf{x}^I) \mathbf{K}(\mathbf{x}^I, \mathbf{x}^I)^{-1} (\mathbf{x}^O - \mu(\mathbf{x}^I)), \\ \Sigma^* &= \mathbf{K}(\mathbf{x}^{I*}), \mathbf{x}^{I*} - \mathbf{K}(\mathbf{x}^{I*}, \mathbf{x}^I) \mathbf{K}(\mathbf{x}^I, \mathbf{x}^I)^{-1} \mathbf{K}(\mathbf{x}^I, \mathbf{x}^{I*}). \end{aligned} \quad (4.145)$$

V zgornjem zapisu je  $\mathbf{K}(\mathbf{x}^I, \mathbf{x}^I)$  mogoče nadomestiti s  $\mathbf{K}(\mathbf{x}^I, \mathbf{x}^I)$ , če je pri spremenljivki  $\mathbf{x}^O$  prisoten tudi šum. V praksi je pogosto privzeta vrednost  $\begin{bmatrix} \mu(\mathbf{x}^I) \\ \mu(\mathbf{x}^{I*}) \end{bmatrix} = 0$ . Gaussov proces je torej lahko popolnoma definiran z njegovo statistiko drugega reda, pri čemer je Gram-ova matrika  $K$  pozitivno semidefinitna z kovariancami, ki so posledica skalarnih produktov vektorjev.

Jedrne funkcije ponazarjajo lastnost podobnosti na način, da bosta dve podobni točki  $\mathbf{x}_i^I$  in  $\mathbf{x}_j^I$  imeli odgovarjajoči vrednosti  $\mathbf{x}_i^O$  in  $\mathbf{x}_j^O$  in bolj močno korelirani kot različni pari točk. Podobnost je seveda odvisna od same aplikacije. S pomočjo lastnosti kovariančne funkcije  $\mathbf{k}(\mathbf{x}^I, \mathbf{x}^I)$  je mogoče vpeljati stacionarnost, izotropnost, gladkost in periodičnost procesa.

Pri opazovanju zveznih časovnih spremenljivk je mogoče navadno upoštevati pretekla opažanja kot infomativna za trenutne podatke in to kot funkcijo, kako dolgo nazaj so bila opažena. To lahko ustreza stacionarni kovarianci, ta je odvisna od Evklidske razdalje  $\|\mathbf{x}_i^O - \mathbf{x}_j^O\|_2$ . Proses obravnavamo potem kot vsesmeren in ni odvisen od smeri  $\mathbf{x}_i^I$  in  $\mathbf{x}_j^I$ . Stacionaren in hkrati izotropen proces je poimenovan kot homogen. Pri vodenju robotov je najpogosteje uporabljena takšna funkcija radialna bazna funkcija (radial basis function, RBF). RBF je pogosto uporabljena, kadar od dveh bližnjih vhodnih vrednosti  $\mathbf{x}_i^I$  in  $\mathbf{x}_j^I$  pričakujemo bližnje izhodne vrednosti  $\mathbf{x}_i^O$  in  $\mathbf{x}_j^O$  (torej je privzeta zveznost). Kadar so predpostavljene šumne vrednosti, je jedrno funkcijo mogoče definirati kot

$$\mathbf{k}(\mathbf{x}_i^I, \mathbf{x}_j^I) = \Theta_1^{GP} \exp\left(-\frac{1}{\Theta_2^{GP}} (\mathbf{x}_i^I - \mathbf{x}_j^I)^T (\mathbf{x}_i^I - \mathbf{x}_j^I)\right) + \Theta_3^{GP} \delta_{i,j}, \quad (4.146)$$

Kjer je  $\delta_{i,j} = \mathbf{I}(i = j)$  enako ena samo pri  $i = j$ , drugače pa je nič. Vrednosti tvorijo  $\mathbf{K}(\mathbf{x}^I, \mathbf{x}^I)$ , kjer je šum povezan z opaženimi vrednostim samo v diagonali (šum ne korelira med vzorci). Ojačanje je  $\Theta_3^{GP}$ . Preostala dva hiperparametra  $\Theta_1^{GP}$  in  $\Theta_2^{GP}$  sta povezana s skaliranjem vhoda in izhoda. Periodične jedrne funkcije so naslednja pomembna družina za uporabo v

GPR, pri učenju humanoidov, vključno s periodičnimi vzorci gibanja. Mogoče je definirati tudi bolj kompleksne kovariančne funkcije, kot linearne kombinacije preprostejših funkcij. Naslednji učinkovit način zapisa jedrnih funkcij je z uporabo modelov verjetnosti. Z upoštevanjem splošnega zapisa  $\mathcal{P}(\mathbf{x}^I)$ , je jedro mogoče zapisati kot  $\mathbf{k}(\mathbf{x}_i^I, \mathbf{x}_j^I) = \mathcal{P}(\mathbf{x}_i^I)\mathcal{P}(\mathbf{x}_j^I)$ , kar lahko gledamo kot notranji produkt v enodimensijskem prostoru možnosti deifniranih kot relacija  $\mathcal{P}(\mathbf{x}^I)$ . Namreč, dva vhoda  $\mathbf{x}_i^I$  in  $\mathbf{x}_j^I$  bosta podobna, če imata oba veliko verjetnost. Ta pristop omogoča uporabo generativnih modelov za razločevanje vrednosti. To lahko prinese nove lastnosti modeliranemu procesu, kot npr. sposobnost uporabe podatkov z manjkajočimi vrednostmi, ali delne odseke razuličnih dolžin (npr. pri HMM).

Zgornja slika kaže primera dveh različnih jeder, RBF in periodičnega. Sledenja slika pa ilustrira razliko med GPR in GMR. Pri uporabi RBF jedra, GPR obnovi gladko trajektorijo, pri čemer je manjkajoči del zapolnjen z upoštevanjem prejšnjega povprečja vrednosti. Obnovljena kovarianca predstavlja zaupanje v oceno iz povprečne trajektorije, ki je tukaj zelo majhno za konstantno za dele trajektorij z dostopnimi podatkovnimi točkami in potem naraste za manjkajoče odseke podatkov. Obnovljena je vsaka izhodna spremenljivka posebej, kar vodi do diagonalne kovariance v  $(\mathbf{x}_1 - \mathbf{x}_2)$  grafu. V primeru GMR so podatkovne točke najprej uporabijo za porazdelitve v sklepu v obliki GMM (tukaj je s štirimi Gaussovimi modeli pokazana skupna porazdelitev  $(t - \mathbf{x}_1 - \mathbf{x}_2)$ ). Nabor podatkov se nato zavrže. GMR je sestavljen iz izračuna utežene vsote pogojnih porazdelitev in ponazoritev z Gaussom. Manjkajoči del podatkov se obravnava z gladkim prehodom med obema linearnima trendoma na obe straneh manjkajočih podatkov. V nasprotju z GPR, variance katerega predstavljajo zaupanje v oceno povprečne vrednosti, obnovljene variance v GMR predstavljajo spremembe in korelacije opažene v podatkih, s polnimi kovariancami, obnovljenimi za izhodne porazdelitve. To nakazuje lokalne koordinacijske vzorce (zeleni elipsoidi na  $(\mathbf{x}_1 - \mathbf{x}_2)$  grafu). Podobno kot pri oceni povprečne vrednosti je obravnavan manjkajoči del podatkov z gladko interpolacijo med opaženima kovariancama na obe straneh. Slika tudi prikazuje, da je mogoče glede na uporabljen kriterij lahko prevladuje eden ali drug pristop. Najprej je treba opozoriti, da GMR je tesno povezan z multivariatnimi normalnimi lastnostmi, medtem ko GPR ni omejen na uporabo Gaussovih jeder in je zato bolj splošen. GMR je konkurenčen pri hitrem računanju, kadar se vhodne in izhodne komponente spremenijo (npr. ocena iz delno znanega vhoda), ali ko je potrebna ocena variance. To zahteva oceno GSM, skupaj s številnimi Gaussovimi funkcijami uporabljenimi v modelu. Kadar so pomembne informacije o koordinaciji in spremembah, je mogoče uporabiti tako imenovani posplošen Wishartov postopek za razvoj kovarianc namesto Gaussovih procesov.

Različne variante GPR so že biule uporabljene za učenje robotov. GPR se lahko uporablja za slednje gibanja ali seganje z robotom, kjer je množica zunanjih parametrov povezana z DMP parametri za kodiranje gibanje. Mogoče je tudi uporabiti Gaussove procese za sprotno izbiro točk učenja pri učenju modela inverzne dinamike. GPR je tudi uporabljen pri 2D bipedalni hodji humanoidnega robota, pri čemer je posplošena neoptimalna trajektorija v sklepih.

## 4.3 DINAMIČNI PRIMITIVI GIBANJA

### 4.3.1 Dinamični primitivi gibanja (DMP)

Dinamični primitivi gibanja (Dynamic movement primitives, DMP) so metoda nadzora, načrtovanja poti iz laboratorija Stefana Schaalja. Predstavljeni so bili že leta 2002, nato pa jih je leta 2013 Auke Ijspeert posodobil. Delo je bilo motivirano z željo, da bi našli način za predstavitev zapletenih gibalnih situacij, ki jih je mogoče fleksibilno prilagoditi brez ročnega nastavljanja parametrov ali skrbi za nestabilnost. Dolgo že velja, da so zapleteni gibi sestavljeni iz nizov preprostih akcijskih elementov, ki se izvajajo zaporedoma in/ali vzporedno, DMP so predlagana matematična formalizacija teh primitivov. Razlika med DMP-ji in predhodno predlaganimi gradniki je v tem, da je vsak DMP nelinearni dinamični sistem. Osnovna ideja je, da vzamemo dinamičen sistem z natančno določenim, stabilnim odzivom, nato pa dodamo člen preko katerega je mogoče sledenje trajektoriji. Obstajata dve vrsti DMP: diskretni in ritmični (periodični). Za diskretna gibanja iz točke v točko predstavlja osnovni sistem točkovni atraktor, za ritmična gibanja pa se uporablja mejni cikel. DMP zagotavljajo kompleksen okvir za specifikacijo trajektorij robota. Temeljijo na nelinearnih diferencialnih enačbah in vodijo do zveznih načinov gibanja, kar je bistvenega pomena za izvajanje robustnega gibanja robota. Dobre lastnosti DMP so:

1. Zagotavljajo gladke prehode v primeru nenadnih sprememb pri opisu gibanja ali pri motnji.
2. Niso eksplisitno odvisni od časa.
3. Zagotavljajo ustrezni okvir za učenje in prilaganje poti s posnemanjem in algoritmi za spodbujevano učenja.
4. Omogočajo učenje celotnih družin trajektorij iz več demonstracij z uporabo statističnih algoritmov učenja.

5. Mogoča je uporaba pri več prostorstnih stopnjah.
6. Mogoče je izvajanje v realnem času, kot tudi poljubno spremiščanje trajektorij, morda na osnovi senzornih signalov in tudi spremiščanje parametrov za nastavljanje.
7. Tako imenovani mehanizem za zaustavitev faze omogoča robotu zaustavitev napredovanje faze, prav tako omogoča sledenje želeni poti tudi v primeru velike napake v trajektoriji.

Takšne lastnosti je mnogo težje izvesti z časovno odvisnimi zapisi, npr. časovno odvisnimi polinomi ali spline. Predstavljajte si, da imamo dva sistema: namišljeni sistem, kjer načrtujemo trajektorije, in pravi sistem, kjer jih izvajamo. Pri DMP pravzaprav načrtujemo trajektorije, ki jim bo sledil naš pravi sistem. DMP ima svojo lastno dinamiko, torej lahko s pravilno nastavljivo DMP generiramo krmilni signal za dejanski sistem. Če DMP tako definira referenco trajektorije za robota, potem v resnični sistem pošljemo nabor sil v zunanjih koordinatah potrebnih za gibanje roke. Resnični sistem mora zatem uporabiti te referenčne vrednosti, jih pretvoriti v navore v sklepah ali mišične sile, ali ali kar koli drugega. DMP torej generira referenco za vodenje resničnega sistema.

Dinamiko nekega sistema lahko zapišemo na različne, vendar v osnovi iste načine, npr. z enačbo 2. reda

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}), \quad (4.147)$$

pri čemer je  $y$  neodvisna spremenljivka. Pri robotu je to ena prostostna stopnja v zunanjih ali notranjih koordinatah,  $g$  je želena referenčna vrednost, medtem ko sta  $\alpha$  in  $\beta$  ojačanji.

Samo nekoliko drugačen zapis iste formulacije bi bil v obliki

$$\begin{aligned} \tau \dot{z} &= \alpha_z(\beta_z(g - z) - z), \\ \tau \dot{y} &= z, \end{aligned} \quad (4.148)$$

kjer je  $z$  samo s  $\tau$  skalirana osnovna spremenljivka  $y$ . Pomensko bi  $\alpha$  in  $\beta$  gledali kot dušenje in togost vzmeti. Pri pravi kombinaciji parametrov, predstavljajo te enačbe globalno stabilen dinamični sistem s stacionarno, atraktor točko  $g$ . To pomeni, da se bo za vsako začetno točko  $y = y_0$  parameter  $y$  izenačil z  $g$  po koncu prehodnega pojava, podobno kot stisnjena vzmet, ki se bo po sprostitvi vrnila v prvotno stanje.

V nadaljevanju zapišimo splošno rešitev za linearni sistem zgornjih nehomogenih diferencialnih enačb. V splošnem je mogoče rešitev takšnega sistema zapisati kot vsoto homogene in parcialne rešitve, torej  $[z(t), y(t)]^T =$

$[z_p(t), y_p(t)]^T + [z_h(t), y_h(t)]^T$ . Tukaj predstavlja  $[z_p(t), y_p(t)]^T$  vsako funkcijo, ki rešuje zgornji sistem obeh diferencialnih enačb, medtem ko je  $[z_h(t), y_h(t)]^T$  splošna rešitev homogenega dela teh dveh diferencialnih enačb, torej zapisa

$$\begin{bmatrix} \dot{z} \\ \dot{y} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -\alpha_z(\beta_z y + z) \\ z \end{bmatrix} = \mathbf{A} \begin{bmatrix} z \\ y \end{bmatrix}, \quad \mathbf{A} = \frac{1}{\tau} \begin{bmatrix} -\alpha_z & -\alpha_z \beta_z \\ 1 & 0 \end{bmatrix}. \quad (4.149)$$

Preveriti je mogoče, da konstantna funkcija  $[z_p(t), y_p(t)]^T = [0, g]^T$  rešuje sistem obeh diferencialnih enačab. Poleg tega tudi poznamo splošno obliko rešitve za homogeni del zapisan v matrični obliki zgoraj, ima obliko  $[z_h(t), y_h(t)]^T = \exp(\mathbf{At})\mathbf{c}$ . Torej je mogoče splošno rešitev zapisati v obliki

$$\begin{bmatrix} z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 0 \\ g \end{bmatrix} + \exp(\mathbf{At})\mathbf{c} \quad (4.150)$$

kjer je mogoče  $\mathbf{c} \in \mathbb{R}^2$  izračunati iz začetnih vrednosti  $[z(0), y(0)]^T = [z_o, y_0]^T$ . Lastne vrednosti  $\mathbf{A}$  so podane z  $\lambda_{1,2} = (-\alpha_z \pm \sqrt{\alpha_z^2 - 4\alpha_z\beta_z})/(2\tau)$ . Stacionarna rešitev za  $z$  in  $y$  konvergira proti  $[0, g]^T$ , če je realni del  $\lambda_{1,2}$  manjši od nič, kar velja za vsak  $\alpha_z, \beta_z, \tau > 0$ . Sistem je kritično dušen, to pomeni konvergenco  $y$  proti  $g$  brez oscilacij in hitreje kot katerakoli druga izbira za  $\mathbf{A}$ , če ima  $\mathbf{A}$  dve negativni lastni vrednosti. To se zgodi pri  $\alpha_z = 4\beta_z$ .

#### 4.3.1.1 Diskretni gibi iz točke v točko

Prvotni diferencialni enačbi zagotavljata konvergenco  $y$  proti  $g$  in sta torej lahko uporabljeni za diskrette gibe iz točke v točko. Da bi pa povečali število možnih trajektorij, ki jih tak zapis omogoča, je mogoče priključiti istim enačbam dodatno nelinearno funkcijo  $f$

$$\begin{aligned} \tau \dot{z} &= \alpha_z(\beta_z(g - y) - z) + f(x), \\ \tau \dot{y} &= z \end{aligned} \quad (4.151)$$

oz. v zapisu z enačbo drugega reda

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f \quad (\tau = 1). \quad (4.152)$$

Kako definirati nelinearno funkcijo  $f$ , da bomo dobili želene odzive, ni trivialno vprašanje. Bistvo DMP zapisov je dodatna nelinearna funkcija  $f$ , ta definira dobro definirano strukturo, ki je preprosto rešljiva in veljavna za veliko število primerov. Ena od možnosti je uporaba linearne kombinacije radialnih baznih funkcij

$$f(x) = \frac{\sum_{i=1}^N \omega_i \Psi_i(x)}{\sum_{i=1}^N \Psi_i(x)} x(g - y_0), \quad \Psi_i(x) = \exp(-h_i(x - c_i)^2), \quad (4.153)$$

kjer so  $c_i$  središča (Gaussovih) radialnih baznih funkcij  $\Psi_i(x)$  razporejenih vzdolž trajektorije,  $y_0$  je začetna vrednost pozicije sistema in varianca  $h_i > 0$ . Komponenta  $g - y_0$ ,  $y_0 = y(0)$  je uporabljena za skaliranje trajektorije, če se začetna in/ali končna konfiguracija spremenita. Vse dokler sta začetek in konec gibanja konstantna ta skalirni faktor nima vpliva in ga je mogoče zanemariti.

Nadalje je upoštevan kanonični sistem, označen kot fazna spremenljivka  $x$ , to uporabimo namesto odvisnosti od časa, z zelo preprosto dinamiko in začetno vrednostjo  $x(0) = 1$ ,  $x$  se približuje 0, medtem ko čas narašča proti neskončnosti

$$\tau \dot{x} = -\alpha_x x. \quad (4.154)$$

Rešitev takšnega sistema je v obliki

$$x(t) = \exp(-\alpha_x t / \tau). \quad (4.155)$$

Zanimiva lastnost uporabe fazne spremenljivke  $x$ , namesto časa  $t$  se odrazi ob primerni modifikaciji diferencialne enačbe za  $x$  tako, da se zaustavi čas z namenom upoštevanja motenj med gibanjem. Pri tem ni potrebno voditi računa o notranji uri sistema.

Fazna spremenljivka  $x$  in posledično  $f(x)$  z naraščanjem časa gravitirata proti nič. Torej vpliv nelinearnega člena  $f(x)$  se s časom zmanjšuje, torej diferencialni enačbi konvergirata proti  $[0, g]^T$ , podobno kot osnovni diferencialni enačbi brez člena  $f(x)$ . Potek trajektorije  $y$ , njenega prvega in drugega odvoda definira mehanizem poimenovan kot Dinamični primitivi gibanja (Dynamic movement primitives, DMP). Pri sistemih z več prostostnimi stopnjami je vsaka predstavljena z neovisnim kompletom osnovnih dveh diferencialnih enačb, pri čemer mora biti fazna spmenljivka ista za vse prostostne stopnje.

Diskretni DMP-ji s svojimi nastavitevami predstavljajo primeren zapis za gibanje iz točke v točko, skupaj z istočasno možnostjo spremicanja različnih lastnosti trajektorij. V tem kontekstu so parametri oblike  $\omega_i$  določeni tako, da lahko robot natančno sledi želeni trajektoriji z integracijo sistema diferencialnih enačb. Drugi parametri so uporabljeni za modulacijo in upoštevanje

motenj. Npr., s spremenjanjem  $g$  lahko prilagodimo končni cilj diskretnega gibanja, medtem ko je  $\tau$  mogoče prilagoditi za spremenjanje njegove hitrosti. Običajno je dovolj določitev parametrov  $c_i$  in  $h_i$  za samo definiranje oblike distribucije, kot tudi števila baznih funkcij  $N$ , s čimer dosežemo želeno točnost trajektorije., Npr. za določeno število  $N$  je mogoče definirati enačbe

$$\begin{aligned} c_i &= \exp(-\alpha_z \frac{i-1}{N-1}), i = 1, \dots, N, \\ h_i &= \frac{2}{(c_{i+1} - c_i)^2}, i = 1, \dots, N-1, h_N = h_{N-1}. \end{aligned} \quad (4.156)$$

Pri tem sta  $c_1 = 1 = x(0)$  in  $c_N = \exp(-\alpha_z) = x(t_T)$ . V primeru predloženosti je mogoče uporabiti tudi boj napredne metode za določitev  $N$ ,  $c_i$  in  $h_i$ . Poskušajmo na grafični način ponazoriti pomen in vpliv posameznih členov. Na začetku predpostavimo, da  $x$  linearno pada proti 0.  $x$  aktivira posamezne bazne funkcije,  $c_i$  je srednja vrednost teh funkcij,  $h_i$  pa njihova varianca. Spodnja slika kaže  $N = 7$  baznih funkcij. Medtem ko se  $x$  zmanjšuje od 1 proti 0 se vključujejo in izključujejo različne bazne funkcije.

Uteži  $\omega_i$  pripadajo posameznim baznim funkcijam, jih poudarijo ali zmanjšajo njihov vpliv. Primer glede na uteži je spet podan na naslednji sliki. Vrednost uteženega seštevka je nato s seštevkom funkcij  $\Psi_i(x)$  v imenovalcu še normirana. Množenje s členom  $x(g - y_0)$  predstavlja preko  $x$  zmanjševanje vrednosti s časom in s tem garantira, da bo katerakoli trajektorija na koncu končala v attractor točki, oz. konvergirala proti želeni končni točki.

V funkciji množenje s členom  $x(g - y_0)$  tudi skalira glede na vrednost  $y_0$ , torej glede na začetno in  $g$  vrednost. Potem ko so vsi parametri v sistemu definirani, je mogoče samo s tem členom prilagoditi delovanje za relativno večjo ali krajsko razdaljo/dolžino trajektorije.

Učinek na spremembo trajektorije si poglejmo na primeru, kjer se z dvema različnima DMP1 in DMP2 gibljemo v dve končni točki 1 in 0.5, potek je prikazan na spodnji sliki.

V primeru ciljne točke postavljene na 2, bi dobili spremenjen potek pri obeh, sledenje želeni trajektoriji bi bilo slabo, skladno z naslednjo sliko. Pri tem skaliranje s členom  $(g - y_0)$  ni bilo uporabljen. Pri teh novih trajektorijah je bila namreč vrednost baznih funkcij prešibka za sledenje referenčni vrednosti. Po vključitvi še člena  $(g - y_0)$ , je potek dejanskih trajektorij ponovno enak želenim referenčnim trajektorijam. To je bil tudi cilj, skaliranje gibanja je sedaj pravilno.

Tukaj obravnavani kanonični sistem ne konvergira linearno proti končni točki, kot smo privzeli zgoraj. Glej diferencialno enačbo za izračun  $x$ . Sledenča slika primerja linearno padanje spremenljivke  $x$  s časom, v primerjavi z eksponentnim zmanjševanjem. To je pomembno, saj so bazne funkcije odvisne in se aktivirajo glede na spremenljivko  $x$ . Ob linearinem zmanjševanju  $x$

bi bile bazne funkcije enakomerno razporejene vzdolž trajektorije. Pri eksponentnem upadu  $x$  in seveda linearinem naraščanju časa  $t$  pa se večino baznih funkcij zgosti takoj na začetku trajektorije, le malo pa se raztezajo proti končni točki trajektorije.

Z namenom enakomerne porazdelitve baznih funkcij glede na čas, je potrebno središča (Gaussovih) radialnih baznih funkcij razporediti primerno glede na čas, kot to npr. kaže naslednja slika. Najbolje je torej izbrati ustrezen čas, glede na tega pa določiti primerno  $x$  vrednost, pripadajočo temu času. Želeli bi si središča Gaussovih funkcij na mestih označenih na sliki.

Dodatno je potrebno pomisliti na varianco vsake funkcije, saj bodo tiste kasnejše funkcije aktivirane dalj časa. To izenačenje dosežemo z zmanjšanjem širine kasnejših baznih funkcij. Eden od možnih pristopov za določitev velikosti variance vsake posamezne bazne funkcije je glede na njeno zaporedno številko.

$$h_i = \frac{\#BFS}{c_i}. \quad (4.157)$$

V tem primeru je varianca odvisna od števila skupnega števila vseh baznih funkcij (v tem primeru označeno z  $\#BFS$ ), deljeno s centrom te bazne funkcije. Z upoštevanjem tega zapisa je mogoče ponovno zapisati bazne funkcije in jih enakomerno narisati na naslednji sliki.

Ponovno velja poudariti splošnost obravnavanega zapisa, tudi v smislu amplitudnega ter časovnega skaliranja. Skaliranje trajektorije po velikosti smo podrobneje že razložili zgoraj. Pri časovnem skaliranju pa bi želeli slediti isti trajektoriji z različnimi hitrostmi, tako počasi kot hitro, vendar vedno po isti poti. S tem namenom dodamo enačbam še en člen, v teh enačbah poimenovan kot  $\tau$ , to je člen za časovno skaliranje. Dosedanjim enačbam

$$\begin{aligned} \ddot{y} &= \alpha_y(\beta_y(g - y) - \dot{y}) + f \\ \dot{x} &= -\alpha_x x, \end{aligned} \quad . \quad (4.158)$$

dodamo še

$$\begin{aligned} \dot{y}^+ &= \tau \ddot{y}, \\ y^+ &= \tau \dot{y}, \\ x^+ &= \tau \dot{x}. \end{aligned} \quad (4.159)$$

Za upočasnitev sistema izberemo vrednost  $\tau$  med 0 in 1, za pohitritev pa vrednost povečamo nad 1.

#### 4.3.1.2 Periodično gibanje

V primeru želenega periodičnega gibanja, je mogoče kot nelianearno funkcijo definirati kombinacijo s periodično funkcijo

$$f(\phi) = \frac{\sum_{i=1}^N \omega_i \Gamma_i(\phi)}{\sum_{i=1}^N \Gamma_i(\phi)} r, \quad \Gamma_i(\phi) = \exp(-h_i(\cos(\phi - c_i))), \quad (4.160)$$

kjer je  $r$  amplituda oscilatorja in  $h_i > 0$ . Z upoštevanjem  $\tau = 1/\Omega$  je originalne diferencialne enačbe mogoče nadomestiti z

$$\begin{aligned} \dot{z} &= \Omega(\alpha_z(\beta_z(g - y) - z) + f(\phi)) \\ \dot{y} &= \Omega z. \end{aligned} \quad (4.161)$$

V tem primeru vpeljemo fazno spremenljivko  $\phi$ , s čimer se izognemo ekplikativni odvisnosti od časa. Predpostavljamo konstantno hitrost faze

$$\dot{\phi} = \Omega, \quad (4.162)$$

tu  $\Omega$  ponazarja frekvenco oscilacij. Normalno so bazne funkcije razporejene enakomerno preko intervala  $[0, 2\pi]$ , torej je

$$c_i = \frac{(2 * i - 1)\pi}{2N}, \quad i = 1, \dots, N \quad (4.163)$$

in  $h_i = 2.5\pi/N, \forall i$ . Fazni interval  $[0, 2\pi]$  pri tem opisuje eno periodo gibanja.

#### 4.3.2 Učenje DMP parametrov iz ene same demonstracije

Do sedaj smo prikazali pomen in vpliv posameznih členov. Kako bi pa morali postopati v idealnem primeru, da bi najprej demonstrirali robotu trajektorijo gibanja, nato pa iz tega izračunali potrebne sile in tem silam primerne trajektorije? Demonstracija gibanja lahko poteka na dva načina. Pri kinestetičnem vodenju oseba drži, premika in vodi gibanje samo vrha robota ali cele robotske roke. V drugi možnosti, pri posnemanju gibanja, oseba sama demonstrira izvedbo naloge, to gibanje pa je običajno zajeto z optičnim merilnim sistemom.

Želena trajektorija je označena kot  $y_d$ , gre za vektor s serijo točk v časovnem zaporedju. Z merjenjem tudi hitrosti in pospeška ali dvakratnim odvajanjem dobimo prvi in drugi časovni odvod.

$$\ddot{y}_d = \frac{\partial}{\partial t} \dot{y}_d = \frac{\partial}{\partial t} \frac{\partial}{\partial t} y_d. \quad (4.164)$$

Enačbo sistema poznamo že iz zgornje razlage, pospešek je znan v vsakem trenutku časa,

$$\ddot{y} = \alpha_y (\beta_y (g - y) - \dot{y}). \quad (4.165)$$

Z poznavanjem in razliko obeh hitrosti (pospeška) je mogoče določiti potrebno silo za generiranje takšne trajektorije ( $f_d = m \cdot \ddot{y}$ ,  $m = 1$ ):

$$\ddot{f}_d = \ddot{y}_d - \alpha_y (\beta_y (g - y) - \dot{y}). \quad (4.166)$$

Prav tako od prej poznamo člen  $f$ , sestavljen je iz utežene vsote baznih funkcij, ki so aktivirane skozi čas. Torej je mogoče uporabiti optimizacijsko tehniko, npr. lokalno uteženo regresijo za izbiro vrednosti uteži pri baznih funkcijah, tako da je funkcija  $f$  skladna z želeno trajektorijo  $f_d$ . Lokalno utežena regresija minimizira razliko

$$\sum_t \Psi_i(t) (f_d(t) - \omega_i(x(t)(g - y_0)))^2. \quad (4.167)$$

Rešitev je v obliki

$$\omega_i = \frac{s^T \Psi_i f_d}{s^T \Psi_i s}. \quad (4.168)$$

Pri čemer sta vektor  $s$  in diagonalna matrika  $\Psi_i$  enaka. Več podrobnosti je v Schall 1998 objavi.

$$\mathbf{s} = \begin{pmatrix} x_{t_0}(g - y_0) \\ \vdots \\ x_{t_N}(g - y_N) \end{pmatrix}, \quad \Psi_i = \begin{pmatrix} \Psi_i(t_0) & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \Psi_i(t_n) \end{pmatrix}. \quad (4.169)$$

Pojasnimo še direktni vpliv kompleksnosti trajektorije na zahtevano število baznih funkcij. Na prvem grafu spodaj želimo slediti najprej sinusni obliki trajektorije, zatem pa zelo nelinearni stopničasti trajektoriji.

V drugem primeru lahko vidimo, da DMP nikoli ne more natančno reproduciri želene poti, se pa približevanje izboljšuje pri povečanem številu osnovnih funkcij. Tovrstno počasno izboljšanje nelinearnih problemov je pričakovati od postavitve osnovnih funkcij. Uporabljeno je bila enakomerna postavitev središč osnovnih funkcij po časovni osi, obstajajo še druge metode za bolj strateško postavitev osnovnih funkcij. Glede na kompleksnost trajektorije v nekem področju je določeno število baznih funkcij, centri baznih funkcij in

njihova širina so temu skladni. V zelo nelinearnih področjih bi pričakovali veliko ozkih baznih funkcij, v linearnih področjih pa manj manj, vendar so te širše. Metoda je poimenovana kot lokalno utežena projekcijska regresija (locally weighted projection regression).

Opisani pristopi so resnično močno orodje, na predstavljenem osnovnem okviru je napravljenih veliko razširitev, npr. vključitev povratne povezave, amplitudno-časovno povezani DMP, uporaba DMP za spremijanje ojačenja ter trajektorije, vključitev minimizacijske funkcije in spodbujevanega učenja (reinforcement learning), prepoznavanje raznih aktivnosti ter drugo.

Predstavimo še nekoliko drugačen izračun parametrov  $\omega_i$ . Tudi tu so na začetku izmerjene s kinestetičnim vodenjem ali kinematicnimi meritvami, kjer so

$$\{y_d(t_j), \dot{y}_d(t_j), \ddot{y}_d(t_j)\}_{j=1}^T. \quad (4.170)$$

Izmerjene časovne vrste za pozicijo, hitrost in pospešek. Izračun lahko začnemo z osnovnima d.e., kjer z nadomestimo z  $\tau\ddot{y}$ , dobimo

$$\tau^2\ddot{y} + \alpha_z\tau\dot{y} - \alpha_z\beta_z(g - y) = f. \quad (4.171)$$

$f$  je definiran enako kot prej v obliki

$$f(x) = \frac{\sum_{i=1}^N \omega_i \Psi_i(x)}{\sum_{i=1}^N \Psi_i(x)} x(g - y_0), \quad \Psi_i(x) = \exp(-h_i(x - c_i)^2). \quad (4.172)$$

Časovna konstanta  $\tau$  mora biti enaka za vse prostostne stopnje. Možna je izbira  $\tau = t_T$ , pri tem  $t_T f$  podaja trajanje gibanja. Seveda je vrednost končne privlačne točke  $g$  drugačna pri različnih prostostnih stopnjah. Vrednost lahko dobimo kar iz  $g = y_d(t_T)$ .

Enačba za sistem 2. reda definira želeno privlačno silo

$$f_d(t_j) = \tau^2\ddot{y}_d(t_j) + \alpha_z\tau\dot{y}_d(t_j) - \alpha_z\beta_z(g - y_d(t_j)). \quad (4.173)$$

Z uporabo okrajšav vektorjev  $\mathbf{w}$  in  $\mathbf{f}$  v obliki

$$\mathbf{f} = \begin{bmatrix} f_d(t_1) \\ \dots \\ f_d(t_T) \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \omega_1 \\ \dots \\ \omega_N \end{bmatrix}. \quad (4.174)$$

Lahko zapišemo sledeči sistem linearnih enačb

$$\mathbf{X} \cdot \mathbf{w} = \mathbf{f} \quad (4.175)$$

Ta sistem je potrebno rešiti, da bi dobili uteži pri funkcijah DMP, ki kodirajo želeno gibanje. Matrika  $\mathbf{X}$  je podana kot

$$\mathbf{X} = (g - y_0) \begin{bmatrix} \frac{\Phi_1(x_1)}{\sum_{i=1}^N \Psi_i(x_i)} x_1 & \dots & \frac{\Phi_N(x_1)}{\sum_{i=1}^N \Psi_i(x_i)} x_1 & \dots \\ \dots & & \dots & \dots \\ \frac{\Phi_1(x_T)}{\sum_{i=1}^N \Psi_i(x_i)} x_T & \dots & \frac{\Phi_N(x_T)}{\sum_{i=1}^N \Psi_i(x_i)} x_T & \dots \end{bmatrix}. \quad (4.176)$$

Točke  $x_i$  dobimo podobno kot v prvotnih enačbah iz zapisa

$$x(t) = \exp(-\alpha_x t / \tau). \quad (4.177)$$

Vrednosti uteži  $\omega_i$  je mogoče izračunati z izračunom sistema linearnih enačb z uporabo najmanjših kvadratov. Takšen DMP zapis potem zagotovi, da robot doseže končno točko  $g$  v času  $t_T$ . Ker diskretni DMP predstavljajo diskretni gib iz točke v točko, se mora na koncu gib zaključiti ter zaustaviti v želeni točki. V kolikor z obravnavanimi orodji ponazorimo drugačne gibe, bo robot prenihal preko končne točke in se zatem vrnil nazaj na ciljno točko. Dinamika tega je odvisna od parametrov sistema 2. reda. Vsaj teoretično ni potrebno, da bi bila hitrost gibanja enaka nič na začetku gibanja.

V enačbah zgoraj predstavljenih so definirane tako, da je zagotovljena konvergenca robota kot dinamičnega sistema. Primer vrednosti parametrov bi bil  $\alpha_x = 2$ ,  $\beta_z = 3$ ,  $\alpha_z = 4\beta_z = 12$ . Trenutno obravnavani sistem uporabi celoten linearni sistem enačb za oceno parametrov  $\omega_i$ , medtem ko so bili v prvem prikazu zgoraj parametri  $\omega_i$  ocenjeni medsebojno neodvisno preko lokalno utežene regresije. Zadnji prikazani primer zapiše trajektorije bolj natančno, saj upošteva medsebojne vplive med funkcijami  $\Psi_i$ . Na drugi strani ima medsebojno neodvisna določitev  $\omega_i$  prednost pri npr. prisotnosti šuma v signalih. Lokalni pristop je posebej uporaben v povezavi s spodbujevanim učenjem, kjer ni potrebna dobra ocena začetnih trajektorij, saj so korigirane v sledenih korakih.

#### 4.3.2.1 Postopno učenje

Namesto direktnega izračuna sistema enačb, je mogoče uteži  $\omega$  tudi oceniti s pomočjo rekurzivne metode najmanjših kvadratov z vključenim faktorjem pozabljanja

$$\begin{aligned} \mathbf{P}_j &= \frac{1}{\lambda} \left( \mathbf{P}_{j-1} - \frac{\mathbf{P}_{j-1} x_j x_j^T \mathbf{P}_{j-1}}{\lambda + x_j^T \mathbf{P}_{j-1} x_j} \right), \\ \mathbf{w}_j &= \mathbf{w}_{j-1} + (f_j - x_j^T \mathbf{w}_{j-1}) \mathbf{P}_j x_j, \end{aligned} \quad (4.178)$$

kjer sta  $\mathbf{w}_j = [\omega_{1,j}, \dots, \omega_{N,j}]^T$  in  $\mathbf{P}_j$  trenutni oceni za parametre naklona in zunanja kovariančna matrika. Začetne vrednosti so enake  $\mathbf{P}_0 = \mathbf{I}$ ,  $\mathbf{w}_0 =$

$0, f_j = f_d(t_j), x_j$ . je  $M$  dimenzijski stolpični vektor povezan s pripadajočo vrstico sistemске matrike  $\mathbf{X}$ ,  $0 < \lambda \leq 1$  je faktor pozabljanja, medtem ko so končne vrednosti uteži podane kot  $\mathbf{w} = \mathbf{w}_T$ . Faktor pozabljanja postavimo na 1 že vprimeru, če je dostopen vsaj en primer gibanja, vendar je za takšen primer primernejša rešitev s standardno metodo najmanjših kvadratov.

#### 4.3.2.2 Oscilatorji z adaptivno frekvenco

Drugače kot pri času trajanja v primeru diskretnega gibanja pa frekvence osciliranja ni mogoče prepoznati direktno iz podatkov. Namesto ocenjevanja frekvence, je mogoče nadomestiti prej uporabljeni privzetek o konstantni hitrosti faze s sistemom

$$\begin{aligned}\dot{\phi}_i &= \Omega_i - Ke(t) \sin(\phi_i), \\ \dot{\Omega}_i &= Ke(t) \sin(\phi_i), \\ \dot{a}_i &= \eta \cos(\phi_i) e(t),\end{aligned}\quad (4.179)$$

kjer sta  $e(t) = y_d(t) - \hat{y}_t$  in  $\hat{y}_t = \sum_{i=1}^L \alpha_i \cos(\phi_i)$ . V primeru poenostavitev  $e(t) = 0$ , postanejo te tri enačbe enake prvotni enačbi  $\dot{\phi} = \Omega$ . Z integracijo teh enačb je mogoče oceniti vrednost frekvence  $\Omega_i$ , ki je zaznana v trajektoriji gibanja. Izbrana je najpomembnejša frekvanca, kot osnovna frekvanca  $\Omega$  za DMP izračune. Drugi avtorji so v literaturi predlagali nekoliko drugačen izračun gibalne frekvence, zasnovane na adaptivni frekvenci oscilatorja v kombinaciji z adaptivno Fourier-jevo vrsto. To vodi do sistema d.e. drugega reda

$$\begin{aligned}\dot{\Omega} &= -Ke(t) \sin(\phi), \\ \dot{\phi} &= \Omega_i - Ke \sin(\phi),\end{aligned}\quad (4.180)$$

kjer je  $\Omega$  pridobljena frekvanca,  $\phi$  je faza,  $K$  je povezovalna konstanta in  $e(t) = y_d(t) - \hat{y}(t)$  predstavlja napako, ozziroma razliko med trenutno vrednostjo  $y$  in ocenjeno vrednostjo  $\hat{y}$ . Ocenjena vrednost je podana kot

$$\hat{y}(t) = \sum_{c=0}^m (\alpha_c \cos(c\phi(t)) + \beta_c \sin(c\phi(t))). \quad (4.181)$$

V tem zapisu  $m$  predstavlja velikost Fourier-jeve vrste. Uteži  $\alpha_c$  in  $\beta_c$  se osvežita z novimi vrednostmi v skladu s sledečim pravilom

$$\begin{aligned}\dot{\alpha}_c &= \eta \cos(c\phi)c, \\ \dot{\beta}_c &= \eta \sin(c\phi)c,\end{aligned}\quad (4.182)$$

$\eta$  je učna konstanta. Velikost Fourier-jeve vrste postavim običajno na nizko vrednost, npr.  $m = 10$ .

### 4.3.3 DMP v kartezičnem prostoru

Metode za planiranje trajektorij v notranjih koordinatah sklepov je pogosto mogoče uporabiti tudi v zunanjih koordinatah, v kartezičnem prostoru. Pri tem se planiranje orientacij pogosto zakomplificira. Za razliko do pozicije, ki jo podajamo z 3D vektorjem, je potrebno orientacije predstaviti z Eulerjevimi koti, ta parametrizacija pa vedno vsebuje singularnosti. Od načina zapisa orientacije si torej želimo, da ne vnaša dodatnih nezveznosti, sicer neobstoječih v realnem svetu.

Najbolj standarden način predstavitve rotacijskega gibanja je s pomočjo rotacijskih matrik, ki predstavljajo devet-dimenzionalno parametrizacijo. Vsaka orientacija je predstavljena z enolično vrednostjo. Polno gibanje robota torej lahko predstavimo s tremi parametri za specifikacijo pozicijskega dela trajektorije in načeloma  $3 \times 3$  parametri za orientacijski del trajektorije. Vsak od teh je načeloma lahko predstavljen v svojem DMP sistemu. Vendar pri izračunu naletimo na nelinearne medsebojne odvisnosti rotacijskih parametrov, kot tudi nelinearne funkcije. Podrobnosti tega izračuna so podane v literaturi.

Orientacijo je mogoče predstaviti tudi z kvaternioni. Kvaternioni zagotavljajo zapis brez singularnosti, za to so potrebni štirje parametri, namesto devetih. Zapis z kvaternioni pa ni enoličen, saj kvaterniona  $\mathbf{q}$  in  $-\mathbf{q}$  predstavlja isto orientacijo. Pri zapisu z kvaternioni izhajamo iz rotacijskega DMP zapisa z matrikami. Nadaljevanje izračuna tukaj ne podajamo.

Primerjava zapisa rotacije z DMP z uporabo matrik in kvaternionov kažeta, da oba zapisa dobro reproducirata želen potek trajektorij. Pride pa pri zapisu z DMP osnovanimi na rotacijskih matrikah do nezveznosti pri  $\|\log(\mathbf{R})\| = \pi$ .

### 4.3.4 DMP tretjega reda

Tako preprosti načini formiranja robotove trajektorije, kot je uporaba polinomov, ali DMP sistem drugega reda zagotavljajo v osnovi zvezen potek samo pozicije in hitrosti v situacijah, kjer se ciljna točka g nenadoma spremeni. Ta problem zaobidemo z uporabo enačb tretjega reda

$$\begin{aligned} \tau \dot{z} &= \alpha_z (\beta_z (r - y) - z) + f(x) \\ \tau \dot{y} &= z \\ \tau \dot{r} &= \alpha_g (g - r). \end{aligned} \quad (4.183)$$

Faza  $x$  je ponovno definirana enako kot v prvotnih enačbah z d.e.

$$\tau \dot{x} = -\alpha_x x. \quad (4.184)$$

Splošna rešitev linearnega dela zgornjega sistema treh enačb, brez  $f(x)$  je podana kot

$$\begin{bmatrix} z(t) \\ y(t) \\ r(t) \end{bmatrix} = \begin{bmatrix} 0 \\ g \\ g \end{bmatrix} + \exp(\mathbf{A}t)\mathbf{c}, \quad \mathbf{A} = \frac{1}{\tau} \begin{bmatrix} -\alpha_z & -\alpha_z\beta_z & \alpha_z\beta_z \\ 1 & 0 & 0 \\ 0 & 0 & \alpha_g \end{bmatrix}, \quad (4.185)$$

matrika  $\mathbf{A}$  ima lastne vrednosti  $\lambda_1 = -\alpha_g/\tau$ ,  $\lambda_{2,3} = (-\alpha_z \pm \sqrt{\alpha_z^2 - 4\alpha_z\beta_z})/(2\tau)$ , ki so vedno realne in negativne, če so  $\tau, \alpha_z, \beta_z, \alpha_g > 0$ . Za take parametre je garantirana konvergenca proti ciljni privlačni točki  $[0, g, g]^T$ . Glede na enačbo za izračun lastnih vrednosti je sistem kritično dušeno  $\alpha_z = 4\beta_z$ , tedaj je  $\lambda_{2,3} = -\alpha_z/(2\tau)$ . Podobno kot prej se  $f(x)$  zmanjša proti nič v primeru  $t \rightarrow \infty$ , to pa garantira konvergenco v isto privlačno točko.

Omeniti tudi velja, da pri izenačenju  $r(t) = g$  sistem tretjega reda postane enak prvotnemu sistemu drugega reda. Torej lahko in moramo uporabljati originalni sistem drugega reda pri učenju DMP parametrov iz samo ene demonstracije. Sistem tretjega reda tako uporabljamamo samo pri izvajaju. Če se končna privlačna točka  $g$  nenadoma spremeni, sistem tretjega reda zagotovi počasen prehod spremenljivke  $r$  v novo ciljno točko. Pri tem je pospešek robota definiran z že znano prvo enačbo v sistemu treh enačb  $\tau \dot{z} = \alpha_z(\beta_z(r - y) - z) + f(x)$  ter  $\ddot{y} = \dot{z}/\tau$ .

Sistem tretjega reda je tudi uporaben za zvezzen prehod z enega DMP zapisa na drugega. Iz prvih dveh enačb (od treh) dobimo

$$r = \frac{\tau^2 \ddot{y} + \tau \alpha_z \dot{y} + \alpha_z \beta_z y - f(x)}{\alpha_z \beta_z}. \quad (4.186)$$

Če so torej robotova pozicija, hitrost in pospešek v trenutku prehoda podana kot  $y_c, \dot{y}_c$  in  $\ddot{y}_c$ , moramo naslednji DMP inicializirati na

$$y(0) = y_c, \quad z(0) = \tau \dot{y}_c, \quad r(0) = \frac{\tau^2 \ddot{y}_c + \tau \alpha_z \dot{y}_c + \alpha_z \beta_z y_c - f(1)}{\alpha_z \beta_z}, \quad x(0) = 1. \quad (4.187)$$

Na ta način zagotovimo zveznost tudi pospeškov.

### 4.3.5 Moduliranje DMPjev

#### 4.3.5.1 Zaustavljanje faze

Običajno enačbe

$$\tau \dot{x} = -\alpha_x x, \quad (4.188)$$

ne integriramo analitično v rešitev

$$x(t) = \exp(-\alpha_x t / \tau). \quad (4.189)$$

Namesto tega modificiramo

$$\begin{aligned} \tau \dot{x} &= -\alpha_x x, \\ \tau \dot{y} &= z, \end{aligned} \quad (4.190)$$

v zapis

$$\begin{aligned} \tau \dot{x} &= \frac{-\alpha_x x}{1 + \alpha_{px} |\tilde{y} - y|}, \\ \tau \dot{y} &= z + 1 + \alpha_{px} |\tilde{y} - y|, \end{aligned} \quad (4.191)$$

kjer  $y$  in  $\tilde{y}$  predstavljata trenutno in želeno pozicijo robota. V kolikor robot ne more slediti želenemu gibanju,  $1 + \alpha_{px} |\tilde{y} - y|$  postane velik, kar posledično povzroči majhno vrednost  $\dot{x}$ . Potek faze je tako zaustavljen, vse dokler ni ujeta želena konfiguracija  $y$ . Na drugi strani, v kolikor robot precizno sledi želeni trajektoriji, potem se zadnji dve enačbi ne razlikujeta od osnovnih enačb. Podobno je mogoče definirati v zunanjih koordinatah, DMP v kartezičnem prostoru. V tem primeru nadomestimo  $\tau \dot{x} = -\alpha_x x$  z enačbo

$$\tau \dot{x} = -\frac{\alpha_x x}{1 + \alpha_{px} (\|\tilde{\mathbf{p}} - \mathbf{p}\| + \gamma d(\tilde{\mathbf{q}}, \mathbf{q}))}, \quad (4.192)$$

kjer je  $\|\tilde{\mathbf{p}} - \mathbf{p}\| + \gamma d(\tilde{\mathbf{q}}, \mathbf{q})$  napaka sledenja trajektorije,  $\tilde{\mathbf{p}}$  in  $\tilde{\mathbf{q}}$  sta trenutni pozicija ter orientacija vrha robota,  $\mathbf{p}$  in  $\mathbf{q}$  sta pripadajoča izhoda iz DMP. Ne pozabimo, da v primeru velikih napak pri sledenju, napaka  $\|\tilde{\mathbf{p}} - \mathbf{p}\| + \gamma d(\tilde{\mathbf{q}}, \mathbf{q})$  postane velika, kar posledično povzroči majhno spremembo faze  $\dot{x}$ . Tako je napredovanje faze zaustavljeno, dokler robot ne zmanjša napake sledenja. Pri obravnavi DMP v kartezičnem prostoru, zgornja enačba  $\tau \dot{y} = z + \alpha_{px} |\tilde{y} - y|$  postane drugačna za pozicijski in orientacijski del trajektorije.

#### 4.3.5.2 Robustnost na motnje s povezovanjem DMPjev

Zapišimo samo enačbe

$$\tau \dot{x} = \frac{-\alpha_x x}{1 + \alpha_{px} \varepsilon(\tilde{\mathbf{p}}, \mathbf{p}_d, \mathbf{p}, \tilde{\mathbf{q}}, \mathbf{q}_d, \mathbf{q})}, \quad (4.193)$$

$$\tau \dot{\mathbf{p}} = \mathbf{z} + \alpha_p p (\tilde{\mathbf{p}} + \mathbf{p}_d - 2\mathbf{p}), \quad (4.194)$$

$$\tau \dot{\mathbf{q}} = \frac{1}{2} (\eta + \alpha_{pr} 2(\log(\tilde{\mathbf{q}} * \bar{\mathbf{q}}) + \log(\tilde{\mathbf{q}} * \bar{\mathbf{q}}))) * \mathbf{q}, \quad (4.195)$$

$$\varepsilon = \|\tilde{\mathbf{p}} - \mathbf{p}\| + \|\mathbf{p}_d - \mathbf{p}\| + \gamma(d(\tilde{\mathbf{q}}, \mathbf{q}) + d(\mathbf{q}_d, \mathbf{q})) \quad (4.196)$$

#### 4.3.5.3 Izgibanje oviram

Izogibanje oviram v Kartezičnem prostoru je lažje, če je tudi gibanje planirano v Kartezičnem prostoru. Predpostavimo 3DOF robota, želja je diskretno gibanje iz točke v točko v Kartezičnem prostoru. 3D pozicijski vektor za 3DOF dinamični sistem označimo kot  $\mathbf{y} = [y_1, y_2, y_3]^T$ . Želimo definirati gib primeren za seganje v ciljno točko  $\mathbf{g} = [g_1, g_2, g_3]^T$ . Na poti do ciljne točke se nahaja tudi ovira pozicionirana v točki  $\mathbf{o} = [o_1, o_2, o_3]^T$ , tej točki se moramo izogniti. Dodatno k pravkar podanim osnovnim oznakam, lahko z namenom izogibanja oviri formuliramo primeren povezovalni člen  $\mathbf{C} = [C_{t,1}, C_{t,2}, C_{t,3}^T]$ , izračunamo ga kot

$$\mathbf{C}_t = \text{sig}(\|\mathbf{o} - \mathbf{y}\|) \mathbf{R} \dot{\mathbf{y}} (\pi - \phi) \exp(-\beta\phi), \quad (4.197)$$

V tem zapisu so uporabljene sledeče okrajšave

$$\phi = \arccos \left( \frac{(\mathbf{o} - \mathbf{y})^T \dot{\mathbf{y}}}{\|\mathbf{o} - \mathbf{y}\| \|\dot{\mathbf{y}}\|} \right), \quad (4.198)$$

$$\text{sig}(x) = \frac{1}{1 + e^{\eta(x-d)}}, \quad (4.199)$$

$$\mathbf{R} = \exp((\frac{\pi}{2} - \phi)\mathbf{n}), \quad (4.200)$$

$$\mathbf{n} = \frac{(\mathbf{o} - \mathbf{y}) \times \dot{\mathbf{y}}}{\|\mathbf{o} - \mathbf{y}\| \|\dot{\mathbf{y}}\|} \quad (4.201)$$

$\gamma$ ,  $\beta$  in  $\eta$  so skalirni faktorji,  $d$  je razdalja na kateri začne ovira vplivati na robotovo gibanje.  $\mathbf{C}_t$  generira povezovalni člen generira komponento hitrosti, ki leži v ravnini definirani z vektorjem  $\mathbf{o} - \mathbf{y}$  ter  $\dot{\mathbf{y}}$ . Je ortogonalna na linijo  $\mathbf{o} - \mathbf{y}$ , ta povezuje vrh robota in oviro.

Že znani enačbi  $\tau \dot{\mathbf{z}} = z$  dodamo člen  $\mathbf{C}_t$ , s čimer zagotovimo izogibanje vrha robota omenjeni oviri.

$$\tau \dot{\mathbf{z}} = \alpha_z (\beta_z (\mathbf{g} - \mathbf{y}) - \mathbf{z}) + \mathbf{f}(x) + \mathbf{C}_t. \quad (4.202)$$

Primeri trajektorij gibanja so prikazani na naslednji sliki. Pri tem ne smemo pozabiti, da prikazani pristop zagotovi izogib ovir le za vrh robota, medtem ko ne zagotavlja izogib ovir vsem segmentom robota. Ena od možnosti je izračun ter uporaba ločenih DMP za vsak robotski segment. Tudi ta metoda ne upošteva strukturnih omejitev robota med segmenti, zato bi bila potrebna nadaljna prilagoditev preden bi pridobili fizično realistično rešitev.

### 4.3.6 Compliant Movement Primitives (CMP)

Metoda DMP temelji na parametričnem zapisu trajektorij (kot tudi ProMP) in je primerna za opise trajektorij kot funkcije ene vhodne spremenljivke (npr. čas). Metoda GMM omogoča zapis trajektorij kot funkcije več vhodnih spremenljivk. Takšen zapis je uporaben v določenih aplikacijah, vendar kompleksnost modela otežuje naknadno prilagajanje naučenega. Za prilagajanje sta metodi DMP in ProMP bolj primerni. Kot vemo je DMP definiran kot nelinearna diferencialna enačba drugega reda. Pri poznavanju parametrov enačbe omogoča tak zapis generiranje trajektorije iz poljubne začetne v želeno končno lego. Karakteristika ProMP je določena na osnovi statističnega opisa izvedenih demonstracij. Vendar je tak zapis v primerjavi z DMP metodo manj primeren za ekstrapolacijo trajektorije. Nasprotno je z DMP metodo spremjanje vmesnih leg na trajektoriji kompleksnejše kot s ProMP metodo.

Metoda CMP (Compliant Movement Primitives, Podajni generatorji gibov) predstavlja nadgradnjo metode DMP, saj poleg trajektorije opisuje tudi naprej v sklepih.

Naj bo robot sestavljen iz togih segmentov, dinamične enačbe gibanja zapisov obliki

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \tau, \quad (4.203)$$

kjer so  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  in  $\ddot{\mathbf{q}}$  sklepni koti, hitrosti in pospeški,  $\mathbf{H}(\mathbf{q})$  je matrika vztrajnosti,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$  so prispevki Coriolis in centripetalne sile,  $\mathbf{g}(\mathbf{q})$  so gravitacijske sile ter  $\epsilon(q, \dot{q}, \ddot{q})$  so nelinearnosti, npr. trenje. Predpostavimo tudi, da je inverzni dinamični model znan in zapisan kot  $\mathbf{f}_{\text{dynamic}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ . Ena od možnosti za sledenje želene vrednosti pozicije  $\mathbf{q}_d$  je s pomočjo PD regulatorja ob upoštevanju inverznega dinamičnega modela

$$\tau_u = \mathbf{K}(\mathbf{q}_d - \mathbf{q}) + \mathbf{D}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{f}_{\text{dynamic}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}). \quad (4.204)$$

Pri tem je  $\tau_u$  krmilni navor,  $\mathbf{D}$  je matrika dušenja,  $\mathbf{K}$  je diagonalna matrika, ki določa togost robota. Robot je tog pri velikih vrednostih  $\mathbf{K}$ . To tudi povzroči boljše sledenje želeni trajektoriji. Z zmanjšanjem vrednosti  $\mathbf{K}$  postane robot bolj podajan. V primeru ne dobro znanega ali netočnega modela se napake sledenja trajektorije znatno povečajo. Takšen pristop, poimenujemo ga lahko tudi standarden pristop, torej ne more kompenzirati kakršnekoli slabosti modela. Z namenom boljšega sledenja, pri še vedno ohranitvi voljnosti robota lahko dodamo še en člen

$$\tau_u = \mathbf{K}(\mathbf{q}_d - \mathbf{q}) + \mathbf{D}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \tau_f + \mathbf{f}_{\text{dynamic}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}). \quad (4.205)$$

$\tau_f$  je dodatni signal navora, ki kompenzira dinamiko povezano z nalogo. Dinamika povezana z nalogo običajno v literaturi ni zajeta v dinamični model robota, določiti pa jo je mogoče z matematičnim modeliranjem naloge. Modeliranje naloge je lahko kompleksno in dolgotrajno, zahteva pa delo eksperta. Za razliko od tega se pri CMP robot sam autonomno nauči specifične vrednosti  $f$ , te so povezane z nalogo v kontroliranem okolju, ali pod nadzorom človeka. Po tej učni fazi za specifično nalogo je robot sposoben točnega izvajanja gibov, ob tem pa je podajen. Podajnost pa pomeni manjše interakcijske sile v primeru nepričakovanih trkov.

#### 4.3.6.1 CMP in njihovo učenje

Najprej definiramo podajno gibanje kot kombinacija želene trajektorije gibanja (sklepnič kotov) in pripadajočih signalov navora

$$[h(t)] = [\mathbf{q}_d(t), \tau_f(t)] \quad (4.206)$$

kjer sta

$$\begin{aligned} \mathbf{q}_d(t) &= [q_{d1}(t), q_{d2}(t), \dots, q_{dP}(t)]^T \\ \tau_f(t) &= [\tau_{d1}(t), \tau_{d2}(t), \dots, \tau_{dP}(t)]^T. \end{aligned} \quad (4.207)$$

$P$  označuje število prostostnih stopenj. V CMP so vrednosti  $\mathbf{q}_d$  najprej pridobljene iz demonstracije operaterja in zapisane v obliki DMP. Pripadajoče vrednosti  $\tau_j$  dobimo z izvajanjem naučenih trajektorij z uporabo velikega ojačenja v krmilniku, kar zagotovi dobro točnost sledenja. Ti od naloge odvisni navori so zapisani kot linearna kombinacija radialnih baznih funkcij. Označimo jih lahko tudi kot primitivi navorov (TP, Torque primitives). Par v kombinaciji DMP ter TP sedaj predstavlja podajno gibanje, lahko ga poimenujemo kot CMP (Compliant Movement Primitives, Podajni generatorji gibov).

#### 4.3.6.2 Učenje trajektorij gibanja

Gibanje človeka je mogoče zajeti na različne načine. Kinestetično vodenje je ena možnost.

$$\mathbf{q}_x(t) = [q_{x1}(t), q_{x2}(t), \dots, q_{xP}(t)]^T. \quad (4.208)$$

Pri tem indeks  $x$  označuje primere,  $P$  pa število DOF. Pozicijski del trajektorije lahko za tem zapišemo kot

$$\tilde{\mathbf{q}}_x(t) = \{\mathbf{q}_x(t_1), \mathbf{q}_x(t_2), \dots, \mathbf{q}_x(t_T)\}. \quad (4.209)$$

tu  $T$  označuje število vzorcev v eni demonstraciji. Z več demonstracijami dobimo komplet gibalnih trajektorij v prostoru sklepov

$$\mathbf{Q}_x = \{\tilde{\mathbf{q}}_{xj}, \mathbf{c}_{qj}\}_{j=1}^N \quad (4.210)$$

kjer  $N$  predstavlja skupno število primerov  $\tilde{\mathbf{q}}_x$ . Vsak primer demonstrirane trajektorije uspešno izvede eno varianto naloge, ki je opisana s pomočjo parametrov naloge  $c_q$ . Parametri naloge lahko opišejo različne zadeve, začetno pozicijo gibanja diskretnega giba v zunanjih koordinatah  $c_q = x_{start}$ , končno točko diskretnega giba v sklepnih koordinatah  $\mathbf{c}_q = [q_1, q_2, \dots, q_P]^T$ , srednjo vrednost pozicije v kartezičnih koordinatah za periodično gibanje v dveh dimenzijah  $\mathbf{c}_q = [p_x, p_y]^T$ . Pri tem je vsaka trajektorija  $\tilde{\mathbf{q}}_x$  kodirana z DMP, uporabijo se torej osnovne DMP enačbe, zapisane v nadaljevanju. Enačbe v nadaljevanju so zapisane za eno DOF, lahko jih uporabimo več za več DOF. Nelinearni sistem diferencialnih enačb torej definira DMP za diskretno gibanje iz toče v točko in za periodično gibanje.

$$\begin{aligned} v\dot{z} &= \alpha_z(\beta_z(g - y) - z) + f(s) \\ v\dot{y} &= z \end{aligned} \quad (4.211)$$

V teh osnovnih enačbah zagotovi linearni del konvergenco  $y$  proti ciljni točki  $g$  v primeru konvergence fproti nič. Nelinearni del  $f(s)$  spremeni obliko gibanja in je zapisan s pomočjo linearne kombinacije  $L_d$  baznih funkcij

$$f(s) = \frac{\sum_{b=1}^{L_d} \omega_{qb} \psi_b(s)}{\sum_{b=1}^{L_d} \psi_b(s)} s. \quad (4.212)$$

$\psi_b$  označuje Gaussove bazne funkcije.

$$\psi_b = \exp(-d_b(s - c_b)^2) \quad (4.213)$$

S centrom  $c_b$  in širino  $d_b > 0$ . Ne pozabimo, da  $f(s)$  ni direktno časovno odvisen. Namesto tega je fazna spremenljivka  $s$  z začetno vrednostjo  $s(0) = 1$  uporabljena v

$$f(s) = v\dot{s} = -\alpha_s s. \quad (4.214)$$

Kot je zgoraj že omenjeno, je fazna spremenljivka uporabljena v primeru zunanjih motenj. Ne pozabimo, da je skupna za vse DOF. V trenutnem primeru je faza enotna za vse CMP, ki poleg DMP vključuje tudi TP.

S primernim definiranjem konstant  $\alpha_z$ ,  $\beta_z$ ,  $v > 0$  in  $\alpha_s > 0$  ta opisani sistem garantirano konvergira proti ciljni točki  $g$ . Praktično uporabljene vrednosti so npr.  $\alpha_z = 48$ ,  $\beta_z = \alpha_z/4$ ,  $\alpha_s = 2$ , ter  $L_d = 30$  te vrednosti so določene eksperimentalno.

Z namenom zapisa primernega signala za učenje, zapišemo prvi dve enačbi v obliki drugega reda

$$v^2 \ddot{y} + \alpha_z v \dot{y} + \alpha_z \beta_z (g - y) = f(s) \quad (4.215)$$

Z zamenjavo spremenljivke  $y$  z demonstrirano trajektorijo n-tega sklepa  $\mathbf{q}_{xn}$  in obema odvodoma  $\dot{\mathbf{q}}_{xn}$  in  $\ddot{\mathbf{q}}_{xn}$  dobi enačba drugega reda obliko

$$\mathbf{f}_{ni} = v^2 \ddot{\mathbf{q}}_{xn}(t_i) + \alpha_z v \dot{\mathbf{q}}_{xn}(t_i) + \alpha_z \beta_z (\mathbf{g} - \mathbf{q}_{xn}(t_i)), \quad i = 1, \dots, T, \quad (4.216)$$

kjer je ciljna točka  $g$  definirana kot končna vrednost referenčne trajektorije  $q_{xn}(t_T)$ . DMP se izračuna z rešitvijo pravkar zapisanega predoločenega sistema enačb. To naredimo z uporabo regresije za vsako sklepno trajektorijo posebej.

Do sedaj zapisane enačbe veljajo za diskretno gibanje. V primeru periodičnega gibanja nadomestimo linearno kombinacijo Gaussovih funkcij  $f(s)$  z linearno kombinacijo periodičnih funkcij zapisanih kot

$$f(\phi) = \frac{\sum_{b=1}^{L_p} \omega_{qb} \Gamma_b(\phi)}{\sum_{b=1}^{L_p} \Gamma_b(\phi)} r, \quad \Gamma_b(\phi) = \exp(h_b(\cos(\phi - c_b) - 1)), \quad (4.217)$$

kjer je  $r$  amplituda oscilatorja in  $h_b > 0$ . Deluječe vrednosti parametrov so  $L_p = 30$  in  $h_b = 2.5L_p$ . Če v osnovnih enačbah nadomestimo  $v$  z  $v = 1/\Omega$ , dobimo

$$\begin{aligned} \dot{z} &= \Omega(\alpha_z(\beta_z(g - y) - z) + f(\phi)) \\ \dot{y} &= \Omega z \end{aligned} \quad (4.218)$$

V tem primeru je diskretna faza  $s$  zamenjana s periodično fazo  $\phi$ , ki je določena s faznim oscilatorjem

$$\dot{\phi} = \Omega, \quad (4.219)$$

$\Omega$  je frekvenca oscilacij.

Zapis periodičnih trajektorij je izveden podobno kot pri diskretnih trajektorijah, le da je pri tem uporabljen frekvenca oscilacij  $\Omega$ . V idealnem primeru je  $\Omega$  določena avtomatsko iz podatkov.

#### 4.3.6.3 Učenje pripadajočih signalov navora

Vzorčni poteki navorov

$$\begin{aligned}\tilde{\tau}_x(t) &= \{\tau_{x1}(t), \tau_{x2}(t), \dots, \tau_{xP}(t)\} \\ \tau_x(t) &= [\tau_{x1}(t), \tau_{x2}(t), \dots, \tau_{xP}(t)]^T,\end{aligned}\quad (4.220)$$

vzorčni poteki trajektorij  $\mathbf{q}_x$ , zapisani kot DMP izvedemo v kontorliranem okolju, oz. pod nadzorom operaterja. Za ta namen je uporabljen standardni krmilnik z npr. velikim ojačanjem

$$\tau_u = \mathbf{K}(\mathbf{q}_d - \mathbf{q}) + \mathbf{D}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{f}_{dynamic}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}). \quad (4.221)$$

Tudi potrebne hitrosti  $\dot{\mathbf{q}}_x$  zagotovijo DMP. Kot že vemo, je potrebno veliko ojačenje  $\mathbf{K}$ , kar zagotovi dobro sledenje trajektoriji. Seveda pri tem tudi predpostavimo, da je robot sposoben natančno slediti želeni trajektoriji. Na tej stopnji se dinamični model specifičen za trenutno nalogu uči z beleženjem navorov v krmilniku. Če so vrednosti  $\mathbf{K}$  majhne, ali je dinamični model netočen, ali ne vključuje trenutne dinamike za nalogu, potem krmilnik ne bo sposoben slediti želeni trajektoriji  $q_x$ , zabeležene pa bodo napačne vrednosti navorov. Specifične navore za nalogu  $\tau_x$  določimo kot razliko med  $\tau_m$  ter znano robotovo dinamiko  $\mathbf{f}_{dynamic}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  v posameznih robotskeh sklepih

$$\tau_x = \tau_m - \mathbf{f}_{dynamic}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}). \quad (4.222)$$

Tukaj predpostavimo, da poznamo robotov dinamični model, medtem ko je razlika  $\tau_m$  v navorih potrebna za kompenzacijo napak povezanih specifično z nalogo. Če model robota ni poznan imamo enostavnejši zapis

$$\tau_u = \mathbf{K}(\mathbf{q}_d - \mathbf{q}) + \mathbf{D}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}), \quad (4.223)$$

navori za učenje so definirani kot

$$\tau_x = \tau_m, \quad (4.224)$$

Kar sedaj kompenzora navore povezane z naloge in tudi robotovo dinamiko.

Vsek demonstriran  $\mathbf{q}_x$  je mogoče izvesti večkrat, vendar pod različnimi pogoji naloge  $c_\tau$  z različnimi podajnimi trajektorijami, ki imajo različne navore glede na naloge, vendar enako pozicijsko trajektorijo. Mogoče je uporabiti isto trajektorijo, vendar različne hitrosti, kar vodi do drugačnih podajnosti z različnimi navori. Še drug primer je premikanje objektov s spremenljivo maso  $c_\tau = m$  ter isto pozicijsko trajektorijo. V tem primeru bodo pripadajoči navori drugačni za vsako vrednost mase. Moramo torej dobiti komplet primernih navorov

$$\mathbf{T}_x = \{\tilde{\tau}_{xk}, \mathbf{c}_{\tau k}\}_{k=1}^{NM}, \quad (4.225)$$

kjer je  $M$  število izvajanj za vsako vzorčno trajektorijo  $\mathbf{q}_{xj}$  pri  $N$  različnih nalogah s  $\mathbf{c}_\tau$ . Linearna kombinacija baznih funkcij je uporabljena za kodiranje navore glede na naloge  $\tilde{\tau}_x$  kot TP. za diskretne gibe je navor za 1 DOF podan kot

$$\tau_x(s) = \begin{cases} \frac{\sum_{b=1}^{L_d} \omega_{\tau b} \psi_b(s)}{\sum_{b=1}^{L_d} \psi_b(s)} & s \geq s_\epsilon \\ \tau_x(s_\epsilon) & s < s_\epsilon, \end{cases} \quad (4.226)$$

kjer  $s_\epsilon$  predstavlja končno vrednost fazne spremenljivke, povezane z signalom navora med učenjem. Na ta način je končna vrednost navora ustrezno definirana, čeprav se CMP morda izvaja tudi preko končne, naučene točke. Za periodično gibanje so navori podani kot

$$\tau_x(\phi) = \frac{\sum_{b=1}^{L_p} \omega_{\tau b} \Gamma_b(\phi)}{\sum_{b=1}^{L_p} \Gamma_b(\phi)}. \quad (4.227)$$

Regresijske tehnike so uporabljene za izračun TP, torej  $\tau_x(s)$  in  $\tau_x(\phi)$  z rešitvijo poenostavljenega sistema enačb

$$f_{ni} = \tau_{xn}(t_i), \quad i = 1, \dots, T. \quad (4.228)$$

S pomočjo demonstracij človeka in izvedb pod različnimi pogoji je močno zajeti komplet  $NM$  primerov podajnega gibanja, torej pare gibalnih trajektorij in pripadajočih navorov

$$\mathbf{H}_x = \left\{ \{\mathbf{q}_{x1}, \tau_{x1}\}, \dots, \{\mathbf{q}_{xM}, \tau_{xM}\}, \right. \\ \left. \{\mathbf{q}_{x2}, \tau_{x(M+1)}\}, \dots, \{\mathbf{q}_{x2}, \tau_{x(2M)}\}, \dots, \right. \\ \left. \{\mathbf{q}_{xN}, \tau_{x((N-1)M+1)}\}, \dots, \{\mathbf{q}_{xN}, \tau_{x(NM)}\} \right\}, \quad (4.229)$$

ki so lahko uporabljeni za podajno izvajanje nalog pri spremenljivih pogojih

$$\mathbf{C}_x = \left\{ \{\mathbf{c}_{q1}^T, \mathbf{c}_{\tau1}^T\}, \dots, \{\mathbf{c}_{q1}^T, \mathbf{c}_{\tau M}^T\}, \right. \\ \left. \{\mathbf{c}_{q2}^T, \mathbf{c}_{\tau(M+1)}^T\}, \dots, \{\mathbf{c}_{q2}^T, \mathbf{c}_{\tau(2M)}^T\}, \dots, \right. \\ \left. \{\mathbf{c}_{qN}^T, \mathbf{c}_{\tau((N-1)M+1)}^T\}, \dots, \{\mathbf{c}_{qj}^T, \mathbf{c}_{\tau(NM)}^T\} \right\}, \quad (4.230)$$

Z zapisom gibalnih trajektorij s pomočjo DMP in pripadajočih signalov navorov kot TP dobimo komplet  $NM$  primerov CMP

$$\mathbf{H}_x^{CMP} = \{\omega_{qk}, \mathbf{g}_{qk}, \omega_{\tau k}, v_k, \mathbf{c}_k\}, \quad k = 1, \dots, NM. \quad (4.231)$$

CMP so predstavljeni kot DMP uteži  $\omega_{qk}$  in cilji  $\mathbf{g}_{qk}$ , TP uteži  $\omega_{\tau k}$ , skupno trajanje DMP in TP  $v_k$  in skupni pogoji nalog  $\mathbf{c}_k = [\mathbf{c}_{qj}^T, \mathbf{c}_{\tau k}^T]^T$ .

#### 4.3.7 Pobabilistic Movement Primitives (ProMP)

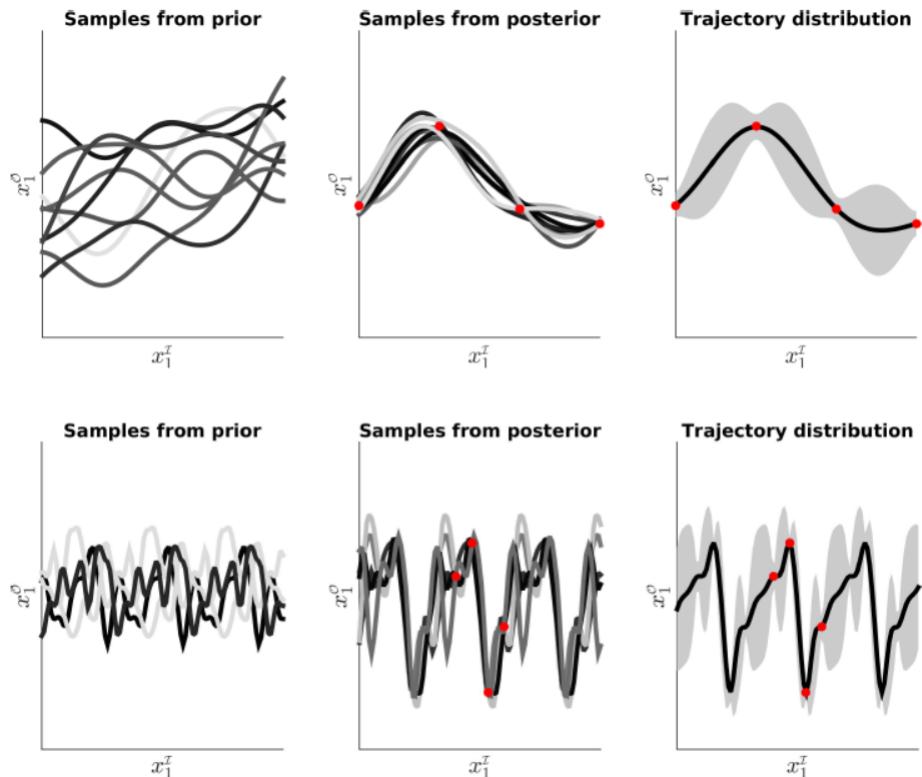
ProMP modelira naučene porazdelitve poti iz stohastičnih gibanj. Verjetnostne operacije, kot npr. pogojevanje lahko uporabimo za definiranje splošne rešitve ali za kombinirane in zlepljene odseke gibanja. Mogoče je definirati stohastični povratnozančni krmilnik, ki omogoča zapis variabilnosti gibanja in povezave med posameznimi prostostnimi stopnjami robota.

#### 4.3.8 Kernelized Movement Primitives (KMP)

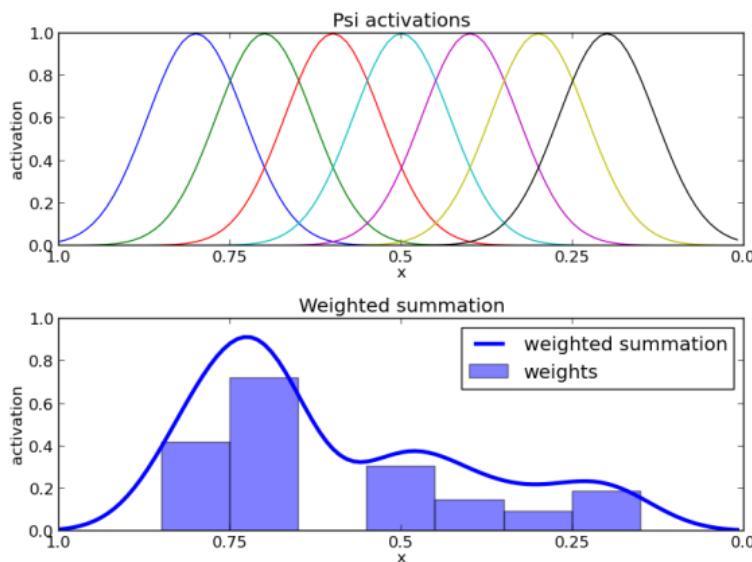
Nekatere omejitve preostalih metod delno odpravi metoda jedrni generatorji gibov (angl. Kernelized Movement Primitives - KMP), ki ohranja statistične lastnosti ProMP metode, omogoča ekstrapolacijo in nprilaganje vmesnih točk ter je primerna za zapis trajektorij kot funkcije več neodvisnih spremenljivk.

#### 4.3.9 Gaussian Mixture Models (GMM)

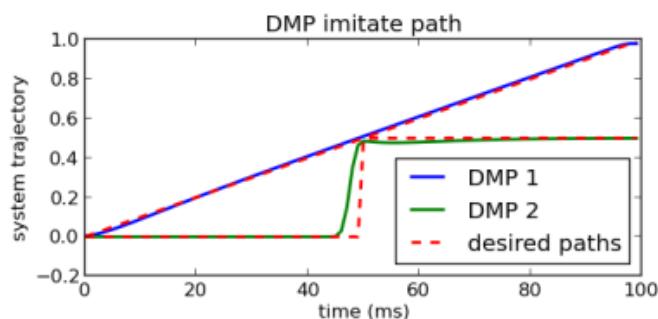
Glej kratek opis v okviru podpoglavlja GMR.



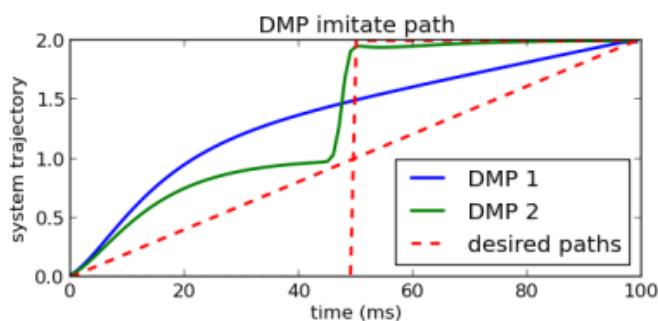
Slika 4.36: Ilustracija Gaussian process regression (GPR) za dve različni jedri (zgoraj RBF, spodaj periodična). Prvi stolpec kaže stohastične vzorce generirane iz predhodne distribucije (prior)  $\mathbf{x}^{O*} \sim \mathcal{N}(\mu(\mathbf{x}^{I*}), \mathbf{K}(\mathbf{x}^{I*}, \mathbf{x}^{I*}))$ . Drugi stolpec prikazuje stohastične vzorce generirane iz kasnejše distribucije (posterior)  $\mathbf{x}^{O*} | \mathbf{x}^O \sim \mathcal{N}(\mu^*, \Sigma^*)$  z podatkovnimi točkama  $\{\mathbf{x}^{O*} | \mathbf{x}^O\}$  označenimi z rdečo barvo. Zadnji stolpec kaže povezano distribuciju (associated)  $\mathcal{N}(\mu^*, \Sigma^*)$ , pri tem je ena standardna devijacija označena z osenčeno površino.



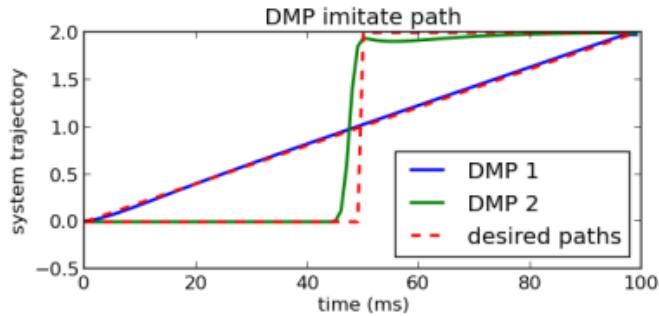
Slika 4.37: Aktivacija funkcij.



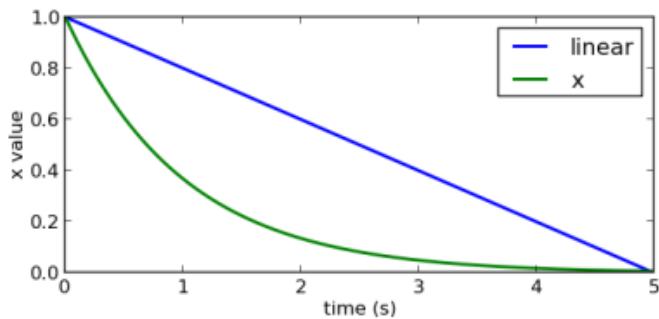
Slika 4.38: Dva primera



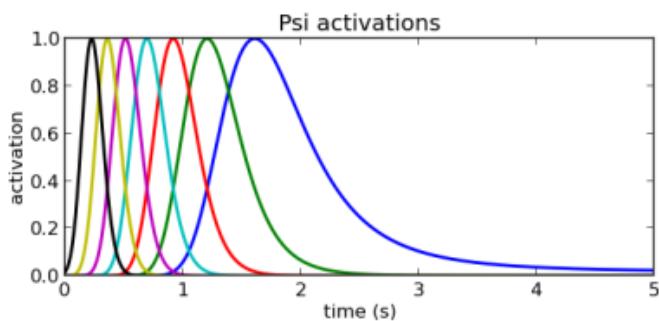
Slika 4.39: Ciljna točka postavljena na 2



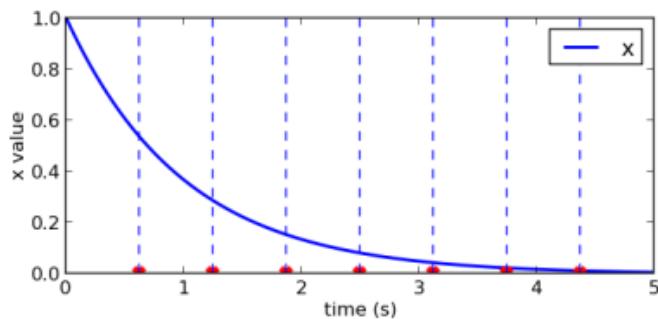
Slika 4.40: Vključitev še dodatnega člena



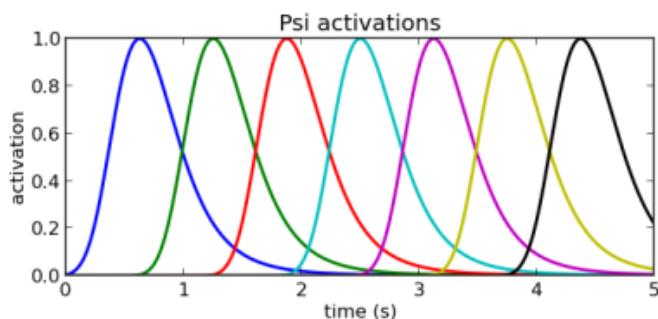
Slika 4.41: Linearno in eksponentno zmanjševanje



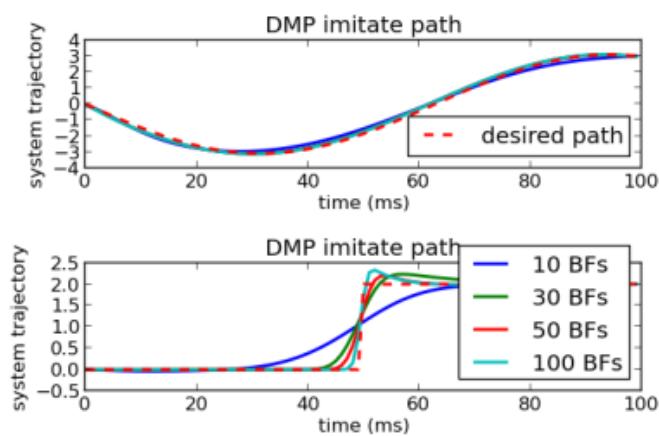
Slika 4.42: Grupacija funkcij pri eksponentnem upadu



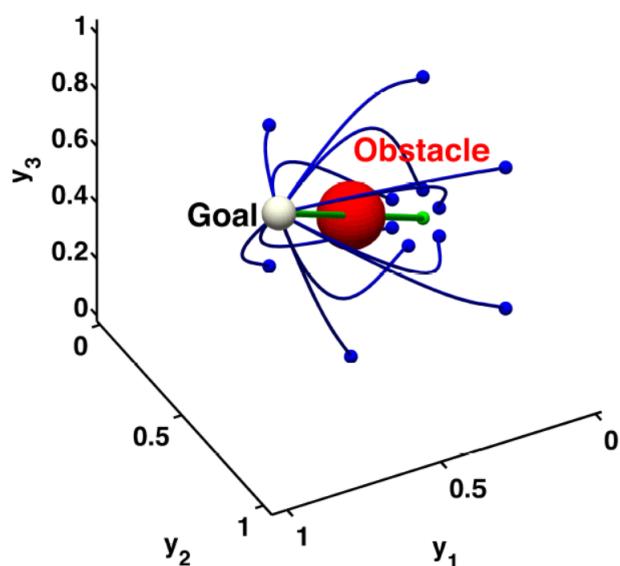
Slika 4.43: Nova razporeditev baznih funkcij



Slika 4.44: Nova razporeditev baznih funkcij



Slika 4.45: Vpliv števila baznih funkcij



Slika 4.46: Ovira je predstavljena z rdečo kroglo. Prikazane so različne trajektorije, izhajajo iz različnih začetnih pozicij, končajo pa se v krogle označeni kot Goal. Prikazana je tudi sredinska trajektorija, ki je uporabljena v situaciji brez prisotne ovire. Začetne točke s trajektorijami brez prečkanja z oviro se spremenijo le malo, medtem ko so druge trajektorije bistveno drugačne.

# Poglavlje 5

## LAGRANGEVA DINAMIKA

Izpeljava dinamičnega modela manipulatorja je ključen element na poti proti simulaciji gibanja, analizi zgradbe in načrtovanju regulacijskih algoritmov. S simulacijo manipulatorjevega gibanja je mogoče testirati regulacijske algoritme in planirati gibanje manipulatorja. Analiza preko dinamičnega modela je dobrodošla pri konstruiranju mehanike prototipnih verzij manipulatorja. Izračun sil in navorov pri izvajanju tipičnih gibov prinese koristne informacije za načrtovanje sklepov, prenosov in aktuatorjev.

Namen tega poglavja je predstavitev dveh metod za izpeljavo enačb gibanja manipulatorja v notranjih koordinatah, koordinatah sklepov. Prva metoda je zasnovana na Lagrangeovi formulaciji in je konceptualno preprosta in sistematična. Druga metoda temelji na Newton-Eulerjevi formulaciji in omogoča izpeljavo modela v rekurzivni obliki. Računsko je ugodnejša, saj izkorišča lastnost odprte kinematične strukture manipulatorja z nično silo na vrhu. Nakazan je tudi problem dinamične identifikacije parametrov. Poglavlje se zaključuje z izpeljavo dinamičnega modela manipulatorja v zunanjih koordinatah in z definicijo elipsoida dinamične manipulabilnosti.

### 5.1 LAGRANGEVA FORMULACIJA

Dinamični model manipulatorja omogoča zapis relacije med pogonskimi navori v sklepih in gibanjem robotske strukture.

Preko Lagrangeove formulacije je možno izpeljati enačbe gibanja na sistematičen način, neodvisno od referenčnega koordinatnega sistema. Najprej izberemo spremenljivke  $\lambda_i$ ,  $i = 1, \dots, n$  imenovane generalizirane kooordinate, ki opisujejo lego manipulatorja z  $n$  prostostnimi stopnjami. Na osnovi

znanih  $\lambda_i$  je definirana Lagrangeova funkcija mehanskega sistema kot funkcija generaliziranih koordinat

$$L = T - U. \quad (5.1)$$

$T$  in  $U$  sta skupna kinetična energija in potencialna energija sistema. Lagrangeove enačbe zapišemo kot

$$\frac{d}{dt} \cdot \frac{\partial L}{\partial \dot{\lambda}_i} - \frac{\partial L}{\partial \lambda_i} = \xi_i, \quad i = 1, \dots, n. \quad (5.2)$$

kjer so  $\xi_i$  generalizirane sile povezane z generaliziranimi koordinatami  $\lambda_i$ .

Za manipulator z odprto kinematično verigo je najbolj logična izbira generaliziranih koordinat v obliki vektorja spremenljivk v sklepih

$$\mathbf{q} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix}. \quad (5.3)$$

K generaliziranim silam prispevajo nekonzervativne sile, torej navori v aktuatorjih sklepov, navori v sklepih zaradi trenja, kot tudi navori v sklepih povzročeni zaradi sil na vrhu manipulatorja v dotiku z okoljem. Navor je funkcija generaliziranih sil.

Primer 1:

Da bi bolje razumeli pristop Lagrangeove formulacije pri izpeljavi dinamičnih enačb, bomo podrobno predstavili preprost primer nihala. Naj bo  $\tau$  pogonski navor v rotacijski osi. Vpliv viskoznega trenja upoštevamo okrog iste osi (Slika 5.1).

Motor posreduje pogonski navor  $\tau$  preko reduksijskega razmerja  $k_r > 1$  in vztrajnostnega momenta  $I_m$  okrog osi motorja.  $\vartheta$  naj bo kot glede na referenčno pozicijo, tako da bo nihalo viselo navzdol pri  $\vartheta = 0$ . Pri izbiri  $\vartheta$  kot generalizirane koordinate lahko kinetično energijo sistema zapišemo kot

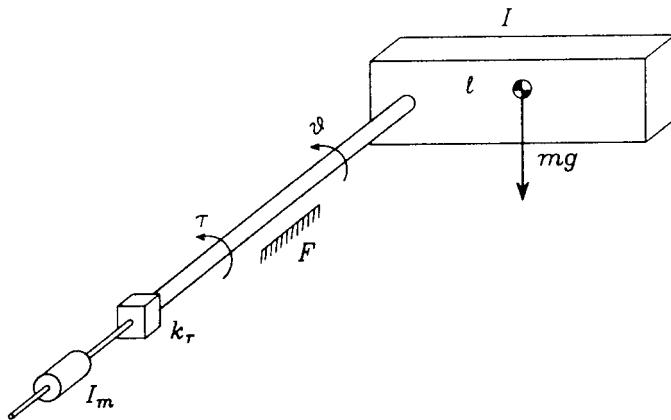
$$T = \frac{1}{2} I \dot{\vartheta}^2 + \frac{1}{2} I_m k_r^2 \dot{\vartheta}^2, \quad (5.4)$$

kjer je  $I$  vztrajnostni moment telesa okrog rotacijske osi. Kot naslednji korak izrazimo potencialno energijo v obliki

$$U = mgl(1 - \cos \vartheta), \quad (5.5)$$

kjer je  $m$  masa telesa,  $g$  je gravitacijski pospešek ( $9.81 \text{m/s}^2$ ) in  $l$  razdalja od težišča do rotacijske osi. Z upoštevanjem obih energij je Lagrangeova funkcija sistema enaka

$$L = \frac{1}{2} I \dot{\vartheta}^2 + \frac{1}{2} I_m k_r^2 \dot{\vartheta}^2 - mgl(1 - \cos \vartheta). \quad (5.6)$$



Slika 5.1: Z motorjem gnano nihalo.

Po odvajanjih izraza za  $L$  je Lagrangeova enačba enaka

$$(I + I_m k_r^2) \ddot{\vartheta} + mgl \sin \vartheta = \xi. \quad (5.7)$$

Generalizirana sila  $\xi$  je podana kot razlika med navorom aktuatorja  $\tau$  in navorom zaradi viskoznega trenja  $F\dot{\vartheta}$ , torej

$$\xi = \tau - F\dot{\vartheta}, \quad (5.8)$$

kar vodi do kompletnega dinamičnega modela sistema v obliki diferencialne enačbe drugega reda

$$(I + I_m k_r^2) \ddot{\vartheta} + F\dot{\vartheta} + mgl \sin \vartheta = \tau. \quad (5.9)$$

### 5.1.1 Izračun kinetične energije

Zamislimo si manipulator z  $n$  togimi segmenti. Skupna kinetična energija je potem podana kot vsota prispevkov zaradi:

- (relativnega) gibanja posameznih segmentov,
- gibanja vsakega aktuatorja,

$$T = \sum_{i=1}^n (T_{l_i} + T_{m_i}). \quad (5.10)$$

$T_{l_i}$  je kinetična energija segmenta  $i$  in  $T_{m_i}$  je kinetična energija motorja, ki poganja segment  $i$ .

Prispevek zaradi gibanja segmenta  $i$  je podan z

$$T_{l_i} = \frac{1}{2} \int_{V_{l_i}} \dot{\mathbf{p}}_i^{*T} \dot{\mathbf{p}}_i^* \rho dV, \quad (5.11)$$

kjer  $\dot{\mathbf{p}}_i^*$  ponazarja vektor linearne hitrosti in  $\rho$  gostoto elementa volumna  $dV$ .  $V_{l_i}$  je volumen segmenta  $i$ .

Upoštevajmo vektor pozicije  $\mathbf{p}_i^*$  infinitezimalnega delca in vektor pozicije  $\mathbf{p}_{l_i}$  za težišče segmenta, oba izražena v osnovnem koordinatenem sistemu (5.2). Vektor razdalje je potem

$$\mathbf{r}_i = [r_{ix} \ r_{iy} \ r_{iz}]^T = \mathbf{p}_i^* - \mathbf{p}_{l_i}, \quad (5.12)$$

in ob upoštevanju  $m_{l_i}$  mase segmenta tudi

$$\mathbf{p}_{l_i} = \frac{1}{m_{l_i}} \int_{V_{l_i}} \mathbf{p}_i^* \rho dV. \quad (5.13)$$

Tako lahko izračunamo vektor težišča. Kot posledico je mogoče hitrost točke na segmentu izraziti kot

$$\dot{\mathbf{p}}_i^* = \dot{\mathbf{p}}_{l_i} + \boldsymbol{\omega}_i \times \mathbf{r}_i = \dot{\mathbf{p}}_{l_i} + \mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_i, \quad (5.14)$$

kjer je  $\dot{\mathbf{p}}_{l_i}$  linearna hitrost težišča in  $\boldsymbol{\omega}_i$  kotna hitrost segmenta (Slika 5.2).

Splošen dokaz

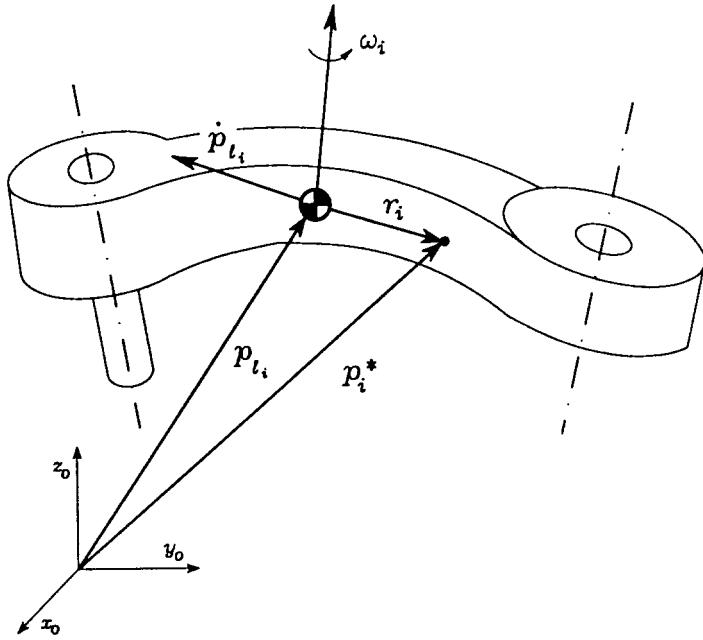
$$\begin{aligned} \mathbf{M}(\mathbf{a})\mathbf{p} &= \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} -a_z p_y + a_y p_z \\ a_z p_x - a_x p_z \\ -a_y p_x + a_x p_y \end{bmatrix} \\ \mathbf{a} \times \mathbf{p} &= \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_x & a_y & a_z \\ p_x & p_y & p_z \end{bmatrix} = \begin{bmatrix} a_y p_z - a_z p_y \\ a_z p_x - a_x p_z \\ a_x p_y - a_y p_x \end{bmatrix} \end{aligned}$$

Trditev je zaradi končnega rezulta  $\boldsymbol{\omega}_i \times \mathbf{r}_i = \mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_i$  pravilna.

Po vstavitvi enačbe za hitrost en. (5.14) v (5.11) lahko ugotovimo, da je kinetična energija vsakega segmenta sestavljena iz sledečih prispevkov.

Translacijski prispevek vsebuje  $\dot{\mathbf{p}}_{l_i}^*$ , linearne hitrost

$$\frac{1}{2} \int_{V_{l_i}} \dot{\mathbf{p}}_{l_i}^T \dot{\mathbf{p}}_{l_i} \rho dV = \frac{1}{2} m_{l_i} \dot{\mathbf{p}}_{l_i}^T \dot{\mathbf{p}}_{l_i}. \quad (5.15)$$



Slika 5.2: Robotski segment.

Medsebojni vpliv (dvakrat prvi krat drugi glede na enačbi (5.11) in (5.14))

$$2\left(\frac{1}{2} \int_{V_{l_i}} \dot{\mathbf{p}}_{l_i}^T \mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_i \rho dV\right) = 2\left(\frac{1}{2} \dot{\mathbf{p}}_{l_i}^T \mathbf{S}(\boldsymbol{\omega}_i) \int_{V_{l_i}} (\mathbf{p}_i^* - \mathbf{p}_{l_i}) \rho dV\right) = 0, \quad (5.16)$$

$\mathbf{r}_i$  vzamemo iz enačbe (5.12) in ugotovimo, da je ta integral po celiem segmentu enak 0, ker je

$$\int_{V_{l_i}} \mathbf{p}_i^* \rho dV = \mathbf{p}_{l_i} \int_{V_{l_i}} \rho dV. \quad (5.17)$$

Rotacijski prispevek je enak

$$\frac{1}{2} \int_{V_{l_i}} \mathbf{r}_i^T \mathbf{S}^T(\boldsymbol{\omega}_i) \mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_i \rho dV = \frac{1}{2} \boldsymbol{\omega}_i^T \left( \int_{V_{l_i}} \mathbf{S}^T(\mathbf{r}_i) \mathbf{S}(\mathbf{r}_i) \rho dV \right) \boldsymbol{\omega}_i,$$

kjer je upoštevana lastnost  $\mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_i = -\mathbf{S}(\mathbf{r}_i) \boldsymbol{\omega}_i$  (tu gre za enakost  $\mathbf{r} \times \boldsymbol{\omega} = -\boldsymbol{\omega} \times \mathbf{r}$ ). Prvi produkt je  $\mathbf{r}_i^T \mathbf{S}^T(\boldsymbol{\omega}_i) = -\boldsymbol{\omega}_i^T \mathbf{S}^T(\mathbf{r}_i)$  in drugi  $\mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_i = -\mathbf{S}(\mathbf{r}_i) \boldsymbol{\omega}_i$ . Glede na izraz za matrični operator  $\mathbf{S}(\cdot)$

$$\mathbf{S}(\mathbf{r}_i) = \begin{bmatrix} 0 & -r_{iz} & r_{iy} \\ r_{iz} & 0 & -r_{ix} \\ -r_{iy} & r_{ix} & 0 \end{bmatrix}, \quad (5.18)$$

dobimo produkt  $\mathbf{S}^T(\mathbf{r}_i)\mathbf{S}(\mathbf{r}_i)$  zatem pomnožimo posamezne komponente obeh matrik, nakar je mogoče integral na levi zapisati kot produkt na desni

$$\frac{1}{2} \int_{V_{l_i}} \mathbf{r}_i^T \mathbf{S}^T(\boldsymbol{\omega}_i) \mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_i \rho dV = \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_{l_i} \boldsymbol{\omega}_i. \quad (5.19)$$

Matrika na desni

$$\begin{aligned} \mathbf{I}_{l_i} &= \begin{bmatrix} \int(r_{iy}^2 + r_{iz}^2) \rho dV & -\int r_{ix} r_{iy} \rho dV & -\int r_{ix} r_{iz} \rho dV \\ * & \int(r_{ix}^2 + r_{iz}^2) \rho dV & -\int r_{iy} r_{iz} \rho dV \\ * & * & \int(r_{ix}^2 + r_{iy}^2) \rho dV \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_{l_i,xx} & -\mathbf{I}_{l_i,xy} & -\mathbf{I}_{l_i,xz} \\ * & \mathbf{I}_{l_i,yy} & -\mathbf{I}_{l_i,yz} \\ * & * & \mathbf{I}_{l_i,zz} \end{bmatrix}, \end{aligned} \quad (5.20)$$

predstavlja tenzor vztrajnosti za segment  $i$  glede na koordinatni sistem  $l_i$ . Posebej velja poudariti, da je pozicija segmenta  $i$  odvisna od trenutne lege manipulatorja in zato je tenzor vztrajnosti, če bi ga izrazili v osnovnem koordinatnem sistemu, tudi odvisen od lege. Če je kotna hitrost za segment  $i$  izražena glede na koordinatni sistem fiksiran na segmentu (kot pri D-H notaciji), potem je

$$\boldsymbol{\omega}_i^i = \mathbf{R}_i^T \boldsymbol{\omega}_i^0. \quad (5.21)$$

$\mathbf{R}_i$  je rotacijska matrika iz koordinatnega sistema na segmentu  $i$  v osnovni k. s.. Zapis  $\boldsymbol{\omega}_i^i$  predstavlja kotno hitrost koordinatnega sistema  $i$  izraženega v k.s.  $i$ , medtem ko npr.  $\boldsymbol{\omega}_i^0$  predstavlja isto veličino izraženo glede na osnovni k.s.. Glede na k.s. segmenta je tenzor vztrajnosti konstanten. Naj  $\mathbf{I}_{l_i}^i$  predstavlja tak tenzor. Potem je mogoče preveriti naslednjo relacijo

$$\mathbf{I}_{l_i} = \mathbf{R}_i \cdot \mathbf{I}_{l_i}^i \cdot \mathbf{R}_i^T, \quad (5.22)$$

$$\mathbf{I}_0 = \mathbf{R}_0^i \cdot \mathbf{I}_i \cdot {\mathbf{R}_0^i}^T. \quad (5.23)$$

$\mathbf{R}_0^i$  je rotacijska matrika, ki transformira iz relativnega k.s. (segmenta) v osnovni k.s.. Ker  $\mathbf{R}_0^i$  časovno ni konstantna vidimo, da se tudi  $\mathbf{I}_0$  glede na enačbo spreminja s časom.

Če os segmenta  $i$  sovpada z glavno osjo vztrajnosti, potem so produkti vztrajnosti enaki nič, kar pomeni, da postane tenzor vztrajnosti glede na težišče diagonalna matrika.

Po seštevanju translacijskih in rotacijskih prispevkov (5.15) in (5.19), po upoštevanju (5.22) je skupna kinetična energija segmenta  $i$  enaka

$$T_{l_i} = \frac{1}{2}m_{l_i} \dot{\mathbf{p}}_{l_i}^T \dot{\mathbf{p}}_{l_i} + \frac{1}{2}\boldsymbol{\omega}_i^T \mathbf{R}_i \mathbf{I}_{l_i}^i \mathbf{R}_i^T \boldsymbol{\omega}_i. \quad (5.24)$$

Na tem mestu je potrebno izraziti kinetično energijo sistema kot funkcijo generaliziranih koordinat, torej spremenljivk v sklepih. V ta namen je mogoče uporabiti geometrijsko metodo izračuna Jacobijeve matrike  $\mathbf{J}$  za vse vmesne segmente razen za vrh

$$\dot{\mathbf{p}}_{l_i} = \mathbf{J}_{P_1}^{(l_i)} \dot{q}_1 + \dots + \mathbf{J}_{P_i}^{(l_i)} \dot{q}_i = \mathbf{J}_P^{(l_i)} \dot{\mathbf{q}}, \quad (5.25)$$

$$\boldsymbol{\omega}_i^0 = \mathbf{J}_{O_1}^{(l_i)} \dot{q}_1 + \dots + \mathbf{J}_{O_i}^{(l_i)} \dot{q}_i = \mathbf{J}_O^{(l_i)} \dot{\mathbf{q}}, \quad (5.26)$$

kjer so upoštevani vsi prispevki Jacobijevih členov za posamezne hitrosti po sklepih do trenutnega segmenta  $i$ . Za pozicijo in orientacijo torej upoštevamo Jacobijeve člene

$$\mathbf{J}_P^{(l_i)} = [\mathbf{J}_{P_1}^{(l_i)} \dots \mathbf{J}_{P_i}^{(l_i)}], \quad (5.27)$$

$$\mathbf{J}_O^{(l_i)} = [\mathbf{J}_{O_1}^{(l_i)} \dots \mathbf{J}_{O_i}^{(l_i)}]. \quad (5.28)$$

Stolpce v matrikah (5.27) in (5.28) je mogoče izračunati podobno, kot smo jih obravnavali v diferencialni kinematiki

$$\mathbf{J}_{P_j}^{(l_i)} = \begin{cases} \mathbf{z}_{j-1} \\ \mathbf{z}_{j-1} \times (\mathbf{p}_{l_i} - \mathbf{p}_{j-1}) \end{cases}, \quad (5.29)$$

za translacijski in rotacijski sklep in

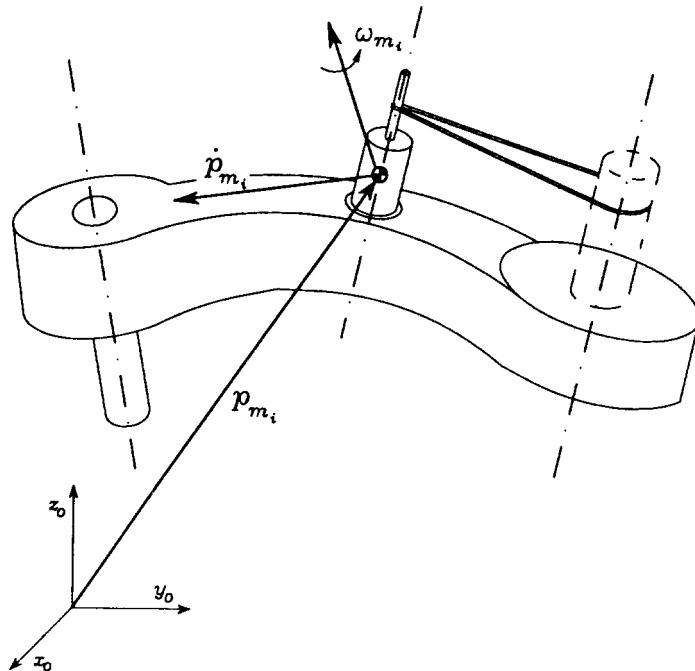
$$\mathbf{J}_{O_j}^{(l_i)} = \begin{cases} 0 \\ \mathbf{z}_{j-1} \end{cases} \quad (5.30)$$

zopet za translacijski in rotacijski sklep.  $\mathbf{p}_{j-1}$  je pozicijski vektor za izhodišče segmenta  $j-1$ ,  $\mathbf{z}_{j-1}$  pa je enotski vektor za os z koordinatnega sistema  $j-1$ .

Na osnovi (5.24) je sedaj mogoče zapisati kinetično energijo za segment  $i$  kot

$$T_{l_i} = \frac{1}{2}m_{l_i} \dot{\mathbf{q}}^T \mathbf{J}_P^{(l_i)T} \mathbf{J}_P^{(l_i)} \dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{J}_O^{(l_i)T} \mathbf{R}_i \mathbf{I}_{l_i}^i \mathbf{R}_i^T \mathbf{J}_O^{(l_i)} \dot{\mathbf{q}}. \quad (5.31)$$

Prispevek kinetične energije zaradi motorja v segmentu  $i$  je mogoče določiti na formalno enak način kot za segment. Zamislimo si primer klasičnega električnega rotacijskega motorja, ki lahko poganja preko ustreznih prenosov



Slika 5.3: Predstavitev kinematike povezane z motorjem  $i$ .

rotacijske ali translacijske prostostne stopnje. Prispevek fiksnega dela motorja (statorja) lahko pripisemo tistemu segmentu, kjer je pričvrščen motor, torej je potrebno upoštevati še prispevek rotorja.

Glede na razmere prikazane na sliki 5.3 je motor segmenta  $i$  pričvrščen na segmentu  $i - 1$ . V praksi poskušamo pri načrtovanju mehanske strukture za manipulator z odprto kinematično verigo namestiti motorje čim bližje bazi manipulatorja in s tem zmanjšati dinamično breme v prvih segmentih verige. Pogonske navore posredujejo motorji preko zobniških mehanskih prenosov. Prispevek zobnikov h kinetični energiji je mogoče upoštevati skupaj z motorjem. Pri tem običajno predpostavimo, da ni vsiljenega gibanja, torej npr. gibanje segmenta  $i$  ne povzroči gibanja v drugih segmentih.

Kinetično energijo za rotor  $i$  je mogoče zapisati kot

$$T_{m_i} = \frac{1}{2} m_{m_i} \dot{\mathbf{p}}_{m_i}^T \dot{\mathbf{p}}_{m_i} + \frac{1}{2} \boldsymbol{\omega}_{m_i}^T \mathbf{I}_{m_i} \boldsymbol{\omega}_{m_i}, \quad (5.32)$$

kjer je  $m_{m_i}$  masa rotorja,  $\dot{\mathbf{p}}_{m_i}$  linearna hitrost težišča motorja,  $\mathbf{I}_{m_i}$  je vztrajnostni tenzor rotorja podan glede na njegovo težišče in  $\boldsymbol{\omega}_{m_i}$  kotna hitrost rotorja.

Naj  $\vartheta_{mi}$  predstavlja kot rotorja. Ob predpostavki, da gre za tog zobniški prenos, lahko zapišemo

$$k_{ri}\dot{q}_i = \dot{\vartheta}_{mi}, \quad (5.33)$$

kjer je  $k_{ri}$  prestavno razmerje zobnikov. V primeru translacijskega pogona predstavlja prestavno razmerje tudi spremembo rotacije v translacijo.

Glede na način transformacije kotne hitrosti in relacijo (5.33) je skupna kotna hitrost rotorja enaka

$$\boldsymbol{\omega}_{mi} = \boldsymbol{\omega}_{i-1} + k_{ri} \dot{q}_i \boldsymbol{z}_{mi}, \quad (5.34)$$

kjer je  $\boldsymbol{\omega}_{i-1}$  kotna hitrost segmenta  $i - 1$  na katerem je motor pritrjen.  $\boldsymbol{z}_{mi}$  predstavlja enotski vektor usmerjen vzdolž rotacijske osi.

Da bi lahko izrazili kinetično energijo rotorja kot funkcijo spremenljivke v sklepnu moramo zapisati linearno hitrost rotorjevega težišča, podobno kot v (5.25) v obliki

$$\dot{\boldsymbol{p}}_{mi} = \boldsymbol{J}_P^{(mi)} \dot{\boldsymbol{q}}. \quad (5.35)$$

Tedaj je Jacobijeva matrika enaka

$$\boldsymbol{J}_P^{(mi)} = [\boldsymbol{J}_{P_1}^{(mi)} \dots \boldsymbol{J}_{P_{i-1}}^{(mi)} \quad 0 \dots 0], \quad (5.36)$$

kjer so stolpci podani kot

$$\boldsymbol{J}_{P_j}^{(mi)} = \begin{cases} \boldsymbol{z}_{j-1} \\ \boldsymbol{z}_{j-1} \times (\boldsymbol{p}_{mi} - \boldsymbol{p}_{j-1}) \end{cases}, \quad (5.37)$$

za translacijsko (v prvi vrsti) in rotacijsko (v drugi vrsti) prostostno stopnjo.  $\boldsymbol{p}_{j-1}$  je pozicijski vektor za izhodišče koordinatnega sistema  $j - 1$ . V enačbi (5.25) lahko upoštevamo  $\boldsymbol{J}_{P_i}^{(mi)} = 0$ , saj je težišče rotorja na osi rotacije.

Kotno hitrost v (5.34) je možno zapisati kot funkcijo spremenljivk v sklepnih, torej

$$\boldsymbol{\omega}_{mi} = \boldsymbol{J}_O^{(mi)} \dot{\boldsymbol{q}}. \quad (5.38)$$

Potem je potrebno izračunati člene Jacobijeve matrike

$$\boldsymbol{J}_O^{(mi)} = [\boldsymbol{J}_{O_1}^{(mi)} \dots \boldsymbol{J}_{O_{i-1}}^{(mi)} \quad \boldsymbol{J}_{O_i}^{(mi)}], \quad (5.39)$$

kjer so stolpci, v skladu z (5.34) in (5.26), podani kot

$$\boldsymbol{J}_{O_j}^{(mi)} = \begin{cases} \boldsymbol{J}_{O_j}^{(l_i)} & j = 1, \dots, i-1 \\ k_{ri} \boldsymbol{z}_{mi} & j = i \end{cases}, \quad (5.40)$$

Pri izračunu druge vrstice v (5.39) je dovolj poznavanje komponent enotskega vektorja za rotorjevo rotacijsko os  $\mathbf{z}_{m_i}$ , podanega glede na osnovni koordinatni sistem. Torej lahko kinetično energijo za rotor  $i$  zapišemo v obliki

$$T_{m_i} = \frac{1}{2} m_{m_i} \dot{\mathbf{q}}^T \mathbf{J}_P^{(m_i)T} \mathbf{J}_P^{(m_i)} \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{J}_O^{(m_i)T} \mathbf{R}_{m_i} \mathbf{I}_{m_i}^{m_i} \mathbf{R}_{m_i}^T \mathbf{J}_O^{(m_i)} \dot{\mathbf{q}}. \quad (5.41)$$

Končno lahko v skladu z en. (5.10) seštejemo posamezne prispevke podane glede na posamezne segmente (5.31) in posamezne rotorje (5.41). Skupna kinetična energija manipulatorja s pogoni je podana s kvadratično formo

$$T = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n b_{ij}(\mathbf{q}) \dot{\mathbf{q}}_i \dot{\mathbf{q}}_j = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}}, \quad (5.42)$$

kjer je  $\mathbf{B}(\mathbf{q})$  matrika vztrajnosti reda  $n \times n$  podana kot

$$\begin{aligned} \mathbf{B}(\mathbf{q}) = \sum_{i=1}^n & \left( m_{l_i} \mathbf{J}_P^{(l_i)T} \mathbf{J}_P^{(l_i)} + \mathbf{J}_O^{(l_i)T} \mathbf{R}_i \mathbf{I}_{l_i}^i \mathbf{R}_i^T \mathbf{J}_O^{(l_i)} \right. \\ & \left. + m_{m_i} \mathbf{J}_P^{(m_i)T} \mathbf{J}_P^{(m_i)} + \mathbf{J}_O^{(m_i)T} \mathbf{R}_{m_i} \mathbf{I}_{m_i}^{m_i} \mathbf{R}_{m_i}^T \mathbf{J}_O^{(m_i)} \right). \end{aligned} \quad (5.43)$$

Matrika vztrajnosti je:

- simetrična,
- pozitivno definitna in
- (v splošnem) odvisna od konfiguracije.

### 5.1.2 Izračun potencialne energije

Podobno kot pri kinetični energiji, je tudi potencialna energija, shranjena v manipulatorju, podana kot vsota prispevkov za vsak segment kot tudi za vsak rotor

$$U = \sum_{i=1}^n (U_{l_i} + U_{m_i}). \quad (5.44)$$

Ob predpostavki, da imamo opravka s togimi segmenti, je prispevek samo zaradi gravitacijskih sil izražen kot

$$U_{l_i} = - \int_{V_{l_i}} \mathbf{g}_0^T \mathbf{p}_i^* \rho dV = -m_{l_i} \mathbf{g}_0^T \mathbf{p}_{l_i}, \quad (5.45)$$

kjer je  $\mathbf{g}_0$  vektor gravitacijskega pospeška v osnovnem koordinatnem sistemu (npr.  $\mathbf{g}_0 = [0, 0, -g]^T$ , če je z os usmerjena vertikalno). Hkrati ob prehodu z leve na desno stran upoštevamo tudi (5.13)

$$\mathbf{p}_{l_i} = \frac{1}{m_{l_i}} \int_{V_{l_i}} \mathbf{p}_i^* \rho dV$$

za izračun težišča segmenta  $i$ . Prispevek rotorja  $i$  je, s podobnim razmišljajem kot pri (5.45), enak

$$U_{m_i} = -m_{m_i} \mathbf{g}_0^T \mathbf{p}_{m_i}. \quad (5.46)$$

Enačbi (5.45) in (5.46) je mogoče vstaviti v (5.44), s čimer je potencialna energija podana kot

$$U = - \sum_{i=1}^n \left( m_{l_i} \mathbf{g}_0^T \mathbf{p}_{l_i} + m_{m_i} \mathbf{g}_0^T \mathbf{p}_{m_i} \right). \quad (5.47)$$

Tu je pomembno, da je potencialna energija odvisna samo od vektorjev  $\mathbf{p}_{l_i}$  in  $\mathbf{p}_{m_i}$ , ki sta funkcija le spremenljivk v sklepih  $\mathbf{q}$  in ne tudi hitrosti v sklepih  $\dot{\mathbf{q}}$ .

### 5.1.3 Enačbe gibanja

Po izračunu skupne kinetične in potencialne energije sistema, ki sta zapisani v enačbah (5.42) in (5.47), je mogoče Lagrangeovo funkcijo (5.1) za manipulator podati v obliki

$$\begin{aligned} L(\mathbf{q}, \dot{\mathbf{q}}) &= T(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}) \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n b_{ij}(\mathbf{q}) \dot{\mathbf{q}}_i \dot{\mathbf{q}}_j + \sum_{j=1}^n \left( m_{l_i} \mathbf{g}_0^T \mathbf{p}_{l_i}(\mathbf{q}) + m_{m_i} \mathbf{g}_0^T \mathbf{p}_{m_i}(\mathbf{q}) \right). \end{aligned} \quad (5.48)$$

Sedaj odvajamo, kot zahtevajo Lagrangeove enačbe (5.2). Pri odvajanju upoštevamo neodvisnost  $U$  od  $\dot{\mathbf{q}}$ .

Sedaj pišemo samo za en segment, zato ni vsote po  $i$ .

$$\frac{d}{dt} \cdot \frac{\partial L}{\partial \dot{\lambda}_i} - \frac{\partial L}{\partial \lambda_i} = \xi_i, \quad (5.49)$$

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}_i} \right) &= \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}_i} \right) = \sum_{j=1}^n b_{ij}(\mathbf{q}) \ddot{\mathbf{q}}_j + \sum_{j=1}^n b_{ij} \frac{db_{ij}(\mathbf{q})}{dt} \dot{\mathbf{q}}_j \\ &= \sum_{j=1}^n b_{ij}(\mathbf{q}) \ddot{\mathbf{q}}_j + \sum_{j=1}^n \sum_{k=1}^n \frac{\partial b_{ij}(\mathbf{q})}{\partial \mathbf{q}_k} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j \end{aligned} \quad (5.50)$$

in

$$\frac{\partial T}{\partial \mathbf{q}_i} = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial b_{jk}(\mathbf{q})}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j. \quad (5.51)$$

Indeksi pri seštevanju so zamenjani. Glede na transformacijo hitrosti iz notranjih v zunanje koordinate (en. (5.25) in (5.35)) je odvod potencialne energije enak

$$\begin{aligned} \frac{\partial U}{\partial \mathbf{q}_i} &= - \sum_{j=1}^n \left( m_{l_j} \mathbf{g}_0^T \frac{\partial \mathbf{p}_{l_j}}{\partial \mathbf{q}_i} + m_{m_j} \mathbf{g}_0^T \frac{\partial \mathbf{p}_{m_j}}{\partial \mathbf{q}_i} \right) \\ &= - \sum_{j=1}^n \left( m_{l_j} \mathbf{g}_0^T \mathbf{J}_{P_i}^{(l_j)}(\mathbf{q}) + m_{m_j} \mathbf{g}_0^T \mathbf{J}_{P_i}^{(m_j)}(\mathbf{q}) \right) = g_i(\mathbf{q}). \end{aligned} \quad (5.52)$$

Zopet so indeksi seštevanja zamenjani. Posledično so enačbe gibanja

$$\sum_{j=1}^n b_{ij}(\mathbf{q}) \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk}(\mathbf{q}) \dot{q}_k \dot{q}_j + g_i(\mathbf{q}) = \xi_i \quad i = 1, \dots, n, \quad (5.53)$$

kjer velja

$$h_{ijk} = \frac{\partial b_{ij}}{\partial \mathbf{q}_k} - \frac{1}{2} \frac{\partial b_{jk}}{\partial \mathbf{q}_i}. \quad (5.54)$$

Fizikalna razlaga (5.53) razkriva:

- za člene s pospeškom:

Koeficienti  $b_{ii}$  predstavljajo vztrajnostni moment v osi sklepa  $i$  v trenutni konfiguraciji manipulatorja, če so ostali sklepi zaklenjeni. Koeficient  $b_{ij}$  popisuje vpliv pospeška sklepa  $j$  na sklep  $i$ .

- za kvadratne hitrostne člene:

Izraz  $h_{ijj} \dot{q}_j^2$  predstavlja centrifugalni člen nastajajoč v sklepu  $i$  zaradi hitrosti sklepa  $j$ . Pomni, da je  $h_{iii} = 0$ , ker je  $\partial b_{ii}/\partial \mathbf{q}_i = 0$ .

Izraz  $h_{ijk} \dot{q}_k \dot{q}_j$  predstavlja Coriolisov člen nastajajoč v sklepu  $i$  zaradi hitrosti sklepov  $j$  in  $k$ .

- za člene odvisne od lege:

Spremenljivka  $\mathbf{g}_i$  predstavlja moment proizveden v sklepu  $i$  manipulatorja v trenutni legi zaradi prisotnosti gravitacije.

Nekatere dinamične povezave med sklepi, npr.  $b_{ij}$  in  $h_{ijj}$  je mogoče zmanjšati ali odpraviti pri načrtovanju strukture, s čimer je mogoče poenostaviti

vodenje.

Nekonzervativne sile, ki opravljajo delo v sklepih manipulatorja so podane z razliko med aktuatorskimi navori  $\boldsymbol{\tau}$  in navori zaradi viskoznega trenja  $\mathbf{F}_v \dot{\mathbf{q}}$  in navori statičnega trenja  $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}})$ .  $\mathbf{F}_v$  predstavlja  $(n \times n)$  diagonalno matriko koeficientov viskoznega trenja. Kot poenostavljen model navorov statičnega trenja je mogoče upoštevati navore Coulombovega trenja  $\mathbf{F}_s \text{sgn}(\dot{\mathbf{q}})$ , kjer je  $\mathbf{F}_s (n \times n)$  diagonalna matrika,  $\text{sgn}(\dot{\mathbf{q}})$  pa ponazarja  $(n \times 1)$  vektor s komponentami podanimi kot signum funkcije posameznih hitrosti v sklepih.

Če sta manipulator in njegov vrh v dotiku z okoljem je del pogonskih navorov uporabljen za uravnoveženje navorov nastajajočih v sklepih zaradi kontaktnih sil. Glede na splošno transformacijo sil in navorov so takšni navori podani kot  $\mathbf{J}^T(\mathbf{q})\mathbf{h}$ , z  $\mathbf{h}$  kot vektorjem sil in momentov deluječih med vrhom manipulatorja in okolico.

Končno je mogoče enačbe gibanja v skladu z (5.53) zapisati v kompaktni matrični obliki, ki predstavlja **dinamični model manipulatorja v notranjih koordinatah**

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{F}_s \text{sgn}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{h}. \quad (5.55)$$

$\mathbf{C}$  je primerna  $(n \times n)$  diagonalna matrika, tako da koeficienti ustrezano enačbi

$$\sum_{j=1}^n c_{ij} \dot{\mathbf{q}}_j = \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j. \quad (5.56)$$

Če manipulatorjeva struktura vsebuje zaprto verigo, je prikladno najprej izračunati ustrezen dinamični model za manipulator z odprto kinematično verigo. Zatem je smiselno izračunati ustrezne navore v pogonskih sklepih. Torej je mogoče enačbe gibanja oblikovati v obliki podobni (5.55), kjer je  $\mathbf{q} = \mathbf{q}_a$  izračunani vektor generaliziranih koordinat.

## 5.2 POMEMBNE LASTNOSTI DINAMIČNEGA MODELA

Dinamični model robotskega manipulatorja lahko v kompaktni matrični obliki zapišemo na naslednji način

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{F}_s \text{sgn}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{h}, \quad (5.57)$$

kjer je

- $\mathbf{B}$  ... vztrajnostna matrika,
- $\mathbf{C}$  ... matrika Coriolisovih ter centrifugalnih prispevkov,
- $\mathbf{F}_v$  ... matrika parametrov viskoznega trenja,
- $\mathbf{F}_s$  ... matrika parametrov statičnega trenja,
- $\mathbf{g}$  ... prispevek gravitacije,
- $\mathbf{J}$  ... Jacobijeva matrika,
- $\mathbf{q}$  ... vektor sklepnih spremenljivk zasuka oziroma razdalje,
- $\boldsymbol{\tau}$  ... vektor sklepnih momentov oziroma sil ter
- $\mathbf{h}$  ... vektor sil vrha manipulatorja v dotiku z okolico.

V nadaljevanju bomo pojasnili dve pomembni lastnosti dinamičnega modela, ki bosta uporabni pri identifikaciji dinamičnih parametrov modela kot tudi pri izvedbi regulacijskih algoritmov.

### 5.3 POŠEVNO-SIMETRIČNA MATRIKA $\dot{\mathbf{B}} - 2\mathbf{C}$

Izbira matrike  $\mathbf{C}$  ni enolična, saj obstaja več različnih matrik  $\mathbf{C}$ , katerih elementi zadoščajo pogoju naslednje enačbe

$$\sum_{j=1}^n c_{ij} \dot{\mathbf{q}}_j = \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j. \quad (5.58)$$

Posebno rešitev je mogoče dobiti iz členov na desni strani enačbe z upoštevanjem koeficientov, kot so zapisani v en. (5.54). Torej je

$$\begin{aligned} \sum_{j=1}^n c_{ij} \dot{\mathbf{q}}_j &= \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j \\ &= \sum_{j=1}^n \sum_{k=1}^n \left( \frac{\partial b_{ij}}{\partial \mathbf{q}_k} - \frac{1}{2} \frac{\partial b_{jk}}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j \right). \end{aligned} \quad (5.59)$$

Z razstavljivo prvega člena na desni strani enačbe ( $1/2 + 1/2$ ) in zamenjavo sumacijskih indeksov  $j$  in  $k$  dobimo

$$\sum_{j=1}^n c_{ij} \dot{\mathbf{q}}_j = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial b_{ij}}{\partial \mathbf{q}_k} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j + \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \left( \frac{\partial b_{ik}}{\partial \mathbf{q}_j} - \frac{\partial b_{jk}}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_j \right), \quad (5.60)$$

iz česar sledi splošen izraz za elemente matrike  $\mathbf{C}$

$$c_{ij} = \sum_{k=1}^n c_{ijk} \dot{\mathbf{q}}_k, \quad (5.61)$$

kjer so

$$c_{ijk} = \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial \mathbf{q}_k} + \frac{\partial b_{ik}}{\partial \mathbf{q}_j} - \frac{\partial b_{jk}}{\partial \mathbf{q}_i} \right) \quad (5.62)$$

Christoffelovi simboli drugega tipa.

Glede na simetrijo matrike vztrajnosti  $\mathbf{B}$ , velja

$$c_{ijk} = c_{ikj}. \quad (5.63)$$

Če matriko  $\mathbf{C}$  določimo na zgoraj opisani način, to pomeni naslednjo pomembno lastnost dinamičnega modela predstavljenega z (5.55). Matrika

$$\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{B}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \quad (5.64)$$

je poševno-simetrična matrika. To pomeni, da za poljuben vektor  $\mathbf{w}$  dimenzij  $(n \times 1)$ , velja naslednja relacija

$$\mathbf{w}^T \cdot \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \cdot \mathbf{w} = 0. \quad (5.65)$$

Dokaz: S substitucijo koeficienta (5.62) v (5.61) dobimo

$$\begin{aligned} c_{ij} &= \frac{1}{2} \sum_{k=1}^n \frac{\partial b_{ij}}{\partial \mathbf{q}_k} \dot{\mathbf{q}}_k + \frac{1}{2} \sum_{k=1}^n \left( \frac{\partial b_{ik}}{\partial \mathbf{q}_j} - \frac{\partial b_{jk}}{\partial \mathbf{q}_i} \right) \dot{\mathbf{q}}_k \\ &= \frac{1}{2} \dot{b}_{ij} + \frac{1}{2} \sum_{k=1}^n \left( \frac{\partial b_{ik}}{\partial \mathbf{q}_j} - \frac{\partial b_{jk}}{\partial \mathbf{q}_i} \right) \dot{\mathbf{q}}_k. \end{aligned} \quad (5.66)$$

Torej je izraz za splošen element matrike  $\mathbf{N}$  iz (5.64) enak

$$n_{ij} = \dot{b}_{ij} - 2c_{ij} = \sum_{k=1}^n \left( \frac{\partial b_{ik}}{\partial \mathbf{q}_j} - \frac{\partial b_{jk}}{\partial \mathbf{q}_i} \right) \dot{\mathbf{q}}_k. \quad (5.67)$$

Iz zgornjega rezultata (iz poševne simetričnosti) sledi naslednja enakost

$$n_{ij} = -n_{ji}. \quad (5.68)$$

Pomembna ugotovitev, ki sledi neposredno iz lastnosti poševne- simetričnosti matrike  $\mathbf{N}$  je pri enakosti  $\mathbf{w} = \dot{\mathbf{q}}$

$$\dot{\mathbf{q}}^T \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = 0. \quad (5.69)$$

V (5.65) smo zamenjali  $\mathbf{w} = \dot{\mathbf{q}}$ . Obratno ne velja, torej en. (5.69) ni enaka en. (5.65), saj je  $\mathbf{N}$  tudi funkcija  $\dot{\mathbf{q}}$ .

Pokazati je mogoče, da en. (5.69) velja za katerokoli izbiro matrike  $\mathbf{C}$ , saj je rezultat načela o ohranitvi energije (Hamiltonov princip). Glede na to načelo, je popolni odvod kinetične energije po času uravnotežen z močjo, ki jo generirajo vse sile, ki delujejo na sklepe manipulatorja. Za mehanski sistem, ki je predmet proučevanja lahko torej zapišemo

$$\frac{1}{2} \frac{d}{dt} (\dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}}) = \dot{\mathbf{q}}^T (\boldsymbol{\tau} - \mathbf{F}_v \dot{\mathbf{q}} - \mathbf{F}_s \operatorname{sgn}(\dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) - \mathbf{J}^T(\mathbf{q}) \mathbf{h}). \quad (5.70)$$

Po odvajjanju leve strani en. (5.70) po času dobimo

$$\frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} \quad (5.71)$$

kar po vstavitevi izraza za  $\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}}$  iz en. (5.55) pomeni

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} (\dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}}) &= \frac{1}{2} \dot{\mathbf{q}}^T (\dot{\mathbf{B}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \dot{\mathbf{q}} \\ &\quad + \dot{\mathbf{q}}^T (\boldsymbol{\tau} - \mathbf{F}_v \dot{\mathbf{q}} - \mathbf{F}_s \operatorname{sgn}(\dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) - \mathbf{J}^T(\mathbf{q}) \mathbf{h}). \end{aligned} \quad (5.72)$$

Direktna primerjava desnih strani en. (5.70) in (5.72) vodi do rezultata določenega z en. (5.69). Torej mora biti  $\dot{\mathbf{q}}^T \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = 0$ .

Če povzamemo, relacija (5.69) velja za katerokoli izbiro matrike  $\mathbf{C}$ , saj je neposredna posledica fizikalnih lastnosti sistema, medtem ko en. (5.65) velja le za posebno izbiro elementov matrike  $\mathbf{C}$  določenih z en. (5.64) in (5.62).

## 5.4 LINEARNOST GLEDE NA DINAMIČNE PARAMETRE

Pomembna lastnost dinamičnega modela je linearost dinamičnih parametrov, ki opisujejo segmente manipulatorja ter motorje.

Da bi določili takšne parametre, je smiselno povezati kinetično in potencialno energijo vsakega rotorja s kinetično in potencialno energijo segmenta na katerem je ta rotor lociran. Torej, z upoštevanjem združitve segmenta  $i$  ter rotorja  $i+1$  (razširjeni segment  $i$ ) je prispevek kinetične energije dan kot

$$T_i = T_{l_i} + T_{m_{i+1}}, \quad (5.73)$$

kjer je

$$T_{l_i} = \frac{1}{2} m_{l_i} \dot{\mathbf{p}}_{l_i}^T \dot{\mathbf{p}}_{l_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_{l_i} \boldsymbol{\omega}_i, \quad (5.74)$$

in

$$T_{m_{i+1}} = \frac{1}{2} m_{i+1} \dot{\mathbf{p}}_{m_{i+1}}^T \dot{\mathbf{p}}_{m_{i+1}} + \frac{1}{2} \boldsymbol{\omega}_{m_{i+1}}^T \mathbf{I}_{m_{i+1}} \boldsymbol{\omega}_{m_{i+1}}. \quad (5.75)$$

Glede na težišče razširjenega segmenta lahko translacijske hitrosti segmenta in rotorja izrazimo kot

$$\dot{\mathbf{p}}_{l_i} = \dot{\mathbf{p}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{C_i, l_i}. \quad (5.76)$$

Vektorski produkt nadomestimo z množenjem z antisimetrično matriko

$$\dot{\mathbf{p}}_{l_i} = \dot{\mathbf{p}}_{C_i} + \mathbf{S}(\boldsymbol{\omega}_i) \mathbf{r}_{C_i, l_i}.$$

Za motor velja

$$\dot{\mathbf{p}}_{m_{i+1}} = \dot{\mathbf{p}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{C_i, m_{i+1}}. \quad (5.77)$$

pri

$$\mathbf{r}_{C_i, l_i} = \mathbf{p}_{l_i} - \mathbf{p}_{C_i} \quad (5.78)$$

$$\mathbf{r}_{C_i, m_{i+1}} = \mathbf{p}_{m_{i+1}} - \mathbf{p}_{C_i}, \quad (5.79)$$

kjer je  $\mathbf{p}_{C_i}$  vektor težišča razširjenega segmenta  $i$ .

Po vstavivti en. (5.76) v (5.74) dobimo

$$\begin{aligned} T_{l_i} &= \frac{1}{2} m_{l_i} \dot{\mathbf{p}}_{C_i}^T \dot{\mathbf{p}}_{C_i} + \dot{\mathbf{p}}_{C_i}^T \mathbf{S}(\boldsymbol{\omega}_i) m_{l_i} \mathbf{r}_{C_i, l_i} \\ &\quad + \frac{1}{2} m_{l_i} \boldsymbol{\omega}_i^T \mathbf{S}^T(\mathbf{r}_{C_i, l_i}) \mathbf{S}(\mathbf{r}_{C_i, l_i}) \boldsymbol{\omega}_i + \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_{l_i} \boldsymbol{\omega}_i. \end{aligned} \quad (5.80)$$

Glede na Steinerjev izrek iz mehanike predstavlja matrika

$$\bar{\mathbf{I}}_{l_i} = \mathbf{I}_{l_i} + m_{l_i} \mathbf{S}^T(\mathbf{r}_{C_i, l_i}) \mathbf{S}(\mathbf{r}_{C_i, l_i}) \quad (5.81)$$

vztrajnostni tenzor glede na težišče razširjenega segmenta  $\mathbf{p}_{C_i}$ , ki vsebuje dodatni prispevek zaradi translacije izhodišča glede na katero je tenzor izračunan (kot v en. (5.78)). Zato rej, lahko en. (5.80) zapišemo kot

$$T_{l_i} = \frac{1}{2} m_{l_i} \dot{\mathbf{p}}_{C_i}^T \dot{\mathbf{p}}_{C_i} + \dot{\mathbf{p}}_{C_i}^T \mathbf{S}(\boldsymbol{\omega}_i) m_{l_i} \mathbf{r}_{C_i, l_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\mathbf{I}}_{l_i} \boldsymbol{\omega}_i. \quad (5.82)$$

Na enak način dobimo z vstavivijo en. (5.77) v en. (5.75) in upoštevanjem celotne kotne hitrosti rotorja (5.34)

$$\boldsymbol{\omega}_{m_i} = \boldsymbol{\omega}_{i-1} + k_{r_i} \dot{\mathbf{q}}_i \mathbf{z}_{m_i},$$

kjer je  $\boldsymbol{\omega}_{i-1}$  kotna hitrost segmenta  $i - 1$  na katerem se nahaja rotor in  $\mathbf{z}_{m_i}$  predstavlja enotski vektor vzdolž osi rotorja, sledičo enačbo

$$\begin{aligned} T_{m_{i+1}} &= \frac{1}{2} m_{m_{i+1}} \dot{\mathbf{p}}_{C_i}^T \dot{\mathbf{p}}_{C_i} + \dot{\mathbf{p}}_{C_i}^T \mathbf{S}(\boldsymbol{\omega}_i) m_{m_{i+1}} \mathbf{r}_{C_i, m_{i+1}} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\mathbf{I}}_{m_{i+1}} \boldsymbol{\omega}_i \\ &+ k_{r,i+1} \dot{\mathbf{q}}_{i+1} \mathbf{z}_{m_{i+1}}^T \mathbf{I}_{m_{i+1}} \boldsymbol{\omega}_i + \frac{1}{2} k_{r,i+1}^2 \dot{\mathbf{q}}_{i+1}^2 \mathbf{z}_{m_{i+1}}^T \mathbf{I}_{m_{i+1}} \mathbf{z}_{m_{i+1}}. \end{aligned} \quad (5.83)$$

Tu je

$$\bar{\mathbf{I}}_{m_{i+1}} = \mathbf{I}_{m_{i+1}} + m_{m_{i+1}} \mathbf{S}^T(\mathbf{r}_{C_i, m_{i+1}}) \mathbf{S}(\mathbf{r}_{C_i, m_{i+1}}). \quad (5.84)$$

Če seštejemo prispevke (5.82) in (5.83) enako kot v en. (5.73), dobimo izraz za kinetično energijo razširjenega segmenta  $i$  v sledeči obliki

$$\begin{aligned} T_i &= \frac{1}{2} m_i \dot{\mathbf{p}}_{C_i}^T \dot{\mathbf{p}}_{C_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\mathbf{I}}_i \boldsymbol{\omega}_i + k_{r,i+1} \dot{\mathbf{q}}_{i+1} \mathbf{z}_{m_{i+1}}^T \mathbf{I}_{m_{i+1}} \boldsymbol{\omega}_i \\ &+ \frac{1}{2} k_{r,i+1}^2 \dot{\mathbf{q}}_{i+1}^2 \mathbf{z}_{m_{i+1}}^T \mathbf{I}_{m_{i+1}} \mathbf{z}_{m_{i+1}}, \end{aligned} \quad (5.85)$$

kjer sta

$$m_i = m_{l_i} + m_{m_{i+1}} \text{ in } \bar{\mathbf{I}}_i = \bar{\mathbf{I}}_{l_i} + \bar{\mathbf{I}}_{m_{i+1}} \quad (5.86)$$

celotna masa oziroma vztrajnostni tenzor razširjenega segmenta. Pri izpeljavi en. (5.85) smo upoštevali en. (5.78) in (5.79), kot tudi sledečo povezavo med položaji težišč

$$m_{l_i} \mathbf{p}_{l_i} + m_{m_{i+1}} \mathbf{p}_{m_{i+1}} = m_i \mathbf{p}_{C_i}. \quad (5.87)$$

Opazimo lahko, da prva dva člena na desni strani en. (5.85) predstavljata prispevek kinetične energije rotorja, ko ta miruje, medtem ko ostala dva člena določata kinetično energijo zaradi vrtenja rotorja.

Ob predpostavki, da je masa rotorja simetrično porazdeljena okrog osi vrtenja, lahko zapišemo vztrajnostni tenzor izražen v koordinatnem sistemu  $\mathbf{R}_{m_i}$  z izhodiščem v težišču in osjo  $\mathbf{z}_{m_i}$  poravnano z osjo rotacije, kot

$$\mathbf{I}_{m_i}^{m_i} = \begin{bmatrix} I_{m_i xx} & 0 & 0 \\ 0 & I_{m_i yy} & 0 \\ 0 & 0 & I_{m_i zz} \end{bmatrix}, \quad (5.88)$$

kjer je  $I_{m_i xx} = I_{m_i yy}$ . To smemo narediti, ker je rotor okrogel in simetričen. Posledica je konstantna vrednost vztrajnostnega tenzorja glede na kakršnokoli rotacijo okrog osi  $\mathbf{z}_{m_i}$  in konstantnost glede na katerikoli koordinatni sistem pritrjen na segment  $i - 1$ .

Ker je naš cilj določiti nabor dinamičnih parametrov, ki naj bi bili neodvisni od konfiguracije manipulatorja, je smiselno izraziti vztrajnostni tenzor  $\bar{\mathbf{I}}_i$  v koordinatnem sistemu  $\mathbf{R}_i$  pritrjenem na segment in vztrajnostni tenzor  $\bar{\mathbf{I}}_{m_{i+1}}$  v koordinatnem sistemu  $\mathbf{R}_{m_{i+1}}$ , tako da je le ta diagonalen.

Glede na en. (5.88) dobimo

$$\mathbf{I}_{m_{i+1}} \mathbf{z}_{m_i} = \mathbf{R}_{m_{i+1}} \mathbf{I}_{m_{i+1}}^{m_{i+1}} \mathbf{R}_{m_{i+1}}^T \mathbf{z}_{m_i}, \quad (5.89)$$

kjer  $\mathbf{I}_{m_{i+1}} = \mathbf{I}_{m_{i+1}zz}$  določa konstantno skalarno vrednost vztrajnostnega momenta rotorja okrog osi rotacije.

Tako kinetična energija (5.31) postane

$$T_i = \frac{1}{2} m_i \dot{\mathbf{p}}_{C_i}^{iT} \dot{\mathbf{p}}_{C_i}^i + \frac{1}{2} \boldsymbol{\omega}_i^{iT} \bar{\mathbf{I}}_i^i \boldsymbol{\omega}_i^i + k_{r,i+1} \dot{\mathbf{q}}_{i+1} \mathbf{I}_{m_{i+1}} \mathbf{z}_{m_{i+1}}^{iT} \boldsymbol{\omega}_i^i + \frac{1}{2} k_{r,i+1}^2 \dot{\mathbf{q}}_{i+1}^2 \mathbf{I}_{m_{i+1}}. \quad (5.90)$$

Glede na linearno sestavljanje hitrosti za segment  $i$ , lahko zapišemo

$$\dot{\mathbf{p}}_{C_i}^i = \dot{\mathbf{p}}_i^i + \boldsymbol{\omega}_i^i \times \mathbf{r}_{i,C_i}^i, \quad (5.91)$$

kjer se vsi vektorji nanašajo na koordinatni sistem  $i$ . Vektor  $\times \mathbf{r}_{i,C_i}^i$  je podan v takšnem sistemu. Po vstavitevi (5.91) v (5.90) dobimo

$$\begin{aligned} T_i &= \frac{1}{2} m_i \dot{\mathbf{p}}_i^{iT} \dot{\mathbf{p}}_i^i + \dot{\mathbf{p}}_i^{iT} \mathbf{S}(\boldsymbol{\omega}_i^i) m_i \mathbf{r}_{i,C_i}^i + \frac{1}{2} \boldsymbol{\omega}_i^{iT} \widehat{\mathbf{I}}_i^i \boldsymbol{\omega}_i^i \\ &\quad + k_{r,i+1} \dot{\mathbf{q}}_{i+1} \mathbf{I}_{m_{i+1}} \mathbf{z}_{m_{i+1}}^{iT} \boldsymbol{\omega}_i^i + \frac{1}{2} k_{r,i+1}^2 \dot{\mathbf{q}}_{i+1}^2 \mathbf{I}_{m_{i+1}}, \end{aligned} \quad (5.92)$$

kjer

$$\widehat{\mathbf{I}}_i^i = \bar{\mathbf{I}}_i^i + m_i \mathbf{S}^T(\mathbf{r}_{i,C_i}^i) \mathbf{S}(\mathbf{r}_{i,C_i}^i) \quad (5.93)$$

predstavlja vztrajnostni tenzor glede na izhodišče koordinatnega sistema  $i$  ob upoštevanju Steinerjevega izreka.

Naj bo  $\mathbf{r}_{i,C_i}^i = [l_{C_ix}, l_{C_iy}, l_{C_iz}]^T$ . Prvi vztrajnostni moment je potem enak

$$m_i \mathbf{r}_{i,C_i}^i = \begin{bmatrix} m_i l_{C_ix} \\ m_i l_{C_iy} \\ m_i l_{C_iz} \end{bmatrix}. \quad (5.94)$$

Iz en. (5.93) dobimo vztrajnostni tenzor razširjenega segmenta i kot

$$\widehat{\boldsymbol{I}}_i^i = \begin{bmatrix} \overline{I}_{ixx} + m_i(l_{C_iy}^2 + l_{C_iz}^2) & -\overline{I}_{ixy} - m_i l_{C_ix} l_{C_iy} & -\overline{I}_{ixz} - m_i l_{C_ix} l_{C_iz} \\ * & \overline{I}_{iyy} + m_i(l_{C_ix}^2 + l_{C_iz}^2) & -\overline{I}_{iyz} - m_i l_{C_iy} l_{C_iz} \\ * & * & \overline{I}_{izz} + m_i(l_{C_ix}^2 + l_{C_iy}^2) \end{bmatrix} =$$

$$\begin{bmatrix} \widehat{I}_{ixx} & -\widehat{I}_{ixy} & -\widehat{I}_{ixz} \\ * & \widehat{I}_{iyy} & -\widehat{I}_{iyz} \\ * & * & \widehat{I}_{izz} \end{bmatrix}. \quad (5.95)$$

Torej je kinetična energija razširjenega segmenta linearna glede na sledeče dinamične parametre: maso, tri komponente prvega vztrajnostnega momenta (5.94), šest komponent vztrajnostnega tenzorja (5.95) ter vztrajnostni moment rotorja.

Pri določanju potencialne energije je smiselno izhajati iz težišča razširjenega segmenta  $i$ , določenega z en. (5.87), torej lahko zapišemo posamezni prispevek potencialne energije kot

$$U_i = -m_i \mathbf{g}_0^{iT} \boldsymbol{p}_{C_i}^i, \quad (5.96)$$

kjer se vektorji nanašajo na koordinatni sistem  $i$ . Glede na zapis

$$\boldsymbol{p}_{C_i}^i = \boldsymbol{p}_i^i + \boldsymbol{r}_{i,C_i}^i$$

lahko en. (5.96) zapišemo v obliki

$$U_i = -\mathbf{g}_0^{iT} (m_i \boldsymbol{p}_i^i + m_i \boldsymbol{r}_{i,C_i}^i), \quad (5.97)$$

kar pomeni, da je potencialna energija linearna glede na maso ter tri komponente prvega vztrajnostnega momenta (5.94).

S seštevkom kinetične in potencialne energije za vse razširjene segmente, lahko Lagrangevo enačbo sistema zapišemo v obliki

$$L = \sum_{i=1}^n (\boldsymbol{\beta}_{T_i}^T - \boldsymbol{\beta}_{U_i}^T) \boldsymbol{\pi}_i. \quad (5.98)$$

$\boldsymbol{\pi}_i$  je vektor dinamičnih parametrov dimenzij  $(11 \times 1)$

$$\boldsymbol{\pi}_i = \begin{bmatrix} m_i & m_i I_{C_ix} & m_i I_{C_iy} & m_i I_{C_iz} & \dots \\ \dots & \widehat{I}_{ixx} & \widehat{I}_{ixy} & \widehat{I}_{ixz} & \widehat{I}_{iyy} & \widehat{I}_{iyz} & \widehat{I}_{izz} & \boldsymbol{I}_{m_i} \end{bmatrix}^T \quad (5.99)$$

v katerem smo povezali vztrajnostni tenzor rotorja  $i$  s parametri segmenta  $i$ , s čimer smo poenostavili zapis.

V en. (5.99) sta  $\beta_{T_i}$  in  $\beta_{U_i}$  vektorja dimenzij ( $11 \times 1$ ), ki omogočata zapis Lagrangeove enačbe kot funkcije vektorja  $\pi_i$ . Takšni vektorji so funkcije posplošenih koordinat mehanskega sistema (in njihovih odvodov  $\beta_{T_i}$ ). Pokazati je mogoče, da vektorja

$$\begin{aligned}\beta_{T_i} &= \beta_{T_i}(q_1, q_2, \dots, q_i, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_i) \text{ in} \\ \beta_{U_i} &= \beta_{U_i}(q_1, q_2, \dots, q_i, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_i)\end{aligned}$$

nista odvisna od spremenljivk sklepov, ki sledijo segmentu  $i$  (torej  $i+1$ ).

Odvajanja, ki jih zahteva Lagrangeova enačba ne spremenijo lastnosti linearnosti parametrov, zato lahko posplošeno silo v sklepu  $i$  zapišemo kot

$$\xi_i = \sum_{j=1}^n y_{ij}^T \pi_j, \quad (5.100)$$

kjer je

$$y_{ij} = \frac{d}{dt} \frac{\partial \beta_{T_j}}{\partial \dot{q}_i} - \frac{\partial \beta_{T_j}}{\partial q_i} + \frac{\partial \beta_{U_j}}{\partial q_i}. \quad (5.101)$$

Ker parcialni odvodi funkcij  $\beta_{T_i}$  in  $\beta_{U_i}$ , ki nastopajo v en. (5.101) izginejo pri  $j < i$ , dobimo sledeč rezultat

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} y_{11}^T & y_{12}^T & \dots & y_{1p}^T \\ 0^T & y_{22}^T & \dots & y_{2p}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \dots & y_{np}^T \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_p \end{bmatrix}. \quad (5.102)$$

ki predstavlja lastnost linearnosti modela manipulatorja glede na primeren nabor dinamičnih parametrov.

V preprostem primeru brez dotika z okoljem ( $\mathbf{h} = 0$ ), je smiselno dodati koeficiente viskoznega trenja ( $\mathbf{F}_{vi}$ ) in Coulombovega statičnega trenja ( $\mathbf{F}_{si}$ ) v parametre vektorja  $\pi_i$ . S tem dobimo skupaj 13 parametrov za posamezen sklep. En. (5.102) lahko zapišemo v strnjeni obliki kot

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \boldsymbol{\pi}, \quad (5.103)$$

kjer je  $\boldsymbol{\pi}$  vektor konstantnih parametrov dimenzij ( $p \times 1$ ) in  $\mathbf{Y}$  matrika dimenzij ( $n \times p$ ), ki je funkcija položajev sklepov, hitrosti in pospeškov. Tako

matriko običajno imenujemo regresor. Dimenzija vektorja parametrov je  $p \leq 13n$ , saj ni nujno, da vseh trinajst parametrov sklepa eksplicitno nastopa v en. (5.103).

## 5.5 DINAMIČNI MODEL V ZUNANJIH KORDINATAH

Do sedaj smo obravnavali eno od možnih in tudi najbolj pogosto obliko zapisa dinamičnega modela manipulatorja v notranjih (sklepnih) koordinatah. Druga možnost predstavlja zapis modela manipulatorja direktno v zunanjih koordinatah. Da bi to lahko napravili je potrebno najti dinamični model, ki predstavlja relacijo med generaliziranimi silami delujočimi na manipulator in kinematičnimi spremenljivkami v zunanjih koordinatah.

Podobno kot kinematični opis manipulatorja v zunanjih koordinatah, kjer prisotnost redundantnih prostostnih stopenj in/ali kinematičnih singularnosti oteži obravnavo, tako je potreben skrben pristop tudi pri izpeljavi dinamičnega modela v zunanjih koordinatah.

Zapis dinamičnega modela s pomočjo Lagrange formulacije v zunanjih koordinatah dovoljuje celovit opis sistema gibanja samo v primeru neredundantnega manipulatorja, kjer spremenljivke zapisane v generalizirani obliki predstavljajo kinetično in potencialno energijo ter nekonzervativne sile, ki opravljajo delo. Takšen pristop ne nudi vedno rešitve pri redundantnem manipulatorju. Povsem normalno je pričakovati v tem primeru dodatno gibanje v sklepih (internal motions), ki pa ne povzroča gibanja vrha.

Izpeljavo dinamičnega modela v zunanjih koordinatah, veljavnega za redundantne in neredundantne manipulatorje je smiseln začeti kar iz modela zapisanega v notranjih koordinatah. Preprosto preoblikovanje znane enačbe nas pripelje do

$$\ddot{\mathbf{q}} = -\mathbf{B}^{-1}(\mathbf{q}) \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{B}^{-1}(\mathbf{q}) \mathbf{g}(\mathbf{q}) + \mathbf{B}^{-1}(\mathbf{q}) \mathbf{J}_G^T(\mathbf{q})(\boldsymbol{\gamma} - \mathbf{h}), \quad (5.104)$$

kjer je vektor  $\boldsymbol{\tau}$  zapisan v obliki ekvivalentnih sil in navorov  $\boldsymbol{\gamma}$ , ki jih povzroča na vrhu. V podanem zapisu predstavlja  $\mathbf{h}$  prispevek zaradi sil in navorov na vrhu konkretno zaradi dotika vrha z okolico,  $\boldsymbol{\gamma}$  pa torej tiste zaradi aktivnosti v sklepih.

Spomnimo se tudi na relacijo diferencialne kinematike

$$\dot{\mathbf{x}} = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}, \quad (5.105)$$

ki po odvodu preide v

$$\ddot{\mathbf{x}} = \mathbf{J}_A(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{J}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}. \quad (5.106)$$

Zapis za  $\ddot{\mathbf{q}}$  zgoraj vsebuje geometrijsko  $\mathbf{J}_G$  matriko, medtem ko se analitični  $\mathbf{J}_A$  pojavlja v  $\ddot{\mathbf{x}}$ . Da bi ta dva zapisa poenotili, se spomnimo na transformacijsko matriko  $\mathbf{T}_A$  med obema  $\mathbf{J}$  matrikama, kar pomeni tudi

$$\mathbf{T}_A^T(\mathbf{x})\boldsymbol{\gamma} = \boldsymbol{\gamma}_A \text{ in } \mathbf{T}_A^T(\mathbf{x})\mathbf{h} = \mathbf{h}_A. \quad (5.107)$$

Sedaj izraz za  $\ddot{\mathbf{q}}$  lahko vstavimo v tistega za  $\ddot{\mathbf{x}}$  in uporabimo tudi transformaciji za  $\boldsymbol{\gamma}$ ,  $\boldsymbol{\gamma}_A$ ,  $\mathbf{h}$ ,  $\mathbf{h}_A$

$$\ddot{\mathbf{x}} = -\mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \dot{\mathbf{q}} - \mathbf{J}_A \mathbf{B}^{-1} \mathbf{g} + \dot{\mathbf{J}}_A \dot{\mathbf{q}} + \mathbf{J}_A \mathbf{B}^{-1} \mathbf{J}_A^T(\mathbf{q})(\boldsymbol{\gamma}_A - \mathbf{h}_A). \quad (5.108)$$

Odvisnosti posameznih spremeljivk od  $\mathbf{q}$  in  $\dot{\mathbf{q}}$  v tej enačbi niso zapisane zaradi enostavnosti. Nadalje lahko vpeljemo nekaj okrajšav, npr.

$$\mathbf{B}_A = (\mathbf{J}_A \mathbf{B}^{-1} \mathbf{J}_A^T)^{-1}, \quad (5.109)$$

$$\mathbf{C}_A \dot{\mathbf{x}} = \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \dot{\mathbf{q}} - \mathbf{B}_A \dot{\mathbf{J}}_A \dot{\mathbf{q}}, \quad (5.110)$$

$$\mathbf{g}_A = \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{g}. \quad (5.111)$$

Z upoštevanjem teh okrajšav lahko sedaj zapišemo  $\ddot{\mathbf{x}}$  v obliki

$$\mathbf{B}_A(\mathbf{x}) \ddot{\mathbf{x}} + \mathbf{C}_A(\mathbf{x}, \dot{\mathbf{x}}) \dot{\mathbf{x}} + \mathbf{g}_A(\mathbf{x}) = (\boldsymbol{\gamma}_A - \mathbf{h}_A), \quad (5.112)$$

ki je po obliki enaka zapisu v notranjih koordinatah. Matriko  $\mathbf{J}_A \mathbf{B}^{-1} \mathbf{J}_A^T$  je mogoče invertirati samo in samo, če ima  $\mathbf{J}_A$  poln rang, torej v primeru brez kinematičnih singularnosti, lahko pa je manipulator redundanten.

Za neredundanten manipulator v nesingularni legi pa dobijo okrajšave obliko

$$\mathbf{B}_A = \mathbf{J}_A^{-T} \mathbf{B} \mathbf{J}_A^{-1}, \quad (5.113)$$

$$\mathbf{C}_A \dot{\mathbf{x}} = \mathbf{J}_A^{-T} \mathbf{C} \dot{\mathbf{q}} - \mathbf{B}_A \dot{\mathbf{J}}_A \dot{\mathbf{q}}, \quad (5.114)$$

$$\mathbf{g}_A = \mathbf{J}_A^{-T} \mathbf{g}. \quad (5.115)$$

Še enkrat povdarmo, da je glavna lastnost izpeljanega modela prav njegova veljavnost tudi za redundantni manipulator, čeprav spremenljivke  $\mathbf{x}$  ne sestavljajo generaliziranih koordinat sistema. V obravnavanem primeru predstavlja  $\mathbf{B}_A$  kinetično energijo.

V nadaljevanju je prikazana uporaba dinamičnega modela zapisanega v zunanjih koordinatah pri reševanju direktne in inverzne dinamike. Naslednja izpeljava je pomembna predvsem za redundantne manipulatorje, saj je za neredundantne manipulatorje že izpeljana enačba popolnoma ustrezna vse dokler je matrika  $\mathbf{J}_A$  nesingularna.

Podobno kot pri zapisu v notranjih koordinatah želimo tudi tukaj izračunati kinematične spremenljivke  $\ddot{\mathbf{x}}(t)$  ter seveda  $\dot{\mathbf{x}}(t)$  in  $\mathbf{x}(t)$  iz danih navorov v sklepih  $\boldsymbol{\tau}(t)$  in sil na vrhu  $\mathbf{h}(t)$ . Za redundantni manipulator ni mogoče direktno uporabiti enačbe

$$\mathbf{B}_A(\mathbf{x}) \ddot{\mathbf{x}} + \mathbf{C}_A(\mathbf{x}, \dot{\mathbf{x}}) \dot{\mathbf{x}} + \mathbf{g}_A(\mathbf{x}) = (\boldsymbol{\gamma}_A - \mathbf{h}_A), \quad (5.116)$$

saj je  $\boldsymbol{\gamma}_A$  mogoče izračunati samo preko  $\boldsymbol{\tau}$ , če obstaja  $\mathbf{J}^T$ . V primeru simulacij je mogoče izračunati rešitve v notranjih koordinatah kot  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$  ki preko enačb direktne kinematike vodijo do  $\mathbf{x}$ ,  $\dot{\mathbf{x}}$ ,  $\ddot{\mathbf{x}}$ .

Zapis inverzne dinamike v zunanjih koordinatah zahteva določitev navorov v sklepih  $\boldsymbol{\tau}(t)$ , ki so potrebni za generiranje gibanja v obliki  $\ddot{\mathbf{x}}(t)$ ,  $\dot{\mathbf{x}}(t)$ ,  $\mathbf{x}(t)$  za dane sile na vrhu  $\mathbf{h}(t)$ . Možna pot reševanja je z izračunom najprej inverzne kinematike do  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$  preko

$$\mathbf{x} = k(\mathbf{q}) \quad (5.117)$$

$$\dot{\mathbf{x}} = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}, \quad (5.118)$$

odvajanja

$$\ddot{\mathbf{x}} = \mathbf{J}_A(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}. \quad (5.119)$$

in zatem potrebnih navorov  $\boldsymbol{\tau}(t)$  preko

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{F}_s \operatorname{sgn}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{h}. \quad (5.120)$$

Za redundantne manipulatorje tako zaobidemo probleme redundantnosti preko izračuna kinematike.

Alternativni pristop  $k$  izračunu inverzne dinamike najprej določi  $\boldsymbol{\gamma}_A$  in zatem navore v sklepih  $\boldsymbol{\tau}$ . Pri tem načinu obstoječe prisotne redundantne prostostne stopnje sploh niso uporabljeni, saj izračunani navori ne generirajo notranjega gibanja v strukturi.

Če želimo dobiti formalno rešitev za redundantni problem v okviru dinamike, potem je najprej potrebno določiti sile vrha povzročene zaradi navorov v sklepih. To je podobno rešitvi v diferencialni kinematiki preko psevdoinverzne  $\mathbf{J}$  matrike

$$\dot{\mathbf{q}} = \mathbf{J}^\# \mathbf{v} + (\mathbf{I} - \mathbf{J}^\# \mathbf{J})\dot{\mathbf{q}}_0. \quad (5.121)$$

Ker je potrebno določiti navore v sklepih je prikladno izraziti model

$$\mathbf{B}_A(\mathbf{x}) \ddot{\mathbf{x}} + \mathbf{C}_A(\mathbf{x}, \dot{\mathbf{x}}) \dot{\mathbf{x}} + \mathbf{q}_A(\mathbf{x}) = (\boldsymbol{\gamma}_A - \mathbf{h}_A), \quad (5.122)$$

v odvisnosti od  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$ . Po upoštevanju izrazov za  $\mathbf{C}_A \dot{\mathbf{x}} = \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \dot{\mathbf{q}} - \mathbf{B}_A \dot{\mathbf{J}}_A \dot{\mathbf{q}}$  in  $\mathbf{g}_A = \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{g}$  dobri model obliko

$$\mathbf{B}_A(\mathbf{x}) \ddot{\mathbf{x}} + \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \dot{\mathbf{q}} - \mathbf{B}_A \dot{\mathbf{J}}_A \dot{\mathbf{q}} + \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{g} = (\boldsymbol{\gamma}_A - \mathbf{h}_A) \quad (5.123)$$

in po združitvi členov

$$\mathbf{B}_A(\ddot{\mathbf{x}} - \dot{\mathbf{J}}_A \dot{\mathbf{q}}) + \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \dot{\mathbf{q}} + \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{g} = (\boldsymbol{\gamma}_A - \mathbf{h}_A). \quad (5.124)$$

Po upoštevanju še  $\ddot{\mathbf{x}} = \mathbf{J}_A(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$  se dva člena medsebojno pokrajšata do

$$\mathbf{B}_A \mathbf{J}_A(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \dot{\mathbf{q}} + \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{g} = (\boldsymbol{\gamma}_A - \mathbf{h}_A). \quad (5.125)$$

Če si izberemo še  $\bar{\mathbf{J}}_A(\mathbf{q}) = \mathbf{B}^{-1}(\mathbf{q}) \mathbf{J}_A^T \mathbf{B}_A(\mathbf{q})$  dobimo

$$\bar{\mathbf{J}}_A^T(\mathbf{B}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}) = (\boldsymbol{\gamma}_A - \mathbf{h}_A). \quad (5.126)$$

Glede na  $\mathbf{B}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} = \boldsymbol{\tau} - \mathbf{J}^T \mathbf{h}$  pa lahko zapišemo tudi

$$\bar{\mathbf{J}}_A^T(\boldsymbol{\tau} - \mathbf{J}_A^T \mathbf{h}_A) = (\boldsymbol{\gamma}_A - \mathbf{h}_A) \quad (5.127)$$

in

$$\bar{\mathbf{J}}_A^T \boldsymbol{\tau} = \boldsymbol{\gamma}_A. \quad (5.128)$$

Splošna rešitev te enačbe je tako v obliki

$$\boldsymbol{\tau} = \mathbf{J}_A^T(\mathbf{q}) \boldsymbol{\gamma}_A + (\mathbf{I} - \mathbf{J}_A^T(\mathbf{q}) \bar{\mathbf{J}}_A^T(\mathbf{q})) \boldsymbol{\tau}_0. \quad (5.129)$$

Pri tem je  $\mathbf{J}_A^T$  desni psevdo inverz od matrike  $\bar{\mathbf{J}}_A$  obtežene z inverzno matriko vztrajnosti  $\mathbf{B}^{-1}$ . Poljubni vektor navorov  $\boldsymbol{\tau}_0$  dimenzij  $n \times 1$  ne prispeva k silam na vrhu, saj se projecira v ničti prostor  $\bar{\mathbf{J}}_A^T$ .

Na koncu lahko ugotovimo, da enačba (5.122) omogoča za dane  $\mathbf{x}$ ,  $\dot{\mathbf{x}}$ ,  $\ddot{\mathbf{x}}$  in  $\mathbf{h}_A(t)$  izračun  $\boldsymbol{\gamma}_A$ . Zatem preko enačbe (5.129) pridemo do navorov  $\boldsymbol{\tau}$ , ki poleg gibanja vrha povzroči tudi gibanje v sklepih redundantne strukture, za kar so potrebni navori  $\boldsymbol{\tau}_0$ .



# Poglavlje 6

## NEWTON-EULERJEVA DINAMIKA

### 6.1 UVOD

**Izpeljava dinamičnih enačb za poljuben serijski mehanizem z eno-prostostnimi rotacijskimi sklepi.**

Newton-Eulerjeva metoda vodi do popolnoma enakega rezultata, kot že znani Lagrangeov zapis, vendar je pot izpeljave bistveno drugačna. Pri Lagrangu gledamo na manipulator kot celoto in analiziramo stanje s pomočjo Lagrangeove funkcije (razlika med kinetično in potencialno energijo sistema). Za razliko od tega pristopa obravnavamo pri Newton-Eulerju vsak segment manipulatorja posebej in ob tem zapišemo enačbe njegovega translacijskega in rotacijskega gibanja. Ker je vsak segment povezan z drugimi, so upoštevanja vredne tudi sile in navori sosednih segmentov, ki se pojavijo v enačbah. S tako imenovano rekurzijo naprej/nazaj je mogoče določiti vse te vplive zaradi medsebojnega sklopa in na koncu popisati gibanje celega manipulatorja. Iz tega vidimo drugačnost Newton-Eulerjevega pristopa, ki se bistveno razlikuje od Lagrangeovega zapisa.

Na tem mestu se lahko upravičeno vprašamo, zakaj bi obravnavali še eno metodo, še en pristop k istemu problemu. Obe metodi sta se razvijali vzporedno, vsaka od njiju pa ima določene prednosti. Newton-Eulerjev zapis je na prvi pogled bolj primeren za rekurzivno izračunavanje ter časovno manj potraten, vendar sta si danes metodi enakovredni. Zato si jo oglejmo z namenom, da bi bolje razumeli delovanje dinamičnega sistema ob osvetlitvi z druge strani.

V vseh mehanskih sistemih lahko opazujemo na eni strani notranje koordinate  $\mathbf{q}(.)$  na drugi strani pa navore v sklepih  $\boldsymbol{\tau}(.)$ , ki so potrebni za sledenje predpisanim trajektorijam. Tu moramo ločiti dva zapisa: (1) kompaktno obliko z eksplicitnimi enačbami za izračun notranjih koordinat, (2) Zanimajo nas pa lahko tudi navori potrebnii za realizacijo konkretnih, v obliki numeričnih vrednosti podanih notranjih koordinat. Pri drugem zapisu nas zanima povsem konkretna numerična rešitev za  $\boldsymbol{\tau}(.)$ , ki povzročijo ustrezni  $\mathbf{q}(.)$ , pri čemer ni pomemben ali potreben zapis v obliki enačb. Za prvi zapis je najbrž primernejša Lagrangeova obravnava, za drugi pa Newton-Eulerjeva. Pri študiju regulacijskih lastnosti ali elastičnih deformacij je zanesljivo primernejši Lagrangeov zapis.

V nadaljevanju bomo prikazali splošne enačbe Newton-Eulerjevega zapisu, na koncu pa jih bomo še uporabili pri obravnavi dvosegmentnega manipulatorja z dvema rotacijskima stopnjama.

## 6.2 LASTNOSTI ANTISIMETRIČNIH MATRIK

Pri računanju relativnih hitrosti in pospeškov teles smo vezani na odvajanje rotacijskih matrik. Pri tej operaciji si pomagamo z antisimetričnimi matrikami, ki imajo nekatere ugodne lastnosti. Zato za uvod najprej ponovimo in pojasnimo nekatere lastnosti antisimetričnih matrik.

Simetrična je matrika takrat, kadar je zgornja podmatrika (nad diagonalo) enaka spodnji podmatriki.

Antisimetričnost pa pomeni, da so tisti členi čez diagonalo enako veliki, vendar nasprotno predznačeni.

Matrika je antisimetrična, če zadovoljuje naslednjo relacijo

$$\mathbf{M}^T + \mathbf{M} = 0. \quad (6.1)$$

V veljavnost trditve (6.1) se lahko prepričamo z uporabo antisimetrične matrike (6.2) in seštevanjem po elementih.

Vektorje tudi lahko zapišemo v obliki antisimetričnih matrik. Če je vektor  $\mathbf{a} = [a_x, a_y, a_z]^T$ , potem pripadajočo antisimetrično matriko, ki opisuje isti vektor  $\mathbf{a}$  podaja izraz

$$\mathbf{M}(\mathbf{a}) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}. \quad (6.2)$$

Naštejmo nekaj lastnosti antisimetričnih matrik, ki jih bomo potrebovali pri izpeljavi Newton - Eulerjevih (NE) dinamičnih enačb večsegmentnih serijsko povezanih mehanizmov.

(L1) Linearnost.

$$\mathbf{M}(\alpha \mathbf{a} + \beta \mathbf{b}) = \alpha \mathbf{M}(\mathbf{a}) + \beta \mathbf{M}(\mathbf{b}). \quad (6.3)$$

(L2) Produkt antisimetrične matrike  $\mathbf{M}$ , ki opisuje nek vektor s poljubnim drugim vektorjem  $\mathbf{p}$  je vektorski produkt obeh.

$$\mathbf{M}(\mathbf{a})\mathbf{p} = \mathbf{a} \times \mathbf{p}. \quad (6.4)$$

Dokaz:

$$\begin{aligned} \mathbf{M}(\mathbf{a})\mathbf{p} &= \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} -a_z p_y + a_y p_z \\ a_z p_x - a_x p_z \\ -a_y p_x + a_x p_y \end{bmatrix} \\ \mathbf{a} \times \mathbf{p} &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_x & a_y & a_z \\ p_x & p_y & p_z \end{vmatrix} = \begin{bmatrix} a_y p_z - a_z p_y \\ a_z p_x - a_x p_z \\ a_x p_y - a_y p_x \end{bmatrix} \end{aligned}$$

Trditev je zaradi enakih rezultatov po obeh izračunih pravilna.

S podobnim malo daljšim izračunom bi lahko dokazali enakost ( $\mathbf{R}$  je rotacijska matrika)

$$\mathbf{R}(\mathbf{a} \times \mathbf{p}) = \mathbf{R}\mathbf{a} \times \mathbf{R}\mathbf{p}. \quad (6.5)$$

(L3) Matrična predstavitev  $\mathbf{M}(\mathbf{a})$  v koordinatnem sistemu, ki je rotiran z  $\mathbf{R}$ , je enaka antisimetrični matriki, ki izraža vektor  $\mathbf{a}$  rotiran z  $\mathbf{R}$ .

$$\mathbf{R} \mathbf{M}(\mathbf{a}) \mathbf{R}^T = \mathbf{M}(\mathbf{R}\mathbf{a}) \quad (6.6)$$

To enakost lahko dokažemo s pomočjo pravkar izpeljanih (L2)

$$\mathbf{M}(\mathbf{a})\mathbf{p} = \mathbf{a} \times \mathbf{p} \text{ in (L3)} \quad \mathbf{R}(\mathbf{a} \times \mathbf{p}) = \mathbf{R}\mathbf{a} \times \mathbf{R}\mathbf{p}.$$

Zapišemo izraz na levi  $\mathbf{R} \mathbf{M}(\mathbf{a}) \mathbf{R}^T$  skupaj z vektorjem  $\mathbf{p}$  v obliki  $\mathbf{R} \mathbf{M}(\mathbf{a}) \mathbf{R}^T \mathbf{p}$  in nadaljujemo

$$\begin{aligned} \dots \text{samo ločimo z oklepaji} &\quad \mathbf{R} \mathbf{M}(\mathbf{a}) \mathbf{R}^T \mathbf{p} = \mathbf{R} [\mathbf{M}(\mathbf{a}) \mathbf{R}^T \mathbf{p}] \\ \dots \text{nad oklepajem uporabimo prvo enačbo (6.4)} &\quad = \mathbf{a} \times \mathbf{R}^T \mathbf{p} \\ \dots \text{pomnožimo z } \mathbf{R} &\quad = (\mathbf{R}\mathbf{a}) \times (\mathbf{R} \mathbf{R}^T \mathbf{p}) \end{aligned}$$

... pokrajšamo na desni zaradi  $\mathbf{I} = \mathbf{R}\mathbf{R}^T = \mathbf{R}\mathbf{a} \times \mathbf{p}$

... upoštevamo ortogonalnost matrike  $\mathbf{R}$ , oziroma upoštevamo že izpeljani izraz (L2)  $\mathbf{M}(\mathbf{a})\mathbf{p} = \mathbf{a} \times \mathbf{p}$ , kar vodi do  $= \mathbf{M}(\mathbf{R}\mathbf{a})\mathbf{p}$

... če pokrajšamo še  $\mathbf{p}$  na obeh straneh dobimo prvotno trditev

$$\mathbf{R}\mathbf{M}(\mathbf{a})\mathbf{R}^T = \mathbf{M}(\mathbf{R}\mathbf{a}). \quad (6.7)$$

Pridobljeni izraz je zelo uporaben. Leva stran podaja podobnostno transformacijo matrike  $\mathbf{M}(\mathbf{a})$ . Enačba pravi, da je matrična predstavitev  $\mathbf{M}(\mathbf{a})$  v koordinatnem sistemu, ki je rotiran s transformacijo  $\mathbf{R}$ , enaka kot antisimetrična matrično izražena vrednost vektorja  $\mathbf{a}$ , ki je rotiran z  $\mathbf{R}$ , torej je  $\mathbf{M}(\mathbf{a})$ .

(L4) Odvod rotacije  $\mathbf{R}$  je enak zmnožku antisimetrične matrike, ki opisuje trenutni vektor kotne hitrosti relativnega koordinatnega sistema, ki rotira glede na osnovni koordinatni sistem in rotacijsko matriko  $\mathbf{R}$ .

$$\dot{\mathbf{R}} = \mathbf{M}(\boldsymbol{\omega}(t))\mathbf{R}. \quad (6.8)$$

Pokažimo, kako do tega pridemo. Privzemimo, da je rotacijska matrika  $\mathbf{R}$  funkcija samo ene spremenljivke  $\Theta$ . Ker je  $\mathbf{R}$  ortogonalna za vse  $\Theta$  velja

$$\mathbf{R}(\Theta)\mathbf{R}(\Theta)^T = \mathbf{I}.$$

Z odvajanjem obeh strani enačbe po kotu  $\Theta$ , na levi strani odvajamo kot produkt najprej prvi nato drugi člen, dobimo

$$\frac{d\mathbf{R}}{d\Theta}\mathbf{R}(\Theta)^T + \mathbf{R}(\Theta)\frac{d\mathbf{R}^T}{d\Theta} = 0. \quad (6.9)$$

Zaenkrat kar določimo, naj velja enakost

$$\mathbf{M} = \frac{d\mathbf{R}}{d\Theta}\mathbf{R}(\Theta)^T$$

in se spomnimo na  $\mathbf{M}^T + \mathbf{M} = 0$ . Ker velja v splošnem  $(\mathbf{A}\mathbf{B}) = (\mathbf{B}\mathbf{A}^T)^T$  lahko pišemo

$$\mathbf{M}^T = \left( \frac{d\mathbf{R}}{d\Theta}\mathbf{R}(\Theta)^T \right)^T = \mathbf{R}(\Theta)\frac{d\mathbf{R}^T}{d\Theta}.$$

$\mathbf{M}$  in  $\mathbf{M}^T$  seštejemo in dobimo

$$\mathbf{M} + \mathbf{M}^T = \frac{d\mathbf{R}}{d\Theta}\mathbf{R}(\Theta)^T + \mathbf{R}(\Theta)\frac{d\mathbf{R}^T}{d\Theta} = 0,$$

kar pomeni, da je definicija za  $\mathbf{M}$  pravilna. Če enačbo  $\mathbf{M} = \frac{d\mathbf{R}}{d\Theta} \mathbf{R}(\Theta)^T$  na obeh straneh pomnožimo z  $\mathbf{R}(\Theta)$ , dobimo

$$\mathbf{M} \mathbf{R}(\Theta) = \frac{d\mathbf{R}}{d\Theta} \mathbf{R}(\Theta)^T \mathbf{R}(\Theta).$$

Ob upoštevanju  $\mathbf{R}(\Theta)\mathbf{R}(\Theta)^T = \mathbf{I}$  na desni strani ostane le  $\dot{\mathbf{R}}$ , na levi pa  $\mathbf{M}(\boldsymbol{\omega}(t))\mathbf{R}$ , torej

$$\dot{\mathbf{R}} = \mathbf{M}(\boldsymbol{\omega}(t))\mathbf{R}.$$

Odvoda  $\mathbf{R}(\Theta)$  ni potrebno izračunavati z odvajanjem, ampak ga lahko dobimo z direktnim množenjem dveh matrik.

## 6.3 IZPELJAVA NEWTON-EULERJEVE FORMULACIJE

Newton-Eulerjeva formulacija temelji na osnovnih **Newtonovih zakonih** [Paul, 1981], ki jih podamo v naslednji obliki:

1. Vsaka akcija ima za posledico reakcijo. Če telo 1 deluje na telo 2 s silo  $\mathbf{f}$  in navorom  $\boldsymbol{\tau}$ , potem telo 2 deluje s silo  $-\mathbf{f}$  in navorom  $-\boldsymbol{\tau}$  na telo 1.
2. Sprememba gibalne količine telesa je enaka vsoti vseh sil, ki delujejo na telo.
3. Sprememba vrtilne količine telesa je enaka vsoti vseh navorov, ki delujejo na telo.

Če privzamemo, da se masa telesa ne spreminja, torej je konstantna, potem lahko napišemo stavek v drugi točki v obliki enačbe

$$\frac{d(m v_0)}{dt} = \mathbf{f}_0, \quad (6.10)$$

kjer je  $m$  masa telesa,  $v_0$  hitrost težišča in  $\mathbf{f}_0$  vsota vseh sil, ki delujejo na težišče telesa. Oba sta izražena v osnovnem koordinatnem sistemu. Ker je v večini robotskih aplikacij masa  $m$  konstantna, se omenjena enačba lahko pretvori v  $m\mathbf{a} = \mathbf{f}$ , kjer je  $\mathbf{a}$  pospešek.

Podobno lahko za trditev v tretjem stavku zapišemo enačbo

$$\frac{d(\mathbf{I}_0 \boldsymbol{\omega}_0)}{dt} = \boldsymbol{\tau}_0, \quad (6.11)$$

kjer je  $\mathbf{I}_0$  matrika vztrajnostnih momentov telesa,  $\boldsymbol{\omega}_0$  kotna hitrost telesa in  $\boldsymbol{\tau}_0$  vsota vseh navorov, ki delujejo na telo. Vsi so izraženi v osnovnem koordinatnem sistemu. Tukaj je pomembna razlika med translacijskim gibanjem in rotacijskim gibanjem. Čeprav je masa telesa ali segmentov lahko konstantna, v večini primerov ni mogoče trditi enako za vztrajnosti. Vztrajnost, izražena v osnovnem koordinatnem sistemu ( $\mathbf{I}_0$ ), je ali pa tudi ni konstantna.  $\mathbf{I}_0$  v glavnem ni konstantna in se spreminja glede na orientacijo telesa glede na osnovni koordinatni sistem. Da bi to pokazali, pripnemo k.s. na telo (segment) in definiramo  $\mathbf{I}_i$  glede na ta k.s.. V težišče telesa (segmenta) torej pripnemo relativni koordinatni sistem in glede nanj izrazimo matriko vztrajnostnih momentov telesa (6.11), ki je konstantna. Obe vztrajnostni matriki sta povezani z izrazom

$$\mathbf{I}_0 = \mathbf{R}_0^i \mathbf{I}_i \mathbf{R}_0^{i^T}. \quad (6.12)$$

Pri tem je  $\mathbf{R}_0^i$  rotacijska matrika, ki transformira iz relativnega k.s. (segmenta) v ničti k.s.. Ker  $\mathbf{R}_0^i$  časovno ni konstantna, vidimo, da se tudi  $\mathbf{I}_0$  spreminja s časom. Po tej poti (iz (6.11) direktno) ni mogoče določiti  $\boldsymbol{\tau}_0$ .

Stranska pot, ki gre mimo teh težav, je zapis enačbe kotne hitrosti s pomočjo vztrajnosti  $\mathbf{I}_i$ , glede na fiksni k.s. segmenta na segmentu

$$\boldsymbol{\tau}_i = \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i) + \mathbf{I}_i \dot{\boldsymbol{\omega}}_i. \quad (6.13)$$

To enačbo želimo sedaj izpeljati in dokazati.

$\mathbf{I}_i$  je (konstantna) matrika vztrajnosti telesa glede na k.s. segmenta,  $\boldsymbol{\omega}_i$  je kotna hitrost. Veličine v tej enačbi so izražene glede na relativni k.s.,  $\boldsymbol{\tau}_i$  je celoten navor na  $i$ -ti segment, zopet izražen v k.s. pripetem na segment robota.

V nadaljevanju bomo izpeljali to enačbo skupaj s takoimenovanim žiroskopskim (gyroscope) členom  $\boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i)$ . Kotna hitrost se transformira kot  $\boldsymbol{\omega}_i = \mathbf{R}_0^{i^T} \boldsymbol{\omega}_0$ , zaradi ortogonalnosti matrike  $\mathbf{R}_0^i$  pa velja tudi obratno  $\boldsymbol{\omega}_0 = \mathbf{R}_0^i \boldsymbol{\omega}_i$ .

Najprej izrazimo v en. (6.11) člen  $\mathbf{I}_0 \boldsymbol{\omega}_0$ , tako da vstavimo (6.12) namesto  $\mathbf{I}_0$  in tudi  $\boldsymbol{\omega}_0$  zapišemo preko rotacijske matrike

$$\boldsymbol{\Gamma}_0 = \mathbf{I}_0 \boldsymbol{\omega}_0 = \mathbf{R}_0^i \mathbf{I}_i \mathbf{R}_0^{i^T} \mathbf{R}_0^i \boldsymbol{\omega}_i = \mathbf{R}_0^i \mathbf{I}_i \boldsymbol{\omega}_i. \quad (6.14)$$

in odvajamo vrtilno količino  $\boldsymbol{\Gamma}_0$  (v osnovnem k.s.) po času, da dobimo vsoto navorov v en. (6.11). Odvajamo jo kot produkt, prvi in zadnji člen, saj je drugi člen (matrika vztrajnosti) konstanten (kar velja za k.s. pripet na segmentu)

$$\dot{\boldsymbol{\Gamma}}_0 = \boldsymbol{\tau}_0 = \dot{\mathbf{R}}_0^i \mathbf{I}_i \boldsymbol{\omega}_i + \mathbf{R}_0^i \mathbf{I}_i \dot{\boldsymbol{\omega}}_i. \quad (6.15)$$

Namesto  $\dot{\mathbf{R}}_0^i$  upoštevamo v skladu z ugotovitvijo na začetku  $\dot{\mathbf{R}}_0^i = \mathbf{M}(\boldsymbol{\omega}_0)\mathbf{R}_0^i$ , torej

$$\dot{\mathbf{I}}_0 = \boldsymbol{\tau}_0 = \mathbf{M}(\boldsymbol{\omega}_0)\mathbf{R}_0^i \mathbf{I}_i \boldsymbol{\omega}_i + \mathbf{R}_0^i \mathbf{I}_i \dot{\boldsymbol{\omega}}_i. \quad (6.16)$$

Lahko pa izrazimo navor, ki deluje na telo tudi v lokalnem koordinatnem sistemu segmenta, ne več v osnovnem k.s.. Že znani izraz za  $\boldsymbol{\tau}_0$  premultipli-ciramo z  $\mathbf{R}_0^{iT}$

$$\boldsymbol{\tau}_i = \mathbf{R}_0^{iT} \boldsymbol{\tau}_0 = \mathbf{R}_0^{iT} \mathbf{M}(\boldsymbol{\omega}_0) \mathbf{R}_0^i \mathbf{I}_i \boldsymbol{\omega}_i + \mathbf{R}_0^{iT} \mathbf{R}_0^i \mathbf{I}_i \dot{\boldsymbol{\omega}}_i. \quad (6.17)$$

Na tem mestu upoštevamo v prvem členu lastnost **(L3)**,  $\mathbf{R} \mathbf{M}(\mathbf{a}) \mathbf{R}^T = \mathbf{M}(\mathbf{R}\mathbf{a})$ , v drugem členu pa  $\mathbf{R}_0^{iT} \mathbf{R}_0^i = \mathbf{I}$  (enotska matrika), torej skupno

$$\boldsymbol{\tau}_i = \mathbf{R}_0^{iT} \boldsymbol{\tau}_0 = \mathbf{M}(\mathbf{R}_0^{iT} \boldsymbol{\omega}_0) \mathbf{I}_i \boldsymbol{\omega}_i + \mathbf{I}_i \dot{\boldsymbol{\omega}}_i. \quad (6.18)$$

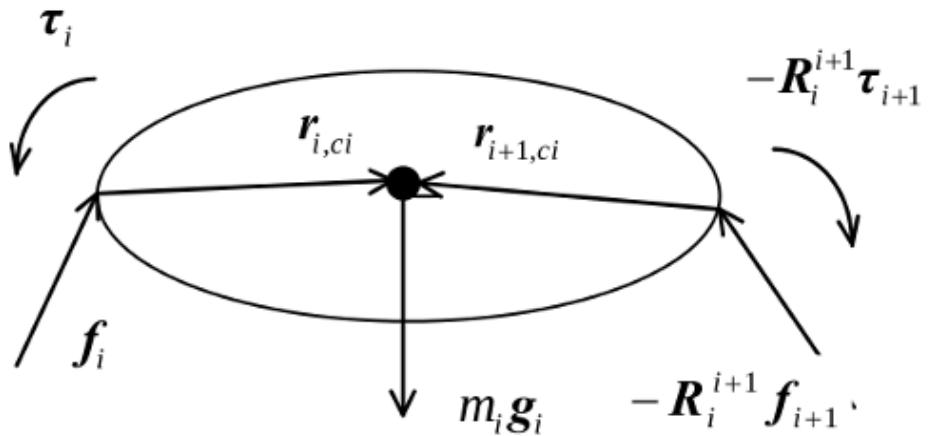
Za izraz v oklepaju je mogoče ugotoviti, da gre za kotno hitrost  $i$ -tega koordinatnega sistema, torej  $\boldsymbol{\omega}_i = \mathbf{R}_0^{iT} \boldsymbol{\omega}_0$ , poleg tega ob hkratnem upoštevanju **(L2)**,  $\mathbf{M}(\mathbf{a})\mathbf{p} = \mathbf{a} \times \mathbf{p}$  dobimo izraz

$$\boldsymbol{\tau}_i = \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i) + \mathbf{I}_i \dot{\boldsymbol{\omega}}_i. \quad (6.19)$$

Tako sta pripravljeni obe en. (6.10) in (6.19), ki sta izhodišče za rekurzivni izračun sil in momentov, ki delujejo v posameznih sklepih obravnavanega mehanizma. Ta navor je podan glede na relativni k.s., če pa bi ga želeli izraziti v osnovnem k.s., bi bilo tudi to možno. V nadaljevanju bomo videli, da je zapis sil in momentov v relativnem k.s. segmenta prikladnejši z enostavnnejšimi enačbami.

Sedaj bomo sestavili Newton-Eulerjev zapis enačb za gibanje  $n$ -segmentnega manipulatorja. Najprej izberemo koordinatne sisteme  $0, \dots, n$ , kjer je  $0$  k.s. splošni oz. globalni k.s., medtem ko je k.s.  $i$  oznaka fiksno pritrjen na segment  $i$ , pri  $i \geq 1$ .

V naslednjem koraku bomo definirali posamezne veličine in parametre, ki opisujejo posamezen segment mehanizma. Vsi so definirani glede na lokalni koordinatni sistem, torej glede na k.s.  $i$ .



Slika 6.1: Sile in momenti, ki delujejo na  $i$ -ti segment.

Najprej definirajmo hitrosti in pospeške različnih točk manipulatorja:

$\mathbf{a}_{c,i}$  = translacijski pospešek težišča  $i$ -tega segmenta,

$\mathbf{a}_{e,i}$  = translacijski pospešek konca  $i$ -tega segmenta (začetek  $i+1$  segmenta),

$\boldsymbol{\omega}_i$  = kotna hitrost  $i$ -tega koordinatnega sistema,

$\boldsymbol{\alpha}_i$  = kotni pospešek  $i$ -tega koordinatnega sistema.

Sledеči vektorji predstavljajo sile in navore:

$\mathbf{g}_i$  = gravitacijski pospešek (glede na  $i$ -ti koordinatni sistem),

$\mathbf{f}_i$  = sila, ki jo  $i-1$  segment izvaja na  $i$ -ti segment,

$\boldsymbol{\tau}_i$  = navor, ki ga  $i-1$  segment izvaja na  $i$ -ti segment,

$\mathbf{R}_i^{i+1}$  = rotacijska matrika, ki povezuje  $i$ -ti koordinatni sistem z  $i+1$  koordinatnim sistemom.

Zadnja skupina vektorjev opisuje fizikalne lastnosti manipulatorja. Poudarimo, da so vsi ti vektorji konstantni glede na  $\mathbf{q}$ . Z drugimi besedami, vsi vektorji v naslednjem spisku so neodvisni od konfiguracije (lege) manipulatorja.

$m_i$  = masa  $i$ -tega segmenta,

$\mathbf{I}_i$  = vztrajnostna matrika  $i$ -tega segmenta glede na  $i$ -ti koordinatni sistem,

- $\mathbf{r}_{i,ci}$  = vektor, ki povezuje  $i$ -ti sklep s težiščem  $i$ -tega segmenta,
- $\mathbf{r}_{i+1,ci}$  = vektor, ki povezuje  $i + 1$  sklep s težiščem  $i$ -tega segmenta,
- $\mathbf{r}_{i,i+1}$  = vektor, ki povezuje  $i$ -ti sklep z  $i + 1$  sklepom.

Sedaj lahko zapišemo obe Newtonovi enačbi, ki veljata za  $i$ -ti segment na sliki 6.1. Ta slika kaže segment  $i$  skupaj z vsemi silami in navori, ki so povezani z njim. Analizirajmo sile in momente prikazane na sliki.  $\mathbf{f}_i$  je sila, s katero deluje segment  $i - 1$  na segment  $i$ . Glede na akcijo/reakcijo deluje tudi segment  $i + 1$  s silo  $-\mathbf{f}_{i+1}$  nazaj na segment  $i$ , vendar glede na naše označevanje in dogovor je ta vektor izražen glede na k.s.  $i + 1$ . Da bi se ta vektor nanašal na k.s.  $i$ , ga je potrebno pomnožiti z rotacijsko matriko  $\mathbf{R}_i^{i+1}$ . Podobno razmišljajmo velja za navor  $\boldsymbol{\tau}_i$  in  $-\mathbf{R}_i^{i+1}\boldsymbol{\tau}_{i+1}$ . Sila  $m_i\mathbf{g}_i$  je gravitacijska sila segmenta, ki deluje v težišču. Ker so vsi parametri izraženi v k.s.  $i$ , je tudi gravitacijski vektor  $\mathbf{g}_i$  izražen kot funkcija  $i$ .

Vsota vseh sil glede na podano sliko in pravkar podano razlago je enaka

$$\mathbf{f}_i - \mathbf{R}_i^{i+1}\mathbf{f}_{i+1} + m_i\mathbf{g}_i = m_i\mathbf{a}_{c,i}. \quad (6.20)$$

V nadaljevanju bomo zapisali vsoto vseh momentov v točki težišča. Tukaj je pomembno poudariti dvoje.

Moment, ki ga v splošnem povzroča sila  $\mathbf{f}$  okrog neke točke je podan kot vektorski produkt  $\mathbf{f} \times \mathbf{r}$ .  $\mathbf{r}$  je vektor razdalje od sile pa do točke, v kateri računamo moment.

V sledeči momentni enačbi se vektor sile  $m_i\mathbf{g}_i$  ne pojavlja, saj deluje prav v težišču  $i$ -tega segmenta. Prav v težišču pa pišemo momentno enačbo, zato ne prispeva k navoru.

$$\boldsymbol{\tau}_i - \mathbf{R}_i^{i+1}\boldsymbol{\tau}_{i+1} + \mathbf{f}_i \times \mathbf{r}_{i,ci} - (\mathbf{R}_i^{i+1}\mathbf{f}_{i+1}) \times \mathbf{r}_{i+1,ci} = \mathbf{I}_i\boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i\boldsymbol{\omega}_i). \quad (6.21)$$

Enačba predstavlja vsoto vseh momentov.

Prihajamo do bistva Newton-Eulerjeve metode, pri kateri iščemo vektorje  $\mathbf{f}_1, \dots, \mathbf{f}_n$  in  $\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_n$  glede na podane trajektorije, torej glede na vektorje  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  in  $\ddot{\mathbf{q}}$ . Iščemo torej sile in navore v sklepih manipulatorja, ki ustreza podanim trajektorijam v posameznih sklepih, njenim prvim in drugim odvodom. Metodo lahko uporabimo v dveh korakih.

Najprej lahko začnemo s prvim segmentom ( $i = 1$ ) in na osnovi trajektorij določimo  $\boldsymbol{\omega}_i$ ,  $\mathbf{a}_{c,i}$ ,  $\mathbf{a}_{e,i}$  in  $\boldsymbol{\alpha}_i$ . Povečamo  $i$  za 1 in postopek ponovimo za

naslednji segment. Tako nadaljujemo rekurzivno proti vrhu robota.

Nadaljujemo s pogojem  $\mathbf{f}_{n+1} = 0$  in  $\boldsymbol{\tau}_{n+1} = 0$ , kar smemo storiti, saj ni segmenta  $n + 1$ . Začnemo nasprotno kot prej, pri  $i = n$ , torej pri zadnjem segmentu. Rekurzija poteka v obratnem vrstnem redu proti bazi robota ( $i = n, n - 1, n - 2 \dots$ ). Silo in navor je možno določiti ob poznavanju (prej določenih) kinematičnih parametrov kot

$$\mathbf{f}_i = \mathbf{R}_i^{i+1} \mathbf{f}_{i+1} + m_i \mathbf{a}_{c,i} - m_i \mathbf{g}_i. \quad (6.22)$$

$$\boldsymbol{\tau}_i = \mathbf{R}_i^{i+1} \boldsymbol{\tau}_{i+1} - \mathbf{f}_i \times \mathbf{r}_{i,ci} + (\mathbf{R}_i^{i+1} \mathbf{f}_{i+1}) \times \mathbf{r}_{i+1,ci} + \mathbf{I}_i \boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i). \quad (6.23)$$

## 6.4 IZRAČUN KINEMATIČNIH VELIČIN

Da pa bo izračun po katerikoli poti možen, je nujno določiti še vse kinematične veličine, ustrezne translacijske in rotacijske pomike, hitrosti in pospeške. Iz  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  in  $\ddot{\mathbf{q}}$  določamo  $\boldsymbol{\omega}_i$ ,  $\mathbf{a}_{c,i}$ ,  $\mathbf{a}_{e,i}$  in  $\boldsymbol{\alpha}_i$ . Izračun je možno napraviti rekurzivno v smeri naraščajočega  $i$ .

Ker nas v našem primeru zanimajo le mehanizmi z rotacijskimi sklepi z eno stopnjo prostosti, bomo izpeljavo kotnih ter translacijskih hitrosti in pospeškov opravili le za omenjen primer. Obravnava translacijskih pomikov je bolj preprosta.

Za označevanje baznega (osnovnega) k.s. je uporabljen indeks (0), za razliko od k.s. pripetega na segmentu, ki je označen z indeksom  $i$ . Tako na primer zapis  $\boldsymbol{\omega}_i$  predstavlja kotno hitrost koordinatnega sistema  $i$  izraženega v k.s.  $i$ , medtem ko  $\boldsymbol{\omega}_i^0$  predstavlja isto veličino izraženo glede na osnovni k.s..

Kotna hitrost  $i$ -tega segmenta je enaka kotni hitrosti  $i - 1$  segmenta, kateri dodamo rotacijo v  $i$ -tem sklepu.

$$\boldsymbol{\omega}_i^0 = \boldsymbol{\omega}_{i-1}^0 + \mathbf{e}_i^0 \dot{\mathbf{q}}_i, \quad (6.24)$$

kjer je  $\mathbf{e}_i^0$  enotski vektor izražen v osnovnem koordinatnem sistemu, ki podaja smer vrtenja  $i$ -tega sklepa in je  $\dot{\mathbf{q}}_i$  skalarna vrednost rotacijske hitrosti  $i$ -tega sklepa. Da bi dobili povezavo med  $\boldsymbol{\omega}_i$  in  $\boldsymbol{\omega}_{i+1}$  moramo gornjo enačbo izraziti v k.s.  $i$  in ne več v osnovnem k.s. kot zgoraj.

Poglejmo malo drugačen zapis za  $\mathbf{R}_0^i$

$$\begin{aligned}\mathbf{R}_0^i &= \mathbf{R}_0^{i-1} \mathbf{R}_{i-1}^i, \\ \mathbf{R}_0^{i T} &= (\mathbf{R}_0^{i-1} \mathbf{R}_{i-1}^i)^T = \mathbf{R}_{i-1}^{i T} \mathbf{R}_0^{i-1 T} \\ \mathbf{R}_0^{i-1 T} \boldsymbol{\omega}_{i-1}^0 &= \boldsymbol{\omega}_{i-1}, \quad \boldsymbol{\omega}_i = \mathbf{R}_0^{i T} \boldsymbol{\omega}_i^0, \quad \mathbf{e}_i = \mathbf{R}_0^{i T} \mathbf{e}_i^0\end{aligned}$$

Ker sta  $\boldsymbol{\omega}_i$  in  $\boldsymbol{\omega}_{i+1}$  izražena vsaka v svojem (drugem) k.s., lahko prejšnjo enačbo zapišemo v  $i$ -tem koordinatnem sistemu v obliki

$$\boldsymbol{\omega}_i = \mathbf{R}_{i-1}^{i T} \boldsymbol{\omega}_{i-1} + \mathbf{e}_i \dot{\mathbf{q}}_i. \quad (6.25)$$

V tej enačbi je os rotacije sklepa  $i$  izražena v k.s.  $i$  kot  $\mathbf{e}_i^0 = \mathbf{R}_i^0 \mathbf{e}_i$ .

V nadaljevanju bomo obravnavali kotni pospešek  $\boldsymbol{\alpha}_i$ . Pri tem najprej omenimo transformacijsko enačbo med kotnim pospeškom podanim v osnovnem in lokalnem k.s.

$$\boldsymbol{\alpha}_i = \mathbf{R}_i^{i T} \boldsymbol{\omega}_i^0. \quad (6.26)$$

Drugache povedano,  $\boldsymbol{\alpha}_i$  je odvod kotne hitrosti koordinatnega sistema  $i$ , vendar izražen v k.s.  $i$ . NE DRŽI trditev  $\boldsymbol{\alpha}_i = \dot{\boldsymbol{\omega}}_i^0$  (drži pa  $\boldsymbol{\alpha}_i^0 = \dot{\boldsymbol{\omega}}_i^0$ )! Podobno velja za hitrosti in pospeške v težiščih segmentov. Z direktnim odvajanjem enačbe (6.24) dobimo izraz za kotni pospešek  $i$ -tega koordinatnega sistema

$$\dot{\boldsymbol{\omega}}_i^0 = \dot{\boldsymbol{\omega}}_{i-1}^0 + \mathbf{e}_i^0 \ddot{\mathbf{q}}_i + \dot{\mathbf{e}}_i^0 \dot{\mathbf{q}}_i. \quad (6.27)$$

Ta zapis je zopet izražen v baznem k.s., zato ga zapišimo v k.s.  $i$ . Ker je  $\mathbf{e}_i^0 = \mathbf{R}_i^0 \mathbf{e}_i$ , odvajamo kot produkt, vendar je odvod enotskega vektorja enak nič, zato ostane samo odvod rotacijske matrike

$$\begin{aligned}\dot{\mathbf{e}}_i^0 &= \dot{\mathbf{R}}_0^i \mathbf{e}_i \\ \dot{\mathbf{e}}_i^0 &= \mathbf{M}(\boldsymbol{\omega}_i^0) \mathbf{R}_0^i \mathbf{e}_i \\ \dot{\mathbf{e}}_i^0 &= \mathbf{M}(\boldsymbol{\omega}_i^0) \mathbf{e}_i^0 \\ \dot{\mathbf{e}}_i^0 &= \boldsymbol{\omega}_i^0 \times \mathbf{e}_i^0\end{aligned}$$

Pri okrajšavi te enačbe smo upoštevali že izpeljane (**L4**)  $\dot{\mathbf{R}} = \mathbf{M}(\boldsymbol{\omega}(t)) \mathbf{R}$ ,  $\mathbf{e}_i^0 = \mathbf{R}_i^0 \mathbf{e}_i$  in (**L2**)  $\mathbf{M}(\mathbf{a}) \mathbf{p} = \mathbf{a} \times \mathbf{p}$ .

Enačbo z odvodom upoštevamo v (6.27), nakar lahko zapišemo

$$\dot{\boldsymbol{\omega}}_i^0 = \dot{\boldsymbol{\omega}}_{i-1}^0 + \mathbf{e}_i^0 \ddot{\mathbf{q}}_i + (\boldsymbol{\omega}_i^0 \times \mathbf{e}_i^0) \dot{\mathbf{q}}_i. \quad (6.28)$$

Sedaj namesto  $\omega_i^0$  upoštevamo en. (6.24) ( $\omega_i^0 = \omega_{i-1}^0 + e_i^0 \dot{q}_i$ ). Zaradi vektorskega produkta  $e_i^0 \times e_i^0 = 0$  velja

$$\begin{aligned}\omega_i^0 \times e_i^0 &= (\omega_{i-1}^0 + e_i^0 \dot{q}_i) \times e_i^0 \\ \omega_i^0 \times e_i^0 &= \omega_{i-1}^0 \times e_i^0\end{aligned}$$

Enačbo (6.28) na obeh straneh pomnožimo z  $R_0^{i^T}$ . Na levi strani enačbe dobimo produkt in upoštevamo okrajšavo  $\alpha_i = R_0^{i^T} \dot{\omega}_i^0$ , kar je mogoče zapisati tudi v obliki  $\alpha_i = R_0^{i^T} \alpha_i^0$ . Enačbo za kotni pospešek  $i$ -tega koordinatnega sistema zapišemo v naslednji obliki

$$\alpha_i = R_0^{i^T} \alpha_{i-1}^0 + R_0^{i^T} e_i^0 \ddot{q}_i + R_0^{i^T} (\omega_{i-1}^0 \times e_i^0) \dot{q}_i \quad (6.29)$$

in še ob upoštevanju  $\omega_i^0 = R_0^i \omega_i$  ter  $e_i^0 = R_i^0 e_i$  na koncu dobimo

$$\alpha_i = R_{i-1}^{i^T} \alpha_{i-1} + e_i \ddot{q}_i + (R_{i-1}^{i^T} \omega_{i-1} \times e_i) \dot{q}_i. \quad (6.30)$$

V zvezi s prvim členom glej razlago pri izpeljavi en. (6.25).

Sedaj se lotimo še izpeljave izrazov za translacijske hitrosti in pospeške težišč posameznih sklepov. Omenimo, da se v nasprotju s kotno hitrostjo ne pojavlja linearna hitrost nikjer v dinamičnih enačbah, vendar je potrebna za izračun linearne pospeška. Položaj težišča  $i$ -tega sklepa v splošnem koordinatnem sistemu podaja izraz

$$\mathbf{p}_{c,i}^0 = R_0^i \mathbf{r}_{i,ci} + \mathbf{p}_{e,i-1}^0. \quad (6.31)$$

Člen na levi strani predstavlja položaj težišča, zadnji člen pa vrh prejšnjega segmenta. Prvi člen na desni je zasukan vektor do obravnavanega težišča. Po dvakratnem odvajjanju zgornjega izraza dobimo izraz za pospešek težišča  $i$ -tega segmenta izraženega v osnovnem koordinatnem sistemu. Po prvem odvajjanju dobimo

$$\mathbf{v}_{c,i}^0 = \dot{\mathbf{p}}_{c,i}^0 = \dot{R}_0^i \mathbf{r}_{i,ci} + \mathbf{v}_{e,i-1}^0. \quad (6.32)$$

kar je translacijska hitrost težišča. Upoštevamo  $\dot{\mathbf{R}} = \mathbf{M}(\boldsymbol{\omega}(t)) \mathbf{R}$

$$\mathbf{v}_{c,i}^0 = \dot{\mathbf{p}}_{c,i}^0 = \mathbf{M}(\boldsymbol{\omega}_i^0) R_0^i \mathbf{r}_{i,ci} + \mathbf{v}_{e,i-1}^0. \quad (6.33)$$

Upoštevamo še  $\mathbf{M}(\mathbf{a}) \mathbf{p} = \mathbf{a} \times \mathbf{p}$ , da dobimo vektorski produkt

$$\mathbf{v}_{c,i}^0 = \dot{\mathbf{p}}_{c,i}^0 = \boldsymbol{\omega}_i^0 \times (R_0^i \mathbf{r}_{i,ci}) + \mathbf{v}_{e,i-1}^0. \quad (6.34)$$

Po drugem odvajjanju in ponovnem upoštevanju najprej  $\dot{\mathbf{R}} = \mathbf{M}(\boldsymbol{\omega}(t))\mathbf{R}$  in zatem  $\mathbf{M}(\mathbf{a})\mathbf{p} = \mathbf{a} \times \mathbf{p}$

$$\begin{aligned}\mathbf{a}_{c,i}^0 &= \dot{\mathbf{v}}_{c,i}^0 \\ \mathbf{a}_{c,i}^0 &= \dot{\boldsymbol{\omega}}_i^0 \times (\mathbf{R}_0^i \mathbf{r}_{i,ci}) + \boldsymbol{\omega}_i^0 \times (\dot{\mathbf{R}}_0^i \mathbf{r}_{i,ci}) + \mathbf{a}_{e,i-1}^0 \\ \mathbf{a}_{c,i}^0 &= \dot{\boldsymbol{\omega}}_i^0 \times (\mathbf{R}_0^i \mathbf{r}_{i,ci}) + \boldsymbol{\omega}_i^0 \times (\boldsymbol{\omega}_i^0 \times (\mathbf{R}_0^i \mathbf{r}_{i,ci})) + \mathbf{a}_{e,i-1}^0.\end{aligned}\quad (6.35)$$

Enako kot pri izpeljavi kotnega pospeška, veljajo transformacijske enakosti iz osnovnega v lokalni k.s.  $\mathbf{a}_{c,i} = \mathbf{R}_0^{iT} \mathbf{a}_{c,i}^0$ ,  $\boldsymbol{\omega}_i^0 = \mathbf{R}_0^i \boldsymbol{\omega}_i$ ,  $\mathbf{a}_{e,i-1}^0 = \mathbf{R}_0^{i-1} \mathbf{a}_{e,i-1}$ .

Gornjo enačbo na levi in desni strani pomnožimo z rotacijsko matriko  $\mathbf{R}_0^{iT}$ .

Na levi strani enačbe po množenju upoštevamo  $\mathbf{a}_{c,i} = \mathbf{R}_0^{iT} \mathbf{a}_{c,i}^0$ .

V prvem členu na desni strani upoštevamo  $\mathbf{R}_0^{iT} \dot{\boldsymbol{\omega}}_i^0 = \dot{\boldsymbol{\omega}}_i$ .

V drugem členu na desni dvakrat upoštevamo  $\mathbf{R}_0^{iT} \boldsymbol{\omega}_i^0 = \boldsymbol{\omega}_i$ .

V zadnjem členu na desni strani upoštevamo  $\mathbf{R}_0^{i-1T} \mathbf{a}_{e,i-1}^0 = \mathbf{a}_{e,i-1}$  oz.  $\mathbf{R}_0^{iT} \mathbf{a}_{e,i-1}^0 = \mathbf{R}_{i-1}^{iT} \mathbf{a}_{e,i-1}$ .

Dobimo izraz za pospešek težišča  $i$ -tega segmenta tudi v lokalnem koordinatnem sistemu  $i$ -tega segmenta

$$\mathbf{a}_{c,i} = \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i,ci}) + \mathbf{R}_{i-1}^{iT} \mathbf{a}_{e,i-1}. \quad (6.36)$$

Podoben izraz velja tudi za pospešek končne točke  $i$ -tega segmenta, le da v vektorskih produktih na dveh mestih nastopa druga ročica

$$\mathbf{a}_{e,i} = \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i,i+1} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i,i+1}) + \mathbf{R}_{i-1}^{iT} \mathbf{a}_{e,i-1}. \quad (6.37)$$

Do tega izraza bi prišli preko podobne izpeljave kot pri izpeljavi za  $\mathbf{a}_{c,i}$ , zato podrobno ni prikazana.

Enačbe (6.25), (6.30), (6.36) in (6.37) so kinematične rekurzivne enačbe, ki smo jih iskali in s katerimi izračunamo izraze za kotne in linearne hitrosti segmentov. Zapišimo jih še enkrat

$$\boldsymbol{\omega}_i = \mathbf{R}_{i-1}^{iT} \boldsymbol{\omega}_{i-1} + \mathbf{e}_i \dot{\mathbf{q}}_i \quad (6.25)$$

$$\boldsymbol{\alpha}_i = \mathbf{R}_{i-1}^{iT} \boldsymbol{\alpha}_{i-1} + \mathbf{e}_i^0 \ddot{\mathbf{q}}_i + (\mathbf{R}_{i-1}^{iT} \boldsymbol{\omega}_{i-1} \times \mathbf{e}_i) \dot{\mathbf{q}}_i. \quad (6.30)$$

$$\mathbf{a}_{c,i} = \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i,ci}) + \mathbf{R}_{i-1}^{iT} \mathbf{a}_{e,i-1} \quad (6.36)$$

$$\mathbf{a}_{e,i} = \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i,i+1} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i,i+1}) + \mathbf{R}_{i-1}^{iT} \mathbf{a}_{e,i-1}. \quad (6.37)$$

Sedaj poznamo vse elemente rekurzivnega izračuna Newton-Eulerjeve metode. Izračun je torej dvostopenjski:

Začnemo z začetnimi pogoji  $\boldsymbol{\omega}_0 = 0$ ,  $\boldsymbol{a}_{c,0} = 0$ ,  $\boldsymbol{a}_{e,0} = 0$  in  $\boldsymbol{\alpha}_0 = 0$ , jih vstavimo v enačbe (6.25), (6.30), (6.36) in (6.37) in izračunamo prav v tem vrstnem redu. Ponovimo rekurzivno od  $i = 1 \dots n$ .

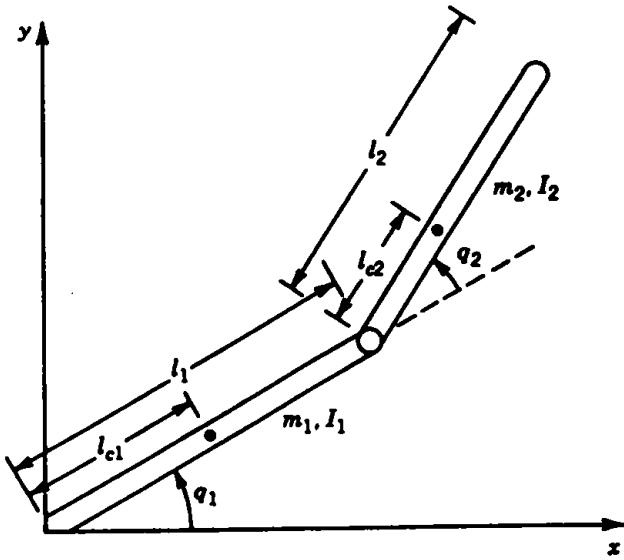
Nadaljujemo s pogoji na vrhu manipulatorja  $\boldsymbol{f}_{n+1} = 0$  in  $\boldsymbol{\tau}_{n+1} = 0$ , ter uporabimo en. (6.22) in (6.23) rekurzivno od  $i = n \dots 1$ .

$$\boldsymbol{f}_i = \boldsymbol{R}_i^{i+1} \boldsymbol{f}_{i+1} + m_i \boldsymbol{a}_{c,i} - m_i \boldsymbol{g}_i \quad (6.22)$$

$$\boldsymbol{\tau}_i = \boldsymbol{R}_i^{i+1} \boldsymbol{\tau}_{i+1} - \boldsymbol{f}_i \times \boldsymbol{r}_{i,ci} + (\boldsymbol{R}_i^{i+1} \boldsymbol{f}_{i+1}) \times \boldsymbol{r}_{i+1,ci} + \boldsymbol{I}_i \boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times (\boldsymbol{I}_i \boldsymbol{\omega}_i). \quad (6.23)$$

### Primer: Dvosegmentni manipulator z rotacijskima sklepoma

V nadaljevanju bo uporabljena Newton-Eulerjeva metoda za izračun dinamike dvosegmentnega manipulatorja z rotacijskima sklepoma, ki je prikazan na sliki 6.2. Iz končnih rezultatov bo tudi razvidno, da Newton-Euler metoda vodi do enakih enačb kot Lagrangeova metoda.



Slika 6.2: Dvosegmentni manipulator z dvema rotacijskima sklepoma.

Začnemo z rekurzijo naprej, od baze proti vrhu manipulatorja, da bi najprej zapisali različne kinematične veličine (hitrosti in pospeške) v odvisnosti od  $q_1, q_2$  in njunih odvodov.

V obravnavanem primeru lahko že iz same slike manipulatorja ugotovimo relacije za kotno hitrost in pospešek prvega in drugega segmenta

$$\begin{aligned}\boldsymbol{\omega}_1 &= \dot{q}_1 \mathbf{k}, & \boldsymbol{\omega}_2 &= (\dot{q}_1 + \dot{q}_2) \mathbf{k}, \\ \boldsymbol{\alpha}_1 &= \ddot{q}_1 \mathbf{k}, & \boldsymbol{\alpha}_2 &= (\ddot{q}_1 + \ddot{q}_2) \mathbf{k}.\end{aligned}$$

Torej ni potrebno uporabiti en. (6.25) in (6.30), ki imajo isti namen.

Vektorji razdalje, ki so neodvisni od konfiguracije, so zopet glede na sliko za obravnavana segmenta enaki

$$\begin{aligned}\mathbf{r}_{1,c1} &= l_{c1} \mathbf{i}, & \mathbf{r}_{2,c1} &= -(l_1 - l_{c1}) \mathbf{i}, & \mathbf{r}_{1,2} &= l_1 \mathbf{i}, \\ \mathbf{r}_{2,c2} &= l_{c2} \mathbf{i}, & \mathbf{r}_{3,c2} &= -(l_2 - l_{c2}) \mathbf{i}, & \mathbf{r}_{2,3} &= l_2 \mathbf{i}.\end{aligned}$$

Za manipulator vzačetni legi, ko je iztegnjen vzdolž  $x$ -osi,  $\mathbf{R}_0^1$  ustreza  $\mathbf{A}_1$ ,  $\mathbf{R}_1^2$  ustreza  $\mathbf{A}_2$  in  $\mathbf{R}_0^2$  ustreza  $\mathbf{T}_0^2$ . Rotacijske matrike  $\mathbf{R}_0^1, \mathbf{R}_1^2, \mathbf{R}_0^2$  so enake

$$\begin{aligned}\mathbf{R}_0^1 &= \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 \\ \sin q_1 & \cos q_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{R}_0^{1T} &= \begin{bmatrix} \cos q_1 & \sin q_1 & 0 \\ -\sin q_1 & \cos q_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{R}_1^2 &= \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 \\ \sin q_2 & \cos q_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{R}_1^{2T} &= \begin{bmatrix} \cos q_2 & \sin q_2 & 0 \\ -\sin q_2 & \cos q_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{R}_0^2 &= \begin{bmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 \\ \sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{R}_0^{2T} &= \begin{bmatrix} \cos(q_1 + q_2) & \sin(q_1 + q_2) & 0 \\ -\sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}.\end{aligned}$$

### 6.4.1 Rekurzija naprej: prvi segment

Začnemo z  $i = 1$ , en. (6.36) ( $\mathbf{a}_{c,i} = \mathbf{R}_{i-1}^i {}^T \mathbf{a}_{e,i-1} + \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i,ci})$ ) in ob upoštevanju  $\mathbf{a}_{e,0} = 0$ ,  $\boldsymbol{\omega}_1 = \dot{q}_1 \mathbf{k}$ ,  $\dot{\boldsymbol{\omega}}_1 = \ddot{q}_1 \mathbf{k}$  in  $\mathbf{r}_{1,c1} = l_{c1} \mathbf{i}$  zapišemo

$$\begin{aligned}\mathbf{a}_{c,1} &= \mathbf{R}_{i-1}^i {}^T \mathbf{a}_{e,i-1} + \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i,ci}) \\ \mathbf{a}_{c,1} &= \ddot{q}_1 \mathbf{k} \times l_{c1} \mathbf{i} + \dot{q}_1 \mathbf{k} \times (\dot{q}_1 \mathbf{k} \times l_{c1} \mathbf{i}).\end{aligned}$$

Zapišemo v matrični obliki

$$\boldsymbol{a}_{c,1} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ 0 & 0 & \ddot{q}_1 \\ l_{c1} & 0 & 0 \end{vmatrix} + \ddot{q}_1 \boldsymbol{k} \times \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ 0 & 0 & \dot{q}_1 \\ l_{c1} & 0 & 0 \end{vmatrix}.$$

Drugi člen še enkrat

$$\boldsymbol{a}_{c,1} = l_{c1} \ddot{q}_1 \boldsymbol{j} + \ddot{q}_1 \boldsymbol{k} \times l_{c1} \dot{q}_1 \boldsymbol{j} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ 0 & 0 & \dot{q}_1 \\ 0 & l_{c1} \dot{q}_1 & 0 \end{vmatrix} = l_{c1} \ddot{q}_1 \boldsymbol{j} - l_{c1} \dot{q}_1^2 \boldsymbol{i} = \begin{bmatrix} -l_{c1} \dot{q}_1^2 \\ l_{c1} \ddot{q}_1 \\ 0 \end{bmatrix}.$$

Kot vidimo, je izračun v k.s. 1 zelo preprost.

Določiti je potrebno  $\boldsymbol{g}_1$ , kjer je  $g$  gravitacijski pospešek. Os  $y$  je usmerjena v strop, gravitacija v osnovni legi torej deluje v  $-y$ , oz.  $-\boldsymbol{j}$  smeri.

$$\boldsymbol{g}_1 = -\boldsymbol{R}_0^{1T} g \boldsymbol{j} = -\begin{bmatrix} \cos q_1 & \sin q_1 & 0 \\ -\sin q_1 & \cos q_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = g \begin{bmatrix} -\sin q_1 \\ -\cos q_1 \\ 0 \end{bmatrix}.$$

Od tu naprej lahko zapise poenostavimo, tako da izpustimo tretjo komponento, ker je v tem dvodimensijskem primeru očitno vedno enaka nič. Podobno bo tudi tretja komponenta pri vseh silah enaka nič, pri navorih pa bosta prvi dve komponenti enaki nič.

Izračun za prvi segment zaključimo s pospeškom konca prvega segmenta. Najlažje ga določimo iz  $\boldsymbol{a}_{c,1}$ , ki je že določen zgoraj, tako da  $l_{c1}$  nadomestimo z  $l_1$

$$\boldsymbol{a}_{e,1} = \begin{bmatrix} -l_1 \dot{q}_1^2 \\ l_1 \ddot{q}_1 \\ 0 \end{bmatrix}.$$

#### 6.4.2 Rekurzija naprej: drugi segment

Začnemo zopet z  $i = 2$  in s splošno en. (6.36)  $(\boldsymbol{a}_{c,i} = \boldsymbol{R}_{i-1}^i \boldsymbol{a}_{e,i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i,ci}))$  in ob upoštevanju že znanega  $\boldsymbol{\omega}_2$  pišemo

$$\boldsymbol{a}_{c,2} = \boldsymbol{R}_1^{2T} \boldsymbol{a}_{e,1} + \dot{\boldsymbol{\omega}}_2 \times \boldsymbol{r}_{2,c2} + \boldsymbol{\omega}_2 \times (\boldsymbol{\omega}_2 \times \boldsymbol{r}_{2,c2}).$$

Poznamo že  $\boldsymbol{\omega}_2 = (\dot{q}_1 + \dot{q}_2)\mathbf{k}$ ,  $\dot{\boldsymbol{\omega}}_2 = \boldsymbol{\alpha}_2 = (\ddot{q}_1 + \ddot{q}_2)\mathbf{k}$ , in  $\mathbf{r}_{2,c2} = l_{c2}\mathbf{i}$ . Vstavimo v zgornjo enačbo in dobimo

$$\mathbf{a}_{c,2} = \mathbf{R}_1^{2T} \mathbf{a}_{e,1} + ((\ddot{q}_1 + \ddot{q}_2)\mathbf{k}) \times l_{c2}\mathbf{i} + ((\dot{q}_1 + \dot{q}_2)\mathbf{k}) \times ((\dot{q}_1 + \dot{q}_2)\mathbf{k}) \times l_{c2}\mathbf{i}$$

$$\mathbf{R}_1^{2T} \mathbf{a}_{e,1} = \begin{bmatrix} \cos q_2 & \sin q_2 \\ -\sin q_2 & \cos q_2 \end{bmatrix} \begin{bmatrix} -l_1\dot{q}_1^2 \\ l_1\ddot{q}_1 \end{bmatrix}.$$

Se pomnožimo

$$\mathbf{R}_1^{2T} \mathbf{a}_{e,1} = \begin{bmatrix} -l_1\dot{q}_1^2 \cos q_2 + l_1\ddot{q}_1 \sin q_2 \\ l_1\dot{q}_1^2 \sin q_2 + l_1\ddot{q}_1 \cos q_2 \end{bmatrix}.$$

Zgornji izraz vstavimo v enačbo za  $\mathbf{a}_{c,2}$  ter tudi ostale člene v enačbi zapišemo v matrični obliki

$$\mathbf{a}_{c,2} = \begin{bmatrix} -l_1\dot{q}_1^2 \cos q_2 + l_1\ddot{q}_1 \sin q_2 \\ l_1\dot{q}_1^2 \sin q_2 + l_1\ddot{q}_1 \cos q_2 \end{bmatrix} + \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0 & \ddot{q}_1 + \ddot{q}_2 \\ l_{c2} & 0 & 0 \end{vmatrix} + ((\dot{q}_1 + \dot{q}_2)\mathbf{k}) \times \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0 & \dot{q}_1 + \dot{q}_2 \\ l_{c2} & 0 & 0 \end{vmatrix}.$$

Matrični produkt zapišemo v drugačni obliki

$$\mathbf{a}_{c,2} = \begin{bmatrix} -l_1\dot{q}_1^2 \cos q_2 + l_1\ddot{q}_1 \sin q_2 \\ l_1\dot{q}_1^2 \sin q_2 + l_1\ddot{q}_1 \cos q_2 \end{bmatrix} + \begin{bmatrix} 0 \\ l_{c2}(\ddot{q}_1 + \ddot{q}_2) \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0 & \dot{q}_1 + \dot{q}_2 \\ 0 & l_{c2}(\dot{q}_1 + \dot{q}_2) & 0 \end{bmatrix}.$$

Poenostavimo in rezultat matričnega produkta zapišemo v obliki vektorja

$$\mathbf{a}_{c,2} = \begin{bmatrix} -l_1\dot{q}_1^2 \cos q_2 + l_1\ddot{q}_1 \sin q_2 \\ l_1\dot{q}_1^2 \sin q_2 + l_1\ddot{q}_1 \cos q_2 \end{bmatrix} + \begin{bmatrix} 0 \\ l_{c2}(\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} + \begin{bmatrix} -l_{c2}(\dot{q}_1 + \dot{q}_2)^2 \\ 0 \end{bmatrix}.$$

Seštejemo vse člene in dobimo končni rezultat  $\mathbf{a}_{c,2}$

$$\mathbf{a}_{c,2} = \begin{bmatrix} -l_1\dot{q}_1^2 \cos q_2 + l_1\ddot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 \\ l_1\dot{q}_1^2 \sin q_2 + l_1\ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) \end{bmatrix}.$$

Gravitacijski vektor  $\mathbf{g}_2$  določimo iz znanega izraza  $\mathbf{g}_2 = -\mathbf{R}_0^{2T} g \mathbf{j}$

$$\mathbf{g}_2 = -\mathbf{R}_0^{2T} g \mathbf{j} = - \begin{bmatrix} \cos(q_1 + q_2) & \sin(q_1 + q_2) & 0 \\ -\sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = g \begin{bmatrix} -\sin(q_1 + q_2) \\ -\cos(q_1 + q_2) \\ 0 \end{bmatrix}.$$

Ker imamo samo dva segmenta, ni potrebno izračunati  $\mathbf{a}_{e,2}$ . S tem je rekurzija naprej zaključena.

### 6.4.3 Rekurzija nazaj: drugi segment

Sedaj bomo s pomočjo rekurzije od vrha nazaj izračunali sile in navore v sklepih. Navore v sklepih povzročijo motorji.

Najprej uporabimo en. (6.22) ( $\mathbf{f}_i = \mathbf{R}_i^{i+1} \mathbf{f}_{i+1} + m_i \mathbf{a}_{c,i} - m_i \mathbf{g}_i$ ) pri  $i = 2$ . Poleg tega vemo, da je  $\mathbf{f}_3 = 0$ , ker ni tretjega segmenta, kinematična veriga pa je odprta.

$$\mathbf{f}_2 = m_2 \mathbf{a}_{c,2} - m_2 \mathbf{g}_2$$

Prav tako velja  $\boldsymbol{\tau}_3 = 0$  in v enačbi (6.23)

$\boldsymbol{\tau}_i = \mathbf{R}_i^{i+1} \boldsymbol{\tau}_{i+1} - \mathbf{f}_i \times \mathbf{r}_{i,ci} + (\mathbf{R}_i^{i+1} \mathbf{f}_{i+1}) \times \mathbf{r}_{i+1,ci} + \mathbf{I}_i \boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i)$  odpadejo prvi in tretji člen. Upoštevamo še  $\mathbf{r}_{2,c2} = l_{c2} \mathbf{i}$ , nakar dobimo

$$\boldsymbol{\tau}_2 = \mathbf{I}_2 \boldsymbol{\alpha}_2 + \boldsymbol{\omega}_2 \times (\mathbf{I}_2 \boldsymbol{\omega}_2) - \mathbf{f}_2 \times l_{c2} \mathbf{i}.$$

Sedaj upoštevamo še že znane  $\boldsymbol{\omega}_2 = (\dot{q}_1 + \dot{q}_2) \mathbf{k}$ ,  $\boldsymbol{\alpha}_2 = (\ddot{q}_1 + \ddot{q}_2) \mathbf{k}$ ,

$$\boldsymbol{\tau}_2 = \mathbf{I}_2 (\ddot{q}_1 + \ddot{q}_2) \mathbf{k} + (\dot{q}_1 + \dot{q}_2) \mathbf{k} \times (\mathbf{I}_2 (\dot{q}_1 + \dot{q}_2) \mathbf{k}) - \mathbf{f}_2 \times l_{c2} \mathbf{i}.$$

Drugi člen je enak 0, saj sta oba  $\boldsymbol{\omega}_2$  in  $\mathbf{I}_2 \boldsymbol{\omega}_2$  vzporedna s  $\mathbf{k}$ . Ker ima sila  $\mathbf{f}_2$  le komponenti  $\mathbf{i}$  in  $\mathbf{j}$ , je vektorski produkt  $\mathbf{f}_2 \times l_{c2} \mathbf{i}$  različen od 0 le v  $\mathbf{k}$  smeri. Za razrešitev vektorskega produkta moramo najprej izračunati  $\mathbf{f}_2$  ob upoštevanju

$$\mathbf{a}_{c,2} = \begin{bmatrix} -l_1 \dot{q}_1^2 \cos q_2 + l_1 \dot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 \\ l_1 \dot{q}_1^2 \sin q_2 + l_1 \dot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} \text{ in } \mathbf{g}_2 = g \begin{bmatrix} -\sin(q_1 + q_2) \\ -\cos(q_1 + q_2) \end{bmatrix},$$

torej

$$\mathbf{f}_2 = m_2 \begin{bmatrix} -l_1 \dot{q}_1^2 \cos q_2 + l_1 \ddot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 \\ l_1 \dot{q}_1^2 \sin q_2 + l_1 \ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} - m_2 g \begin{bmatrix} -\sin(q_1 + q_2) \\ -\cos(q_1 + q_2) \end{bmatrix},$$

$$\mathbf{f}_2 = m_2 \begin{bmatrix} -l_1 \dot{q}_1^2 \cos q_2 + l_1 \ddot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 + g \sin(q_1 + q_2) \\ l_1 \dot{q}_1^2 \sin q_2 + l_1 \ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) + g \cos(q_1 + q_2) \end{bmatrix}.$$

Nadaljujemo s  $\boldsymbol{\tau}_2$

$$\boldsymbol{\tau}_2 = \mathbf{I}_2 (\ddot{q}_1 + \ddot{q}_2) \mathbf{k} - \mathbf{f}_2 \times l_{c2} \mathbf{i}.$$

Vstavimo izraz za  $\mathbf{f}_2$

$$\boldsymbol{\tau}_2 = \mathbf{I}_2 (\ddot{q}_1 + \ddot{q}_2) \mathbf{k}$$

$$- m_2 \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ i\_komp\_f & l_1 \dot{q}_1^2 \sin q_2 + l_1 \ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) + g \cos(q_1 + q_2) & 0 \\ l_{c2} & 0 & 0 \end{vmatrix}.$$

Izraze, ki se skrivajo v  $i\_komp\_f$  naprej ne rabimo več.  $\tau_2$  zapišemo krajše

$$\begin{aligned}\tau_2 &= \mathbf{I}_2(\ddot{q}_1 + \ddot{q}_2)\mathbf{k} \\ &- [m_2 l_1 l_{c2} \dot{q}_1^2 \sin q_2 + m_2 l_1 l_{c2} \dot{q}_1 \cos q_2 + m_2 l_{c2}^2 (\ddot{q}_1 + \ddot{q}_2) + m_2 l_{c2} g \cos(q_1 + q_2)] \mathbf{k}.\end{aligned}$$

Velja torej  $\tau_2 = \tau_2 \mathbf{k}$ , kar bi lahko primerjali z Lagrangeovim zapisom.

#### 6.4.4 Rekurzija nazaj: prvi segment

Izračun bomo končali z določitvijo enačb, ki pripadajo prvemu segmentu. Še enkrat upoštevamo en. (6.22) in (6.23) pri pogoju  $i = 1$ .

$$\mathbf{f}_i = \mathbf{R}_i^{i+1} \mathbf{f}_{i+1} + m_i \mathbf{a}_{c,i} - m_i \mathbf{g}_i$$

$$\tau_i = \mathbf{R}_i^{i+1} \tau_{i+1} - \mathbf{f}_i \times \mathbf{r}_{i,ci} + (\mathbf{R}_i^{i+1} \mathbf{f}_{i+1}) \times \mathbf{r}_{i+1,ci} + \mathbf{I}_i \boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i).$$

Pri  $i = 1$  dobimo silo

$$\mathbf{f}_1 = \mathbf{R}_1^2 \mathbf{f}_2 + m_1 \mathbf{a}_{c,1} - m_1 \mathbf{g}_1$$

in navor

$$\tau_1 = \mathbf{R}_1^2 \tau_2 - \mathbf{f}_1 \times \mathbf{r}_{1,c1} + (\mathbf{R}_1^2 \mathbf{f}_2) \times \mathbf{r}_{2,c1} + \mathbf{I}_1 \boldsymbol{\alpha}_1 + \boldsymbol{\omega}_1 \times (\mathbf{I}_1 \boldsymbol{\omega}_1)$$

Upoštevamo  $\mathbf{r}_{1,c1} = l_{c1} \mathbf{i}$  in  $\mathbf{r}_{2,c1} = -(l_1 - l_{c1}) \mathbf{i}$ ,

$$\tau_1 = \mathbf{R}_1^2 \tau_2 - \mathbf{f}_1 \times l_{c1} \mathbf{i} - (\mathbf{R}_1^2 \mathbf{f}_2) \times (l_1 - l_{c1}) \mathbf{i} + \mathbf{I}_1 \boldsymbol{\alpha}_1 + \boldsymbol{\omega}_1 \times (\mathbf{I}_1 \boldsymbol{\omega}_1).$$

Sedaj lahko sam izračun poenostavimo iz dveh razlogov

1.  $\tau_2 = \mathbf{R}_1^2 \tau_2$ , ker rotacijska matrika ne vpliva na tretjo komponento vektorja.

2. Žiroskopski člen  $\boldsymbol{\omega}_1 \times (\mathbf{I}_1 \boldsymbol{\omega}_1)$  je zopet enak nič iz istega razloga kot prej, saj  $\boldsymbol{\omega}_1$  in  $(\mathbf{I}_1 \boldsymbol{\omega}_1)$  kažeta v smer  $\mathbf{k}$ .

Enačbo za silo  $\mathbf{f}_1$  vstavimo v enačbo za navor

$$\mathbf{f}_1 = \mathbf{R}_1^2 \mathbf{f}_2 + m_1 \mathbf{a}_{c,1} - m_1 \mathbf{g}_1$$

$$\tau_1 = \mathbf{R}_1^2 \tau_2 - [\mathbf{R}_1^2 \mathbf{f}_2 + m_1 \mathbf{a}_{c,1} - m_1 \mathbf{g}_1]_1 \times l_{c1} \mathbf{i} - (\mathbf{R}_1^2 \mathbf{f}_2) \times (l_1 - l_{c1}) \mathbf{i} + \mathbf{I}_1 \boldsymbol{\alpha}_1.$$

Ugotovimo, da je člen  $\mathbf{f}_2$  vpleten v dva vektorska produkta  $(\mathbf{R}_1^2 \mathbf{f}_2) \times l_{c1} \mathbf{i}$  z nasprotnim predznakom, ki se med sabo odštejeta.

$$\tau_1 = \mathbf{R}_1^2 \tau_2 - m_1 \mathbf{a}_{c,1} \times l_{c1} \mathbf{i} + m_1 \mathbf{g}_1 \times l_{c1} \mathbf{i} - (\mathbf{R}_1^2 \mathbf{f}_2) \times l_1 \mathbf{i} + \mathbf{I}_1 \boldsymbol{\alpha}_1.$$

Razrešimo vsakega od treh vektorskih produktov posebej.

Za razrešitev  $m_1 \mathbf{a}_{c,1} \times l_{c1} \mathbf{i}$  potrebujemo pospešek  $\mathbf{a}_{c,1} = \begin{bmatrix} -l_1 \dot{q}_1^2 \\ l_1 \ddot{q}_1 \end{bmatrix}$ , znan iz direktne rekurzije, ki ga vstavimo v prvi vektorski produkt,

$$m_1 \begin{bmatrix} -l_{c1} \dot{q}_1^2 \\ l_{c1} \ddot{q}_1 \end{bmatrix} \times l_{c1} \mathbf{i} = m_1 \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -l_{c1} \dot{q}_1^2 & l_{c1} \ddot{q}_1 & 0 \\ l_{c1} & 0 & 0 \end{vmatrix} = -m_1 l_{c1}^2 \ddot{q}_1 \mathbf{k}.$$

Za razrešitev  $m_1 \mathbf{g}_1 \times l_{c1} \mathbf{i}$  potrebujemo  $\mathbf{g}_1 = g \begin{bmatrix} -\sin q_1 \\ -\cos q_1 \end{bmatrix}$  prav tako znan iz direktne rekurzije, ki ga vstavimo v drugi vektorski produkt,

$$\begin{aligned} m_1 \mathbf{g}_1 \times l_{c1} \mathbf{i} &= m_1 g \begin{bmatrix} -\sin q_1 \\ -\cos q_1 \end{bmatrix} \times l_{c1} \mathbf{i} = m_1 g \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -\sin q_1 & -\cos q_1 & 0 \\ l_{c1} & 0 & 0 \end{vmatrix} \\ &= m_1 g l_{c1} \cos q_1 \mathbf{k}. \end{aligned}$$

Malo bolj komplikiran je produkt  $\mathbf{R}_1^2 \mathbf{f}_2 \times l_1 \mathbf{i}$ . Za njegov izračun rabimo

$$\mathbf{R}_1^2 = \begin{bmatrix} \cos q_2 & -\sin q_2 \\ \sin q_2 & \cos q_2 \end{bmatrix} \text{ in}$$

$$\mathbf{f}_2 = m_2 \begin{bmatrix} -l_1 \dot{q}_1^2 \cos q_2 + l_1 \ddot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 + g \sin(q_1 + q_2) \\ l_1 \dot{q}_1^2 \sin q_2 + l_1 \ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) + g \cos(q_1 + q_2) \end{bmatrix}$$

$$\mathbf{R}_1^2 \mathbf{f}_2 \times l_1 \mathbf{i}$$

$$= m_2 \begin{bmatrix} \cos q_2 & -\sin q_2 \\ \sin q_2 & \cos q_2 \end{bmatrix} \begin{bmatrix} -l_1 \dot{q}_1^2 \cos q_2 + l_1 \ddot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 + g \sin(q_1 + q_2) \\ l_1 \dot{q}_1^2 \sin q_2 + l_1 \ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) + g \cos(q_1 + q_2) \end{bmatrix}$$

Po množenju dobimo dva člena  $u_1$  in  $u_2$ .

$$\begin{aligned} u_1 &= m_2 \cos q_2 \left[ -l_1 \dot{q}_1^2 \cos q_2 + l_1 \ddot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 + g \sin(q_1 + q_2) \right] \\ &\quad - m_2 \sin q_2 \left[ l_1 \dot{q}_1^2 \sin q_2 + l_1 \ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) + g \cos(q_1 + q_2) \right] \end{aligned}$$

$$\begin{aligned} u_2 &= m_2 \sin q_2 \left[ -l_1 \dot{q}_1^2 \cos q_2 + l_1 \ddot{q}_1 \sin q_2 - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 + g \sin(q_1 + q_2) \right] \\ &\quad + m_2 \cos q_2 \left[ l_1 \dot{q}_1^2 \sin q_2 + l_1 \ddot{q}_1 \cos q_2 + l_{c2}(\ddot{q}_1 + \ddot{q}_2) + g \cos(q_1 + q_2) \right] \end{aligned}$$

Oba člena je mogoče še okrajšati, vendar ker bomo v nadaljevanju rabili le

drugega, zato okrajšamo le tega

$$\begin{aligned} u_2 = m_2 & \left[ -l_1 \dot{q}_1^2 \sin q_2 \cos q_2 + l_1 \ddot{q}_1 \sin^2 q_2 \right. \\ & - l_{c2} (\dot{q}_1 + \dot{q}_2)^2 \sin q_2 + g \sin q_2 \sin(q_1 + q_2) \left. \right] \\ & + m_2 \left[ l_1 \dot{q}_1^2 \sin q_2 \cos q_2 + l_1 \ddot{q}_1 \cos^2 q_2 \right. \\ & \left. + l_{c2} (\ddot{q}_1 + \ddot{q}_2) \cos q_2 + g \cos q_2 \cos(q_1 + q_2) \right] \end{aligned}$$

Prvi člen v prvi in prvi člen v drugi vrstici se odštejeta. Drugi člen v prvi vrstici in drugi člen v drugi vrstici imata isto predpono, vsota  $l_1 \ddot{q}_1 (\sin^2 q_2 + \cos^2 q_2) = l_1 \ddot{q}_1$ .

Zadnja dva člena v prvi in drugi vrstici pa razstavimo. Iz trigonometrije sledi

$$\begin{aligned} \sin \alpha \sin \beta &= \frac{1}{2} [\cos(\alpha - \beta) - \cos(\alpha + \beta)] \\ \cos \alpha \cos \beta &= \frac{1}{2} [\cos(\alpha - \beta) + \cos(\alpha + \beta)] \end{aligned}$$

torej je vsota

$$\sin \alpha \sin \beta + \cos \alpha \cos \beta = \frac{1}{2} [\cos(\alpha - \beta) - \cos(\alpha + \beta) + \cos(\alpha - \beta) + \cos(\alpha + \beta)].$$

Ostaneta prvi in tretji, odštejeta se drugi in četrti člen

$$\sin \alpha \sin \beta + \cos \alpha \cos \beta = \cos(\alpha - \beta)$$

torej  $\alpha = q_2$  in  $\beta = q_1 + q_2$

$$\sin \alpha \sin \beta + \cos \alpha \cos \beta = \cos(\alpha - \beta) = \cos q_1.$$

Končno pišemo tretji vektorski produkt v obliki

$$\begin{aligned} \mathbf{R}_1^2 \mathbf{f}_2 \times l_1 \mathbf{i} &= \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \times l_1 \mathbf{i} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_1 & u_2 & 0 \\ l_1 & 0 & 0 \end{vmatrix} = -l_1 u_2 \\ &= -\left( l_1 \ddot{q}_1 - l_{c2} (\dot{q}_1 + \dot{q}_2)^2 \sin q_2 + l_{c2} (\ddot{q}_1 + \ddot{q}_2) \cos q_2 + g \cos q_1 \right) l_1 m_2. \end{aligned}$$

In nazadnje določimo še člen  $\mathbf{I}_1 \boldsymbol{\alpha}_1$  pri  $\boldsymbol{\alpha}_1 = \ddot{q}_1 \Rightarrow \mathbf{I}_1 \ddot{q}_1$ .

Sedaj vse posamezne člene v  $\boldsymbol{\tau}_1$  vstavimo v skupno enačbo z elementi v smeri  $\mathbf{k}$

$$\boldsymbol{\tau}_1 = \boldsymbol{\tau}_2 + m_1 l_{c1}^2 \ddot{q}_1 + m_1 g l_{c1} \cos g_1 + \mathbf{I}_1 \ddot{q}_1 - \mathbf{R}_1^2 \mathbf{f}_2 \times l_1 \mathbf{i}.$$

In še izraz za

$$-\boldsymbol{R}_1^2 \boldsymbol{f}_2 \times l_1 \boldsymbol{i} = \left( l_1 \ddot{q}_1 - l_{c,2} (\dot{q}_1 + \dot{q}_2)^2 \sin q_2 + l_{c,2} (\ddot{q}_1 + \ddot{q}_2) \cos q_2 + g \cos q_1 \right) l_1 m_2.$$

Končno dobimo moment

$$\begin{aligned} \boldsymbol{\tau}_1 &= \boldsymbol{\tau}_2 + m_1 l_{c1}^2 \ddot{q}_1 + m_1 g l_{c1} \cos q_1 + \boldsymbol{I}_1 \ddot{q}_1 + l_1^2 m_1 \ddot{q}_1 \\ &\quad - l_1 l_{c,2} m_2 (\dot{q}_1 + \dot{q}_2)^2 \sin q_2 + l_1 l_{c,2} m_2 (\ddot{q}_1 + \ddot{q}_2) \cos q_2 + l_1 m_2 g \cos q_1. \end{aligned}$$

# Literatura

- [Bajd, 2002] Bajd, T. (2002). Osnove robotike. Fakulteta za elektrotehniko.
- [Bartels et al., 1995] Bartels, Richard H, Beatty, John C, in Barsky, Brian A (1995). An introduction to splines for use in computer graphics and geometric modeling. Morgan Kaufmann.
- [Calinon, 2016] Calinon, Sylvain (2016). A tutorial on task-parameterized movement learning and retrieval. Intelligent service robotics, 9(1):1–29.
- [Calinon et al., 2007] Calinon, Sylvain, Guenter, Florent, in Billard, Aude (2007). On learning, representing, and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37(2):286–298.
- [Calinon in Lee, 2017] Calinon, Sylvain in Lee, Dongheui (2017). Learning control. mediatum. ub. tum. de. [Online]. Available: <https://mediatum.ub.tum.de/doc/1432385/file.pdf>. [Accessed: 17-Aug-2021].
- [Deniša et al., 2015] Deniša, Miha, Gams, Andrej, Ude, Aleš, in Petrič, Tadej (2015). Learning compliant movement primitives through demonstration and statistical generalization. IEEE/ASME transactions on mechatronics, 21(5):2581–2594.
- [Developer, 2021] Developer, D (2021). Bézier curves. developer.roblox.com. [Online]. Available: <https://developer.roblox.com/en-us/articles/Bezier-curves>. [Accessed: 17-Aug-2021].
- [Forte et al., 2012] Forte, Denis, Gams, Andrej, Morimoto, Jun, in Ude, Aleš (2012). On-line motion synthesis and adaptation using a trajectory database. Robotics and Autonomous Systems, 60(10):1327–1339.
- [Huang et al., 2019] Huang, Yanlong, Rozo, Leonel, Silvério, João, in Caldwell, Darwin G (2019). Kernelized movement primitives. The International Journal of Robotics Research, 38(7):833–852.

- [Ijspeert et al., 2013] Ijspeert, Auke Jan, Nakanishi, Jun, Hoffmann, Heiko, Pastor, Peter, in Schaal, Stefan (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373.
- [Ijspeert et al., 2001] Ijspeert, Auke Jan, Nakanishi, Jun, in Schaal, Stefan (2001). Trajectory formation for imitation with nonlinear dynamical systems. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No. 01CH37180)*, volume 2, strani 752–757. IEEE.
- [Kamermans, 2021] Kamermans, Mike (2021). A primer on bézier curves. [Pomax.github.io.\[Online\]](https://pomax.github.io/bezierinfo/). Available: [https://pomax.github.io/bezierinfo.](https://pomax.github.io/bezierinfo/) [Accessed: 17-Aug-2021].
- [Murray et al., 2017] Murray, Richard M, Li, Zexiang, in Sastry, S Shankar (2017). *A mathematical introduction to robotic manipulation*. CRC press.
- [Paraschos et al., 2013] Paraschos, Alexandros, Daniel, Christian, Peters, Jan, Neumann, Gerhard, et al. (2013). Probabilistic movement primitives. *Advances in neural information processing systems*.
- [Paul, 1981] Paul, R.P. (1981). *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press.
- [Righetti et al., 2006] Righetti, Ludovic, Buchli, Jonas, in Ijspeert, Auke Jan (2006). Dynamic hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena*, 216(2):269–281.
- [Schaal et al., 2007] Schaal, Stefan, Mohajerian, Peyman, in Ijspeert, Auke (2007). Dynamics systems vs. optimal control—a unifying view. *Progress in brain research*, 165:425–445.
- [Schiavicco L., 2002] Schiavicco L., Siciliano B. (2002). *Modelling and Control of Robot Manipulators*. Springer.
- [Study, 2021] Study, Wolf (2021). Dynamic movement primitives part 1: The basics. [studywolf.wordpress.com.\[Online\]](https://studywolf.wordpress.com/2013/11/16/dynamic-movement-primitives-part-1-the-basics/). Available: <https://studywolf.wordpress.com/2013/11/16/dynamic-movement-primitives-part-1-the-basics/>. [Accessed: 17-Aug-2021].
- [Ude et al., 2010] Ude, Aleš, Gams, Andrej, Asfour, Tamim, in Morimoto, Jun (2010). Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815.

# GRŠKA ABECEDA

$A$	$\alpha$	alfa	$I$	$\iota$	jota	$R$	$\rho$	rho
$B$	$\beta$	beta	$K$	$\kappa$	kapa	$\Sigma$	$\sigma$	sigma
$\Gamma$	$\gamma$	gama	$\Lambda$	$\lambda$	lambda	$T$	$\tau$	tau
$\Delta$	$\delta$	delta	$M$	$\mu$	mi	$\Upsilon$	$\upsilon$	ipsilon
$E$	$\epsilon$	epsilon	$N$	$\nu$	ni	$\Phi$	$\phi$	fi
$Z$	$\zeta$	zeta	$\Xi$	$\xi$	ksi	$X$	$\chi$	hi
$H$	$\eta$	eta	$O$	$\circ$	omikron	$\Psi$	$\psi$	psi
$\Theta$	$\theta$	theta	$\Pi$	$\pi$	pi	$\Omega$	$\omega$	omega