~~Robot: előírt mozgás~~

1. ROBOTIRÁNYÍTÁSI ALAPISMERETEK

1.1. A robot, mint irányítandó folyamat.

Előírt mozgás

- Pont-pont irányítás : pálya nincs előírva
- Folytonos pályairányítás

6DOF robot szabtech.-tulajdonságai

- 6 motor rpm/áram szab.: beállítás, tartás
- Csukló egymástól függő koordinátája, mozgása
- Előírt pályát tartani kell

Robot generációk

1. Pályaleíró program szerint vezérli egy mg. [illegible]
2. Környezetüket szenzorokkal érzékelik és ez befolyásolja is a viselkedésüket, pl. akadálykerülés
3. Mesterséges intelligencia, jó alkalmazkodás, hangvezérlés stb.

1.2 Egy kis fejszámolás

Szóhossz, pipeline, lebegőpontos egység befolyásolja a sebességet.

1.3 A robot geometriája – direkt geometriai feladat

1.3.1 Lineáris transzformációk

$$\underline{\underline{A}} = [\underline{a}\times]$$

$$\underline{a}\times\underline{i} = \begin{vmatrix} \underline{i} & \underline{j} & \underline{k} \\ a_x & a_y & a_z \\ 1 & 0 & 0 \end{vmatrix} = \begin{bmatrix} 0 \\ a_z \\ -a_y \end{bmatrix};\ \underline{a}\times\underline{j} = \begin{bmatrix} -a_z \\ 0 \\ a_x \end{bmatrix};\ \underline{a}\times\underline{k} = \begin{bmatrix} a_y \\ -a_x \\ 0 \end{bmatrix}$$

$$\underline{\underline{A}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

$$\underline{\underline{A}} = \underline{t}\langle \underline{t}, \cdot \rangle \quad (\underline{t} \text{ egységv.})$$

$$\underline{\underline{A}} = \underline{a}\langle \underline{b}, \cdot \rangle = [\underline{a} \circ \underline{b}]$$

$$\underline{a}\langle \underline{b}, \underline{i} \rangle = \underline{a}\, b_x$$

$$\underline{a}\langle \underline{b}, \underline{j} \rangle = \underline{a}\, b_y$$

$$\underline{a}\langle \underline{b}, \underline{k} \rangle = \underline{a}\, b_z$$

$$\underline{\underline{A}} = \begin{bmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{bmatrix}$$

RRT

~~Robot: előírt mozgás~~

1. ROBOTIRÁNYÍTÁSI ALAPISMERETEK

1.1. A robot, mint irányítandó folyamat.

Előírt mozgás

- Pont-pont irányítás: pálya nincs előírva
- Folytonos pályairányítás

6DOF robot szabtech.-tulajdonságai

- 6 motor rpm/áram szab.: beállítás, tartás
- Csukló egymástól függő koordinátája, mozgása
- Előírt pályát tartani kell

Robot generációk

1. Pályaleíró program szerint vezérli egy mg. [illegible]
2. Környezetüket szenzorokkal érzékelik és ez befolyásolja is a viselkedésüket, pl. akadálykerülés
3. Mesterséges intelligencia, jó alkalmazkodás, hangvezérlés stb.

1.2 Egy kis fejszámolás

Szóhossz, pipeline, lebegőpontos egység befolyásolja a sebességet.

1.3 A robot geometriája - direkt geometriai feladat

1.3.1 Lineáris transzformációk

$\underline{\underline{A}} = [\underline{a}\times]$

$$\underline{a}\times\underline{i} = \begin{vmatrix} \underline{i} & \underline{j} & \underline{k} \\ a_x & a_y & a_z \\ 1 & 0 & 0 \end{vmatrix} = \begin{bmatrix} 0 \\ a_z \\ -a_y \end{bmatrix};\ \underline{a}\times\underline{j} = \begin{bmatrix} -a_z \\ 0 \\ a_x \end{bmatrix};\ \underline{a}\times\underline{k} = \begin{bmatrix} a_y \\ -a_x \\ 0 \end{bmatrix}$$

$$\underline{\underline{A}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

$\underline{\underline{A}} = \underline{t}\langle \underline{t}, \cdot\rangle$ ($\underline{t}$ egységv.)

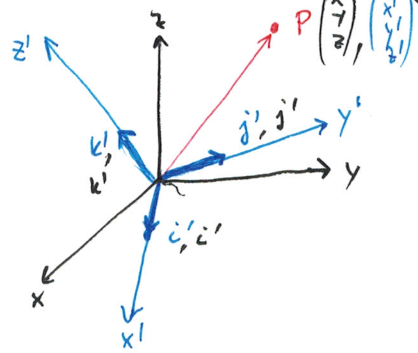
$\underline{\underline{A}} = \underline{a}\langle \underline{b}, \cdot\rangle = [\underline{a} \circ \underline{b}]$

$$\underline{a}\langle \underline{b}, \underline{i}\rangle = \underline{a}\, b_x$$
$$\underline{a}\langle \underline{b}, \underline{j}\rangle = \underline{a}\, b_y$$
$$\underline{a}\langle \underline{b}, \underline{k}\rangle = \underline{a}\, b_z$$

$$\underline{\underline{A}} = \begin{bmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{bmatrix}$$

1.3.2. Koordináta-transzformáció

(K) (K') P



(K') → (K)

$$\begin{array}{ccc} \mathbb{R}^3 & \xrightarrow{I} & \mathbb{R}^3 \\ \begin{bmatrix} i' \\ j' \\ k' \end{bmatrix} & & \begin{bmatrix} i \\ j \\ k \end{bmatrix} \\ \mathbb{R}^3 & \xrightarrow{\underline{\underline{A}}} & \mathbb{R}^3 \end{array}$$

$I(i') = i'$

$I(j') = j'$

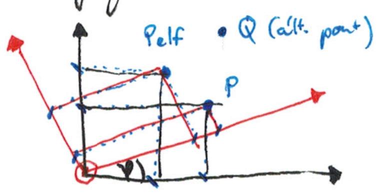
$I(k') = k'$

$\underline{\underline{A}} = [i' \; j' \; k']$

$\underline{\underline{A}}$: (K' orientációja az álló (K-hoz képest.
és ugyanígy az álló P pont koordinátaleírásával kapcsolatát is megadja.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \underline{\underline{A}} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Pl. forgatás z tengely körül φ szöggel



(K) (K')

$$\begin{array}{ccc} \mathbb{R}^3 \; P & \xrightarrow{Rot^z_\varphi} & \mathbb{R}^3 \; P_{elf} \\ \begin{bmatrix} i \\ j \\ k \end{bmatrix} & & \begin{bmatrix} i \\ j \\ k \end{bmatrix} \\ \mathbb{R}^3 & \xrightarrow{\underline{\underline{R}}^z_\varphi} & \mathbb{R}^3 \end{array}$$

$$\begin{bmatrix} x_{elf} \\ y_{elf} \\ z_{elf} \end{bmatrix} = \underline{\underline{R}}^z_\varphi \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x'_{elf} \\ y'_{elf} \\ z'_{elf} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x_{elf} \\ y_{elf} \\ z_{elf} \end{bmatrix} = \underline{\underline{R}}^z_\varphi \begin{bmatrix} x'_{elf} \\ y'_{elf} \\ z'_{elf} \end{bmatrix}$$

$\underline{\underline{R}}^z_\varphi$: ~~K-ban~~ megadja K' orientációját K-hoz képest, ill. adott pontot ezen orientáció szerint transzformál

és: K'-beli pont leírását megadja K-ban.

$$\boxed{\begin{array}{ccc} \mathbb{R}^3 \, (K) & \xrightarrow{Rot^z_\varphi} & \mathbb{R}^3 \, (K) \\ P & \underline{\underline{R}}^z_\varphi & P_{elf} \end{array}} \quad \boxed{\begin{array}{ccc} \mathbb{R}^3 \, (K') & I & \mathbb{R}^3 \, (K) \\ Q & \xrightarrow{\underline{\underline{R}}^z_\varphi} & Q \end{array}}$$

$$\underline{\underline{R}}^x_\varphi = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & c_\varphi & -s_\varphi \\ \cdot & s_\varphi & c_\varphi \end{bmatrix} \qquad \underline{\underline{R}}^y_\varphi = \begin{bmatrix} c_\varphi & \cdot & s_\varphi \\ \cdot & 1 & \cdot \\ -s_\varphi & \cdot & c_\varphi \end{bmatrix} \qquad \underline{\underline{R}}^z_\varphi = \begin{bmatrix} c_\varphi & -s_\varphi & \cdot \\ s_\varphi & c_\varphi & \cdot \\ \cdot & \cdot & 1 \end{bmatrix}$$

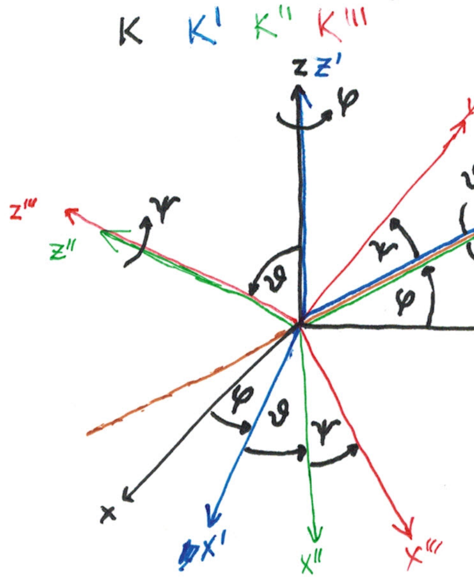
1.3.3 Rodrigues-képlet

t egységvektor ($|t| = 1$) menti forgatás

$$\underline{\underline{R}}_{\varphi}^{\underline{t}} = C_\varphi \cdot \underline{\underline{I}} + (1 - C_\varphi)\cdot[\underline{t}\circ\underline{t}] + S_\varphi \cdot [\underline{t}\times]$$

$$\left(\underline{\underline{R}}_{\varphi}^{\underline{t}}\right)^{-1} = \underline{\underline{R}}_{-\varphi}^{\underline{t}} = \underline{\underline{R}}_{\varphi}^{-\underline{t}} = \left(\underline{\underline{R}}_{\varphi}^{\underline{t}}\right)^T$$

1.3.4 Az orientáció jellemzése Euler-szögekkel



1. $t = [x, y] \cap [x''', y''']$
(xy sík metszésvonala K és K'''-ban)
2. K': K-hoz képest z tengely körül φ-vel úgy, hogy y' és t egybeessen (t per def benne van xy-ban)
3. K'': K'-höz képest y' körül ϑ szöggel úgy, hogy z'' és z''' egybeessen ($z' = z \perp t$ és $z'' = z''' \perp t$ ∴ lehetséges)
4: K''': K''-höz képest z'' körül elforgatva ψ-vel úgy, hogy y'' és y''' egybeessen ($t \in [x''', y''']$ és erre merőleges $z'' = z'''$)

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = Rot(z'', \psi) \begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = Rot(y', \vartheta) \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = Rot(z, \varphi) \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = Rot(z, \varphi)\, Rot(y', \vartheta)\, Rot(z'', \psi) \begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} C_\varphi & -S_\varphi & \cdot \\ S_\varphi & C_\varphi & \cdot \\ \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} C_\vartheta & \cdot & S_\vartheta \\ \cdot & 1 & \cdot \\ -S_\vartheta & \cdot & C_\vartheta \end{bmatrix} \begin{bmatrix} C_\psi & -S_\psi & \cdot \\ S_\psi & C_\psi & \cdot \\ \cdot & \cdot & 1 \end{bmatrix}}_{} \begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix}$$

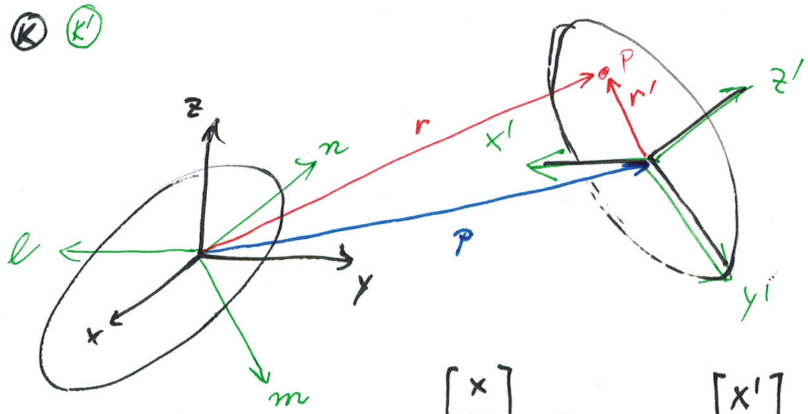
<u>Euler</u>$(\varphi, \vartheta, \psi)$ → direkt feladat

Inverz feladat: adott mx-ből Euler-szögek

Ha 6-DOF robotban utolsó három szegmense Euler-kézcsukló, akkor egyszerűsödik az inverz geometriai feladat (poz + orient.)

1.3.5. Merev testek relatív helyzetének jellemzése homogén koordinátákkal.

Ⓚ Ⓚ'



$(\underline{l},\underline{m},\underline{n})$ párh. (x',y',z')-vel

$$\underline{r} = \underline{r}' + \underline{p}$$

P pont K bázisban:

$$\begin{bmatrix}x\\y\\z\end{bmatrix} = [\underline{l}\ \underline{m}\ \underline{n}]\begin{bmatrix}x'\\y'\\z'\end{bmatrix} + \underline{p} = x'\underline{l} + y'\underline{m} + z'\underline{n} + \underline{p}$$

$(\underline{l},\underline{m},\underline{n})$ K bázisban van felírva természetesen és $\underline{p}$ is

$$\begin{bmatrix}x\\y\\z\\1\end{bmatrix} = \left[\begin{array}{ccc|c}\underline{l} & \underline{m} & \underline{n} & \underline{p}\\ \cdot & \cdot & \cdot & 1\end{array}\right]\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix} = \left[\begin{array}{c|c}\underline{\underline{A}} & \underline{p}\\ \hline \underline{0}^T & 1\end{array}\right]\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix}$$

Tehát K és K' relatív helyzete leírható a köv. mx-al:

$$\underline{\underline{T}}_{KK'} = \begin{bmatrix}\underline{\underline{A}} & \underline{p}\\ \underline{0}^T & 1\end{bmatrix} = \begin{bmatrix}\underline{l} & \underline{m} & \underline{n} & \underline{p}\\ \cdot & \cdot & \cdot & 1\end{bmatrix}$$

Ez a „homogén koordinátás mátrix" („homogén mátrix")

És persze a K→K' helyzeten kívül megadja a K'-beli pontok K-beli leírását

Azonosságok homogén mx-okkal

$$\begin{bmatrix}\underline{\underline{A}}_1 & \underline{p}_1\\ \underline{0}^T & 1\end{bmatrix}\begin{bmatrix}\underline{\underline{A}}_2 & \underline{p}_2\\ \underline{0}^T & 1\end{bmatrix} = \begin{bmatrix}\underline{\underline{A}}_1\underline{\underline{A}}_2 & \underline{\underline{A}}_1\underline{p}_2 + \underline{p}_1\\ \underline{0}^T & 1\end{bmatrix}$$

$$\begin{bmatrix}\underline{\underline{I}} & \underline{p}\\ \underline{0}^T & 1\end{bmatrix}\begin{bmatrix}\underline{\underline{A}} & 0\\ \underline{0}^T & 1\end{bmatrix} = \begin{bmatrix}\underline{\underline{A}} & \underline{p}\\ \underline{0}^T & 1\end{bmatrix} \qquad K \xrightarrow[\underline{p}]{\text{eltolás}} K' \xrightarrow[\underline{\underline{A}}]{\text{forg}} K''$$

$$\begin{bmatrix}\underline{\underline{A}} & 0\\ \underline{0}^T & 1\end{bmatrix}\begin{bmatrix}\underline{\underline{I}} & \underline{p}\\ \underline{0}^T & 1\end{bmatrix} = \begin{bmatrix}\underline{\underline{A}} & \underline{\underline{A}}\underline{p}\\ \underline{0}^T & 1\end{bmatrix} \qquad K \xrightarrow[\underline{\underline{A}}]{\text{forg}} K' \xrightarrow[\underline{p}]{\text{eltolás}} K''$$

$$\begin{bmatrix}\underline{\underline{A}} & \underline{p}\\ \underline{0}^T & 1\end{bmatrix}^{-1} = \begin{bmatrix}\underline{\underline{A}}^{-1} & -\underline{\underline{A}}^{-1}\underline{p}\\ \underline{0}^T & 1\end{bmatrix}$$

Ha $\underline{\underline{A}}$ :: forgatás $\underline{\underline{A}}^{-1} = \underline{\underline{A}}^T$

Ha $\underline{\underline{A}} = A_1 \cdot A_2 \cdot \ldots$ :: forgatások

$\underline{\underline{A}}^{-1} = \ldots \cdot A_2^{-1} \cdot A_1^{-1} = \ldots \cdot A_2^T \cdot A_1^T$

1.3.6. Merev, nyílt láncú elágazás nélküli robot irányítása

Robotszegmensek: 0..m

Csuklóváltozók: $q_1 .. q_m$ ($q_i \to$ i. szegmens) T: d, a R: ϑ, α

K_B: abszolút bázis, világkoordináta K_E: végberendezés koordinátarendszere

Tengelyek: $t_0 .. t_{m-1}$ ($t_i \to q_{i+1}$ változó) i. szegmens)

K_i keret helyzetének leírása

Ált 0..3: „kar"
4..6: „csukló"
végberendezés: „kézfej"

$$\underline{\underline{T}}_{K_{i-1},K_i} \triangleq \underline{\underline{T}}_{i-1,i}$$

Konstans trafók:

$$\underline{\underline{T}}_{B,0} \qquad \underline{\underline{T}}_{m,E}$$

$$\underline{\underline{T}}_{0,m} = \underline{\underline{T}}_{0,1} \cdot \underline{\underline{T}}_{1,2} \cdot \ldots \cdot \underline{\underline{T}}_{m-1,m} = \begin{bmatrix} \underline{\underline{A}}_{0,m} & \underline{p}_{0,m} \\ \underline{0}^T & 1 \end{bmatrix} = \prod_{i=0}^{m-1} \underline{\underline{T}}_{i,i+1}$$

$$\underline{\underline{T}}_{i,i-1} \triangleq \underline{\underline{T}}_{i-1,i}^{-1}$$

$$\begin{tikzcd} K_B \arrow[r, "T_{B,0}"] & K_0 \arrow[r, "T_{0,m}"] & K_m \arrow[r, "T_{m,E}"] & K_E \arrow[lll, bend left, "T_{B,E}"] \end{tikzcd}$$

robot transzformációs gráf

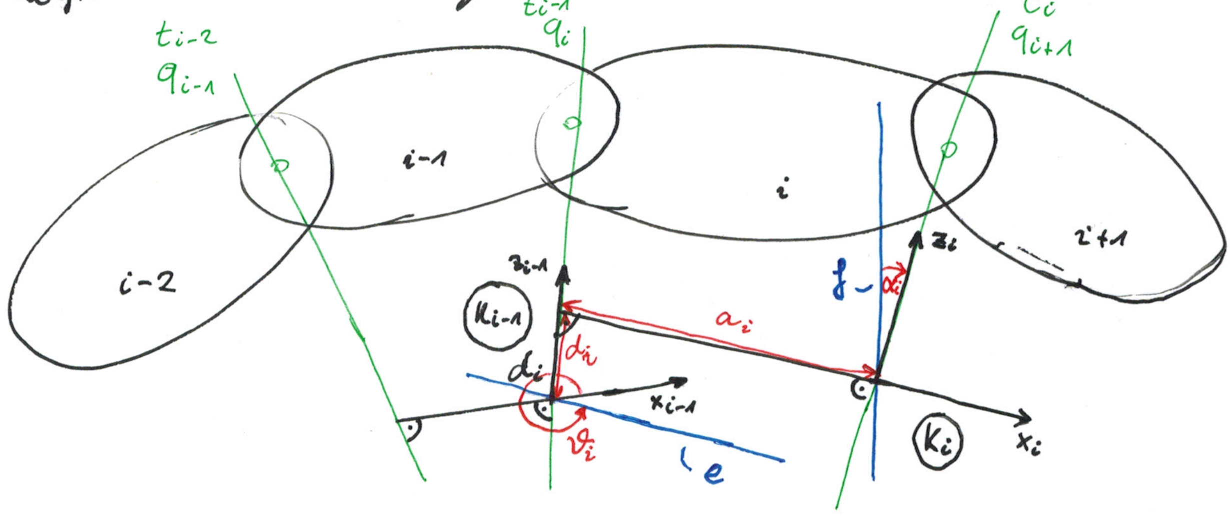
$$\underline{\underline{T}}_{B,E} = \underline{\underline{T}}_{B,0} \cdot \underline{\underline{T}}_{0,m} \cdot \underline{\underline{T}}_{m,E}$$

$$\underline{\underline{T}}_{0,m} = \underline{\underline{T}}_{B,0}^{-1} \cdot \underline{\underline{T}}_{B,E} \cdot \underline{\underline{T}}_{m,E}^{-1}$$

Inverz feladatnál: $\underline{\underline{T}}_{B,0}$; $\underline{\underline{T}}_{m,E}$ ismert, $\underline{\underline{T}}_{B,E}$ előírt és a $\underline{\underline{T}}_{0,m}$-et előállító q_i-ket keressük $\forall i \in 1..m$.

1.3.7. Robotok leírása Denavit–Hartenberg-alak segítségével

Cél: konvenciók bevezetésével szomszédos szegmensek relatív helyzetének leírásához szükséges változók számának 4-re csökkentése 6-ról



Kikötések:

- Rotációs csukló forgástengelye z-irányú
- Transzlációs csukló elmozdulása z-irányú

Feltételezések:

- K_{i-1} ismert
- z_{i-1} iránya: [i-1] és [i] közti t_{i-1} iránya
- K_i keret x_0 tengelye merőleges t_{i-1} és t_i tengelyekre
- K_i origója x_i és t_i metszéspontja
- z_i: t_i iránya

DH-paraméterek:

ϑ_i: szög amivel z_{i-1} körül ~~el kell~~ elforgatva x_{i-1}-ből x_i-t adja, ill. vele párhuzamost.

d_i: az előző elforgatás után x_{i-1} ~~el...~~ (e) és x_i távolsága.

a_i: az előző után z_{i-1} és x_i metszéspontjának távolsága K_i origójától
(x_i és z_i metszéspontja; távolság/eltolás x_i mentén)

α_i: szög, amivel x_i körül ~~el kell forgatni~~ elforgatva z_{i-1}-et z_i-vel párhuzamost kapunk (előzőek után K_{i-1} K_i-vel egybeesik)

$$\underline{T_{i-1,i}} = Rot(z, \vartheta_i) \cdot Trans(z, d_i) \cdot Trans(x, a_i) \cdot Rot(x, \alpha_i)$$

$$= \begin{bmatrix} C\vartheta_i & -S\vartheta_i & \cdot & \cdot \\ S\vartheta_i & C\vartheta_i & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & d_i \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & a_i \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & C\alpha_i & -S\alpha_i & \cdot \\ \cdot & S\alpha_i & C\alpha_i & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

Spec. esetek

- t_{i-1} és t_i párhuzamos: d_i tetszőlegesen választható x_i-vel
- [0] esetén $z_0 = t_0$ és x_0 tetszőleges csak $z_0 \perp x_0$ legyen.
- [m] esetén z_m tetszőleges és $x_m \perp z_{m-1}$

Pl. Puma-560

i	q_i	ϑ_i	d_i	a_i	α_i
1	∘ ϑ_1	ϑ_1	0	0	$-90°$
2	∘ ϑ_2	ϑ_2	d_2	a_2	$0°$
3	∘ ϑ_3	ϑ_3	0	0	$90°$
4	∘ ϑ_4	ϑ_4	d_4	0	$-90°$
5	∘ ϑ_5	ϑ_5	0	0	$90°$
6	∘ ϑ_6	ϑ_6	0	0	0

O változó
— konstans

Változók teljesen meghatározzák a $T_{0,m}$ mx-ot.

Direkt geom feladat:

$$q=\begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \rightarrow T_{0,m}=\begin{bmatrix} A_{0,m} & p_{0,m} \\ 0 & 1 \end{bmatrix}$$

$$T_{B,E}=T_{B,0}\,T_{0,m}\,T_{m,E}$$

Inverz geometriai feladat:

$$T_{0,m}=T_{B,0}^{-1}\cdot T_{B,E}\cdot T_{m,E}^{-1}$$

$$T_{0,m}=\begin{bmatrix} A_{0,m} & p_{0,m} \\ 0 & 1 \end{bmatrix} \rightarrow q=\begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix}$$ Ez erősen robotspecifikus.

Puma-560 levezetés (példa)

K_0: x_0, y_0, z_0-al adott. (z_0 a tengely mentén, x_0 tetszőleges)

K_1: z_1 tengelyirányba néz $x_1 \parallel z_0 \times z_1$, de mondjuk $x_1 \neq z_0 \times z_1$

ϑ_1: változó ($0°$)

$d_1 = 0$

$a_1 = 0$

$\alpha_1 = -90°$ (igazából itt is fordítva kéne)

$x_1^* = z_0 \times z_1$ $\quad \vartheta_1^* =$ változó ($+180°$)

$d_1^* = 0$

$a_1^* = 0$

$\alpha_1^* = 90°$

ϑ_2: változó ($0°$)

$d_2 =$ konst d_2

$a_2 =$ konst a_2

$\alpha_2 = 0°$

$\vartheta_2^* =$ változó ($180°$)

$d_2^* =$ konst d_2

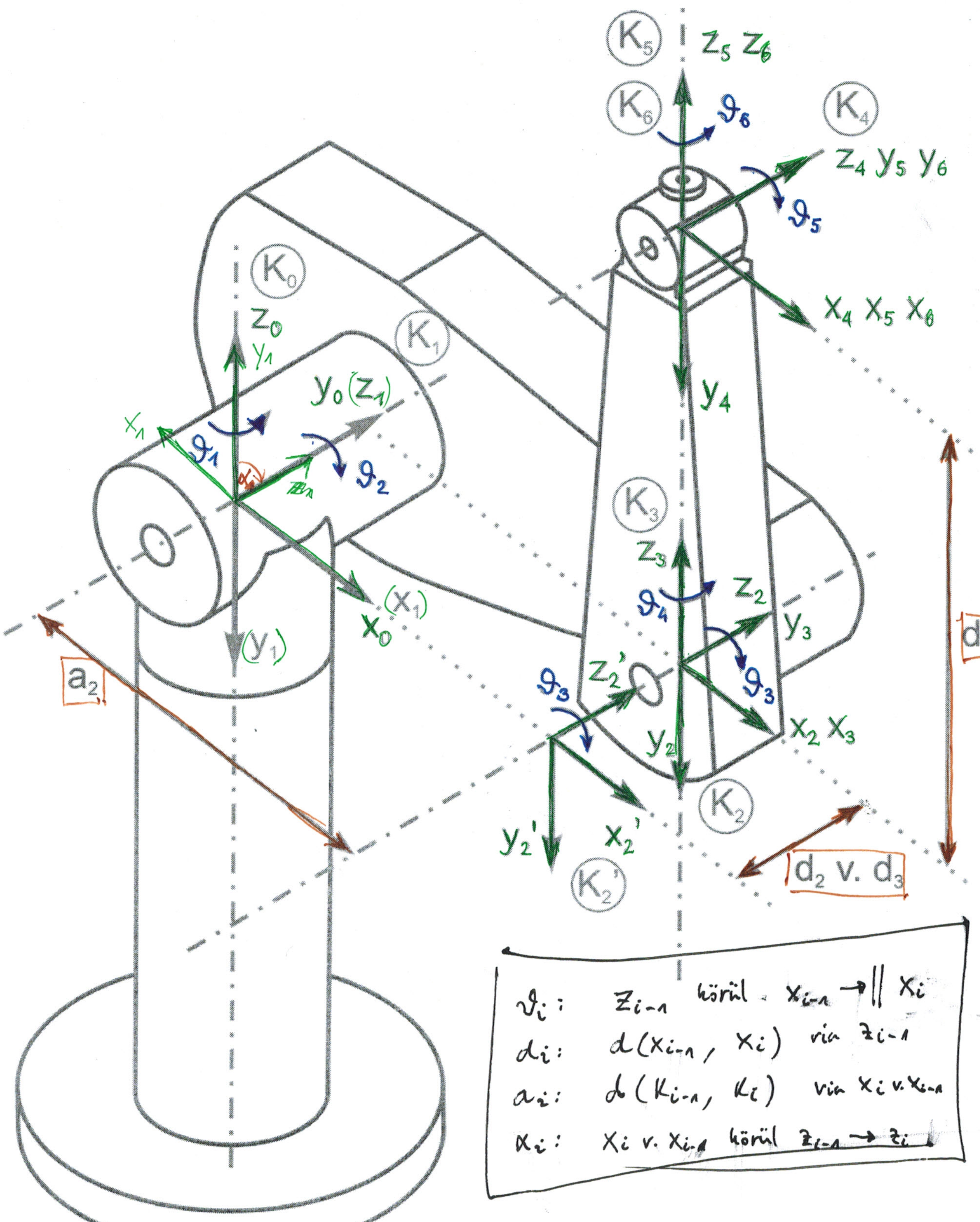
$a_2^* =$ konst a_2

$\alpha_2^* = 0°$ $\quad$ innentől $K_3 = K_3^*$

~~$\vartheta_2' =$ változó ($0°$)~~

RKT

~~kereteit, amelyek segítségével meghatározhatók a robot Den~~
DH) paraméterei, így könnyedén felírhatóak lesznek $\mathbf{T}_{i-1,i}$ má



1.4.1 ábra – A Puma 560-as robot koordinátarendszerei

eghatározása:

K_3: ϑ_3: változó
d_3: 0
d_4: 0
α_3: 90°

K_4: ϑ_4: változó
d_4: konst d_4
a_4: 0
α_4: −90°

K_5: ϑ_5: változó
d_5: 0
a_5: 0
α_5: 90°

K_6: ϑ_6: változó
d_6: 0
a_6: 0
α_6: 0°

$$T_{3,6} = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & d_4 \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} \left[\begin{array}{c|c} \underline{\underline{Euler}}(\vartheta_4, \vartheta_5, \vartheta_6) & \underline{0} \\ \hline \underline{0}^T & 1 \end{array}\right] = \begin{bmatrix} \underline{\underline{Euler}}(\vartheta_4, \vartheta_5, \vartheta_6) \\ \underline{0}^T \end{bmatrix}$$

$$T_{0,6} = T_{0,3} \cdot \begin{bmatrix} \underline{\underline{Euler}}(\vartheta_4, \vartheta_5, \vartheta_6) & \begin{matrix} 0 \\ 0 \\ d_4 \end{matrix} \\ \underline{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \underline{l} & \underline{m} & \underline{n} & \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,6} = \begin{bmatrix} l_{0,3} & m_{0,3} & n_{0,3} & p_{0,3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{\underline{Euler}}(\vartheta_4, \vartheta_5, \vartheta_6) & \begin{matrix} 0 \\ 0 \\ d_4 \end{matrix} \\ \underline{0}^T & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \underline{\underline{A}}_{0,3} \cdot \underline{\underline{Euler}}(\vartheta_4, \vartheta_5, \vartheta_6) & p_{0,3} + d_4 \cdot n_{0,3} \\ \underline{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \underline{l} & \underline{m} & \underline{n} & \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1.5. Az inverz geometriai feladat megoldása Puma 560 robotra

I/52. oldalon ábra sormán van

Legyen $\underline{\underline{A}}_{3,6} = \underline{\underline{Euler}}(\vartheta_4, \vartheta_5, \vartheta_6)$

$$T_{0,6} = \begin{bmatrix} \underline{\underline{A}}_{0,3} \cdot \underline{\underline{A}}_{3,6} & \underline{p}_{0,3} + d_4 \cdot \underline{n}_{0,3} \\ \underline{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \underline{l} & \underline{m} & \underline{n} & \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\underline{p} = \underline{p}_{0,3}(q_1, q_2, q_3) + d_4 \cdot \underline{n}_{0,3}(q_1, q_2, q_3)$$

Feladat: $\underline{l}, \underline{m}, \underline{n}, \underline{p}$ ismert $\rightarrow q_1 .. q_6 = ?$

($\underline{p}$ és d_4 ismert)

Innen $q_1 .. q_3$ ismeretlenek meghatározhatóak

$$[\underline{l}\ \underline{m}\ \underline{n}] = \underline{\underline{A}}_{0,3}(q_1, q_2, q_3) \cdot \underline{\underline{A}}_{3,6}(q_4, q_5, q_6)$$

$$\underline{\underline{A}}_{3,6} = \underline{\underline{A}}_{0,3}^T \cdot [\underline{l}\ \underline{m}\ \underline{n}] \rightarrow q_4 .. q_6 \text{ meghatározható}$$

1.6. Robot differenciális mozgása

Inverz feladat: nemlineáris, nem zárt alakban felírható

→ numerikus eljárás
→ valós idejű alkalmazásban nem jó.

Helyette deriváltak felírása: lokális, konfigurációfüggő

q_i: csuklókoordináták
K_m: megfogó felerősítés helye...
v_m: sebesség
ω_m: szögsebesség
K_{i-1}: álló koordinátarendszer
[i]..[m]: szegmensek: merev test
$^j x$: x változó a j-edik ~~szegmensben~~ keretben
[i] előtti q_i csukló iránya: s_{i-1} (DH-alaknál $s_{i-1} = z_{i-1}$ egybeesik)

$^m\underline{v}_{i,m}$ és $^m\underline{\omega}_{i,m}$ parciális {/szög}sebesség K_m keretorigójában, amit q_i változása eredményez

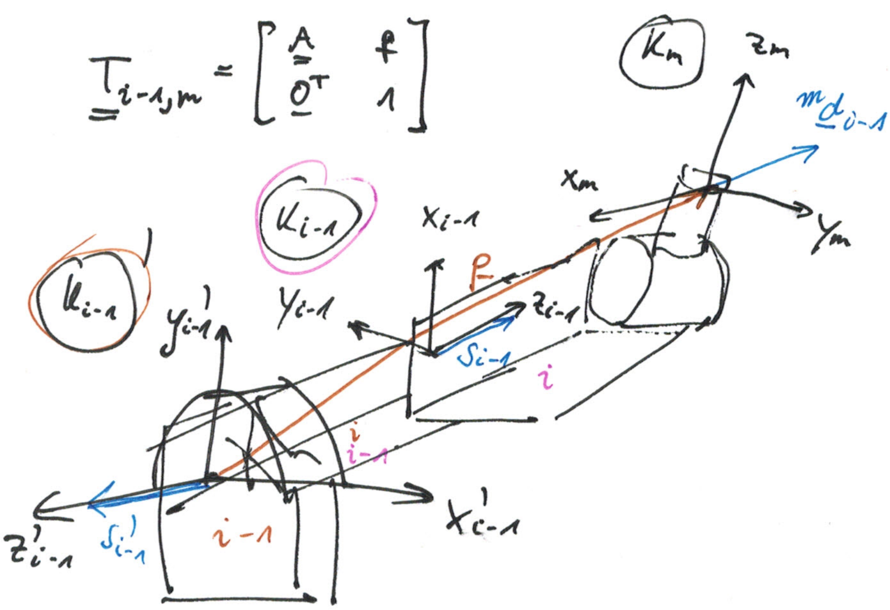
Ezek irányai

$^m\underline{d}_{i-1}$ és $^m\underline{t}_{i-1}$

$$^m\underline{v}_{i,m} = {^m\underline{d}_{i-1}} \cdot \dot{q}_i$$

$$^m\underline{\omega}_{i,m} = {^m\underline{t}_{i-1}} \cdot \dot{q}_i$$

$$\underline{\underline{T}}_{i-1,m} = \begin{bmatrix} \underline{\underline{A}} & \underline{p} \\ \underline{0}^T & 1 \end{bmatrix}$$



$$^{m}\underline{d}_{i-1} = \underline{\underline{A}}^{-1} \cdot {}^{i-1}\underline{s}_{i-1}$$ ← (K_m)-ben kifejezve

Transzlációs csukló

Rotációs csukló

$$^{m}\underline{t}_{i-1} = \underline{\underline{A}}^{-1} \cdot {}^{i-1}\underline{s}_{i-1}$$

$$^{m}\underline{d}_{i-1} = \underline{\underline{A}}^{-1} \cdot \left({}^{i-1}\underline{s}_{i-1} \times \underline{p}\right)$$

1.6.2. A robot Jacobi-mátrixa

$$^{m}\underline{v}_{m} = \sum_{i=1}^{m} {}^{m}\underline{d}_{i-1} \cdot \dot{q}_i$$

$$^{m}\underline{\omega}_{m} = \sum_{i=1}^{m} {}^{m}\underline{t}_{i-1} \cdot \dot{q}_i$$

$$\begin{bmatrix} ^{m}\underline{v}_m \\ ^{m}\underline{\omega}_m \end{bmatrix} = \underbrace{\begin{bmatrix} ^{m}\underline{d}_0 & \dots & ^{m}\underline{d}_{m-1} \\ ^{m}\underline{t}_0 & \dots & ^{m}\underline{t}_{m-1} \end{bmatrix}}_{^{m}\underline{\underline{J}}_m} \cdot \dot{\underline{q}}$$

$^{m}\underline{\underline{J}}_m$: robot [m]. szegmenséhez tartozó K_m keretben kifejezett Jacobi-mátrixa.

Ez persze függ az aktuális csuklókoordinátáktól.

1.6.3 Direkt kinematikai feladat

$$\begin{bmatrix} ^{m}\underline{v}_m \\ ^{m}\underline{\omega}_m \end{bmatrix} = {}^{m}\underline{\underline{J}}_m \cdot \dot{\underline{q}}$$ [illegible] @ K_m

@ K_0

$$^{0}\underline{v}_m = \underline{\underline{A}}_{0,m} \cdot {}^{m}\underline{v}_m$$

$$^{0}\underline{\omega}_m = \underline{\underline{A}}_{0,m} \cdot {}^{m}\underline{\omega}_m$$

$$^{0}\underline{\underline{J}}_m = \begin{bmatrix} \underline{\underline{A}}_{0,m} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{A}}_{0,m} \end{bmatrix} \cdot {}^{m}\underline{\underline{J}}_m$$

$$\begin{bmatrix} ^{0}\underline{v}_m \\ ^{0}\underline{\omega}_m \end{bmatrix} = {}^{0}\underline{\underline{J}}_m \cdot \dot{\underline{q}}$$

1.6.4. Inverz kinematikai feladat

$$\underline{\underline{T}}_{0,m}(q) = \begin{bmatrix} \underline{l} & \underline{m} & \underline{n} & \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ így lokálisan linearizálható

Így $\dot{\underline{q}}$ megoldható és ennek integrálásával q is.

$$\dot{\underline{q}} = {}^0\underline{\underline{J}}_m^{-1} \cdot \begin{bmatrix} {}^0\underline{v}_m \\ {}^0\underline{\omega}_m \end{bmatrix}$$

6-DOF esetén $\underline{\underline{J}}_m$ kvadratikus, invertálható, ha nem szinguláris.
De vannak olyan q konfigok, ahol $\underline{\underline{J}}_m$ szinguláris → adott irányban akkor nem tud elmozdulni.

1.6.5 Statikus erők és nyomatékok transzformálása

Kérdés: külső erő, nyomaték ⟶ milyen csuklónyomaték ?

$\begin{bmatrix} \underline{f}_m \\ \underline{\tau}_m \end{bmatrix}$ → [m]-re ható erő
→ [m]-re ható nyomaték

Minden K_0-ban kifejezve $(x = {}^0x)$

$\underline{\tau} = (\tau_1, .., \tau_m)^T$
általánosított csuklónyomatékok (tr → erők ; rot → nyomatékok)

K_m keretben ható f_m, t_m
Virtuális munka elve

$$\partial W = \underline{\tau}^T \cdot \partial \underline{q}$$

$$\partial W = \begin{bmatrix} \underline{f}_m \\ \underline{t}_m \end{bmatrix}^T \cdot \begin{bmatrix} \underline{v}_m \\ \underline{\omega}_m \end{bmatrix} \cdot \partial t$$

$$\underline{\tau}^T \cdot \partial q = \begin{bmatrix} \underline{f}_m \\ \underline{t}_m \end{bmatrix}^T \cdot \begin{bmatrix} \underline{v}_m \\ \underline{\omega}_m \end{bmatrix} \partial t \qquad / \begin{bmatrix} \underline{v}_m \\ \underline{\omega}_m \end{bmatrix} = \underline{\underline{J}}_m \cdot \frac{\partial q}{\partial t}$$

$$\underline{\tau}^T \cancel{\partial q} = \begin{bmatrix} \underline{f}_m \\ \underline{t}_m \end{bmatrix}^T \cdot \underline{\underline{J}}_m \cancel{\partial q}$$

$$\underline{\tau}^T = \begin{bmatrix} \underline{f}_m \\ \underline{\tau}_m \end{bmatrix}^T \cdot \underline{\underline{J}}_m$$

$$\underline{\tau} = \underline{\underline{J}}_m^T \cdot \begin{bmatrix} \underline{f}_m \\ \underline{t}_m \end{bmatrix}$$

1.7. A robot dinamikája

Dinamika: nyomatékok és gyorsulások kapcsolata

$$\xrightarrow{\text{direkt}}$$

$$\xleftarrow[\text{inverz}]{\cancel{\text{indirekt}}}$$

$F = m \cdot a$

$M = \Theta \cdot \beta$

Mozgásegyenlet

$$\underbrace{\underline{\underline{H}}(q)\cdot\ddot{q}}_{\text{inerciális rész}} + \underbrace{\underline{h}(q,\dot{q})}_{\text{zavaró hatások}} = \underline{\tau}$$

@ csuklókoordináták terében.

$\underline{\underline{H}}(q)$: pozitív definit. „inerciamátrix" ~ robot tehetetlensége

$\underline{h}(q,\dot{q})$: akkor kell, hogy a gyorsulásától ne változzanak meg

pl. gravitáció, centrif., coriolis-erő.

$$\tau_i = \sum_{j=1}^{m} D_{ij}\ddot{q}_j + \sum_{j=1}^{m}\sum_{k=1}^{m} D_{ijk}\dot{q}_j\dot{q}_k + D_i$$

D_{ii} : eff. inercia

D_{ij} : csatoló inercia $(i \neq j)$ $(D_{ij} = D_{ji})$

D_{ijj} : centrifugális hatás

D_{ijk} : coriolis $(j \neq k)$

D_i : grav.

1.8 Robotirányítási módszerek

Szabad mozgás: kis hiba OK

Kontaktus: kis hiba NEM OK

↳ hibrid pozíció és erőirányítás

- Pont-pont ir.
- Folytonos pálya ir.

lineáris interpoláció { Descartes-koordinátákban → egyenes vagy; csuklókoordinátákban → görbe

sokkal kevesebb számítás

Még az időparaméter is fontos

$\underline{\underline{T}}_{BE}(t)$ pozícióalgoritmus: $\underline{q}$ inv. geom. feladatból

${}^B\underline{v}_E(t)$ és ${}^B\underline{\omega}_E(t)$ sebességalgoritmus: $\dot{\underline{q}} = \underline{\underline{J}}_E^{-1} \begin{bmatrix} \underline{v}_E \\ \underline{\omega}_E \end{bmatrix}$ $\dot{q}$ inv. kin. F.-ből.

végberendezés Jacobi-mx-a.

Kell hozzá $\forall t$: csuklókoordináták

Számításokat mindig el kell végezni

↳ Innen q vagy $\dot{q}$ lehet alapjel a szabályozási algoritmusnak

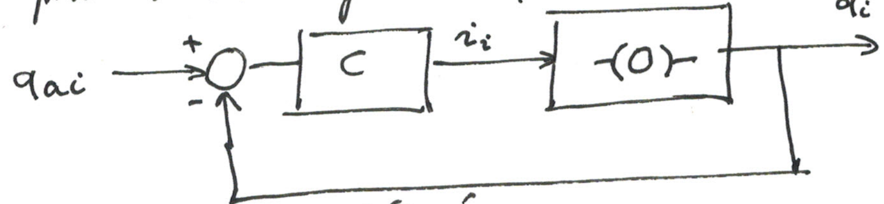
1.8.1. Decentralizált szervóhajtások

A legtöbb mai ilyen

Robot → nem lineáris

Szervók → lineárisak kb.

Központi proci → inverz feladat (q_i) számítása



Szervónál egyszerű szabályozó

Többi tag csatoló hatása: zajként kezeljük

↳ Nagy szervó jelek nagy sebességnél/gyorsulásnál lokálisan nehezen eliminálhatóak.

1.8.2 Kiszámított nyomatékos (nemlineáris visszacsatolás) módszer

meghajtó nyomatékok a motorokon: $\tau = \underline{\underline{H}}(\underline{q}) \cdot \underline{\ddot{q}} + \underline{h}(\underline{q}, \underline{\dot{q}})$

(Tegyük fel: a τ nyomaték hibátlanul követi a τ_a alapjelet)

$\underline{\tau_a} = \underline{\underline{H}}(\underline{q}) \cdot \underline{u} + \underline{h}(\underline{q}, \underline{\dot{q}})$

Vagyis: $u = \ddot{q}$ (ehhez $H(q)\ddot{q} + h(q,\dot{q}) = H(q)u + h(q,\dot{q})$

kell h. $H(q)$ poz. szemidef $\Rightarrow$ inv.ható)

u: beavatkozó jel

robot dinamikus modelljének ismerete kell

Egymástól független kettős ~~integrátok~~ integrátorok

$$u_i = \ddot{q}_{ai} + k_{Pi}(q_{ai} - q_i) + k_{Ii}\int_0^t (q_{ai} - q_i)\,dt + k_{Di}(\dot{q}_{ai} - \dot{q}_i)$$

↑ Ezzel a PID-alg.-al szabályozunk

Alapjelként igényli: $\{q_{ai}, \dot{q}_{ai}, \ddot{q}_{ai}\}$

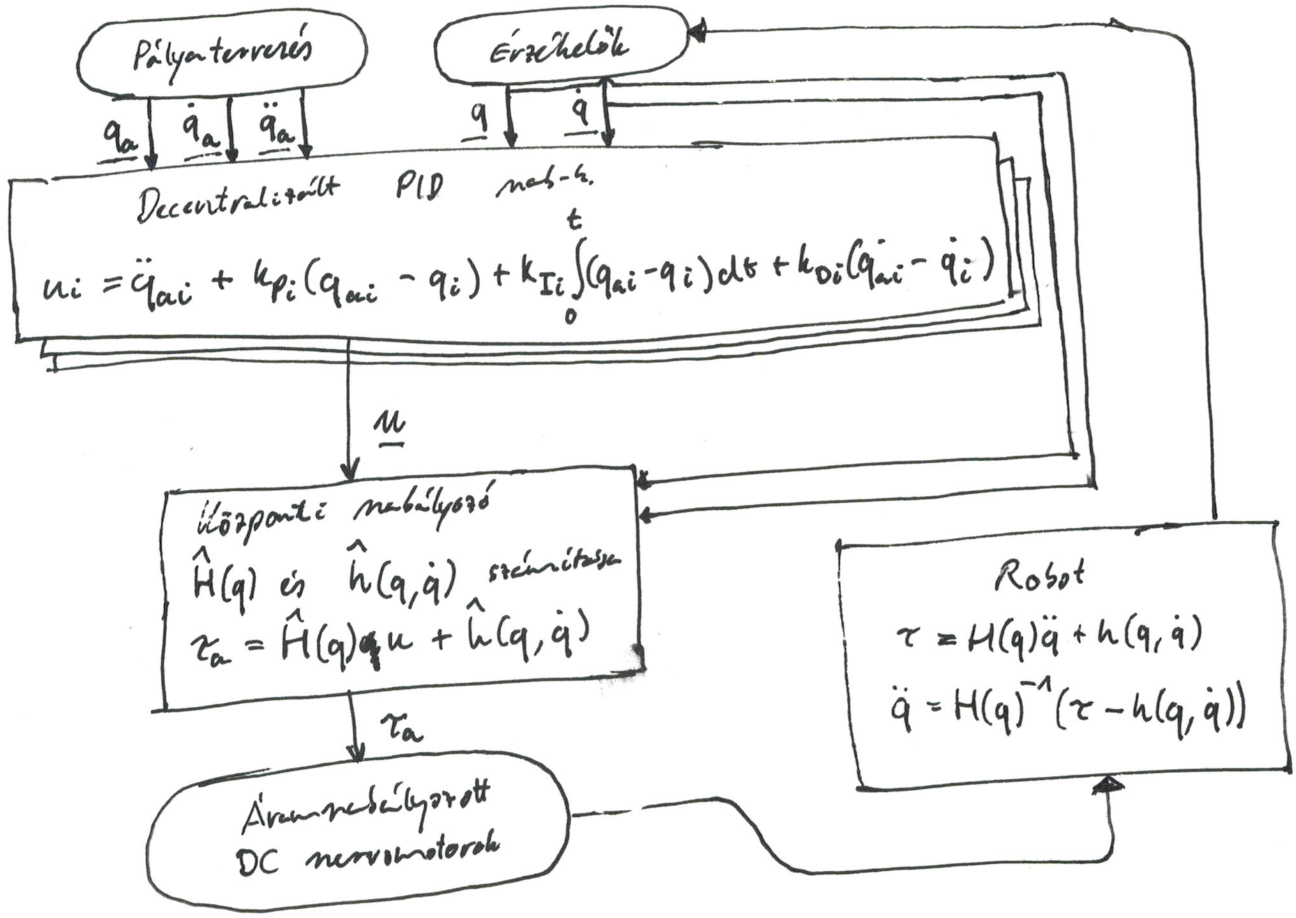
$$\ddot{q}_i = u_i$$

$$(\ddot{q}_{ai} - \ddot{q}_i) + k_{Di} \cdot (\dot{q}_{ai} - \dot{q}_i) + k_{Pi}(q_{ai} - q_i) + k_{Ii}\int_0^t (q_{ai} - q_i)\,dt = 0$$

szabályozási hiba: $q_{di} \triangleq q_{ai} - q_i$ $\quad \downarrow \frac{d}{dt}$

$$\dddot{q}_{di} + k_{Di}\ddot{q}_{di} + k_{Pi}\dot{q}_{di} + k_{Ii} q_{di} = 0$$

Megfelelő k paraméterekkel ennek a megoldása exponenciálisan lecsengő.



Szabályozó (kis mért.ben pontatlan) számított modellje

$$\hat H(q), \quad \hat h(q,\dot q)$$

Ekkor a rendszer működése:

$$H(q)\ddot q + h(q,\dot q) = \hat H(q)u + \hat h(q,\dot q)$$

$$\Rightarrow \ddot q = H(q)^{-1}\left\{\hat H(q)u + \hat h(q,\dot q) - h(q,\dot q)\right\} \neq u$$

!

[Csuklónként kis telj. CPU → PID-hez
Közp. proc → dinam. modell
Másik proc (!) → alapjel világkoordinátákból csuklókoordinátákba]

Procik
$2 + n(DOF)$

~~Alkbrom~~

1.8.3 Hibrid pozíció és erőirányítás

Érintkezés → kinematikai korlátozások

Pontos pozícióirányítás nem elég

Kell: erők mérése

DC motorok árammérése
+ egyszerű
+ olcsó
- csak becslésre

Robotcsukló utolsó szegmense és megfogó közé 6-komponensű erő- és nyomatékérzékelő

Korlátozott mozgásirányban erőt (nyomatékot) szabályoz

Korlátozás nélküli irányban: pozíció

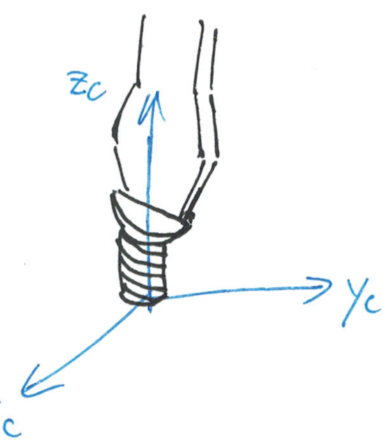
K_C compliance frame → engedékenységi keret

Természetes korlátozások esnek a tengelyirányba!
↳ ezen irányokban erőirányítás
↳ többi irányban pozícióirányítás

Nem korlátozott irányban mesterséges korlátozások

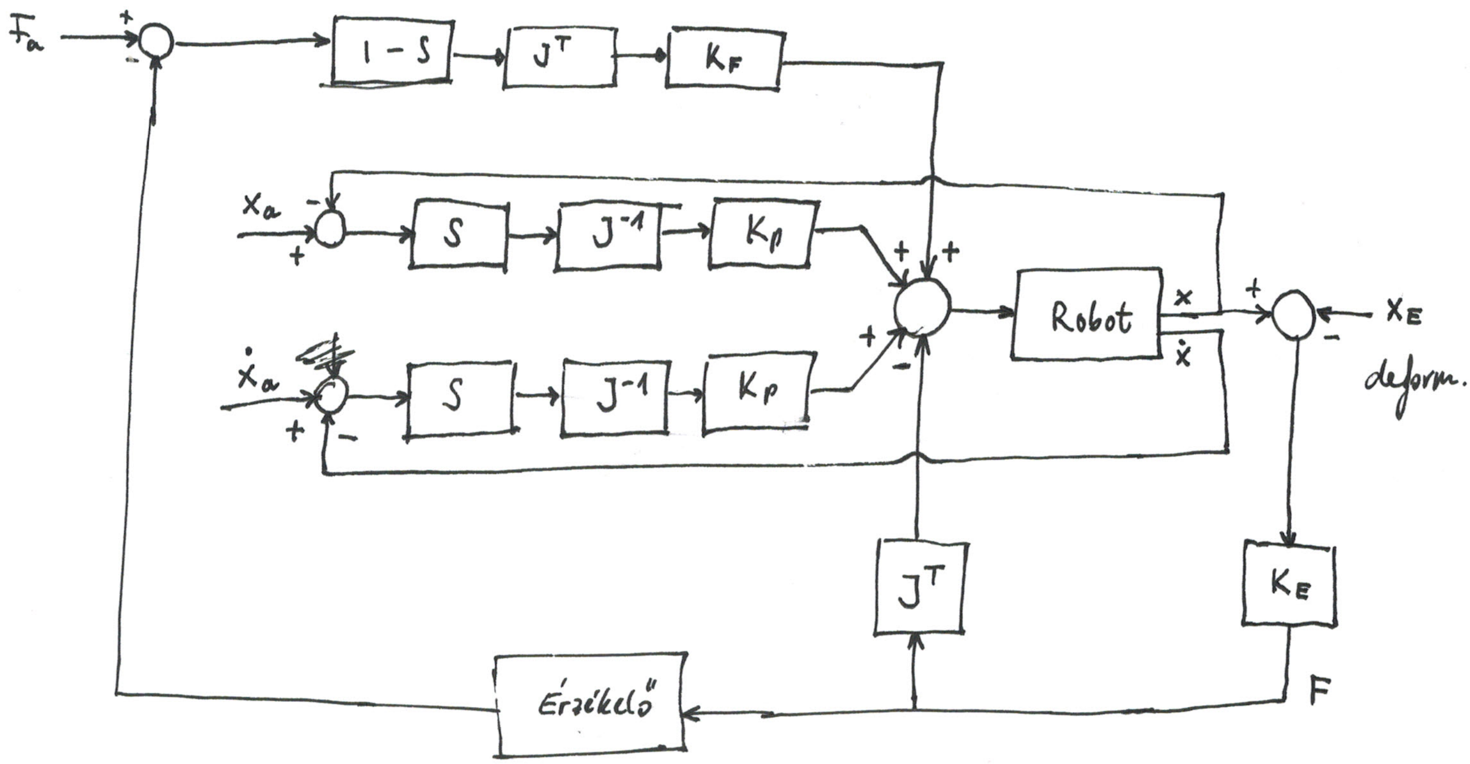
Pozíció/erő előírása

Ezeket K_C-ben értelmezzük



Term. korlátozások		Mest. k.
$v_x = 0$	→	$f_x = 0$
$\omega_x = 0$	→	$\tau_x = 0$
$\omega_y = 0$	→	$\tau_y = 0$
$f_y = 0$	$(\neg v_y)$ →	$v_y = 0$
$f_z = 0$	$(\neg v_z)$ →	$v_z = -p \cdot \omega_a$
$\tau_z = 0$	$(\neg \omega_z)$ →	$\omega_z = -\omega_a$

$S \Leftarrow 1$



X_a : pozícióalapjel

$\dot{x}_a$: sebességalapjel

F_a : erőalapjel

x_E : kontaktuspontban rugalmas alakváltozás

$F = K_E \cdot (x - x_E)$ kontaktus erőt eredményez

Jacobi m-x-en keresztül → tengelynyomatékok

S: szelekciós mx 1→poz. ir. 0→erőir.

ott: $S = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & 1 \end{bmatrix}$

$$Sv = S \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} v_y \\ v_x \\ \vdots \\ \omega_z \end{bmatrix} \qquad (I - S)f = (I - S) \begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} f_x \\ \cdot \\ \cdot \\ \tau_x \\ \tau_y \\ \cdot \end{bmatrix}$$

2. A NOKIA-PUMA 560-AS ROBOT IRÁNYÍTÓRENDSZERE

2.1. Inkrementális adók

Impulzusjeladó

Analóg jeladó

+ pontosság
+ teljes reprod.
+ megbízható
+ zavarérzéketlen

- abs. poz. info hiánya
- kihelyezett elektr. tápigénye
- jelvezetékszám

Nullimpulzuslogika

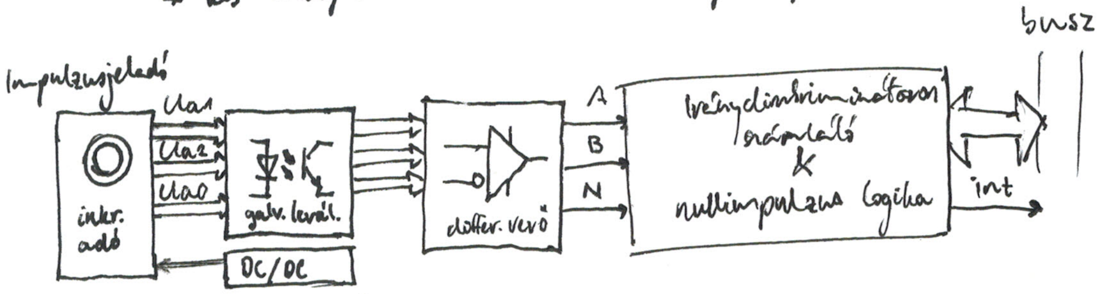
↳ Mechanikai véghelyzetkapcsolóval együtt megmondja annak segítségével az elért abszolút pontosságot → végig kell odáig & nullimpulzusig menni

↳ Potenciométer (NP560 is) abs. pozíció inf.

Megmondja legalább melyik körülfordulásban vagyunk

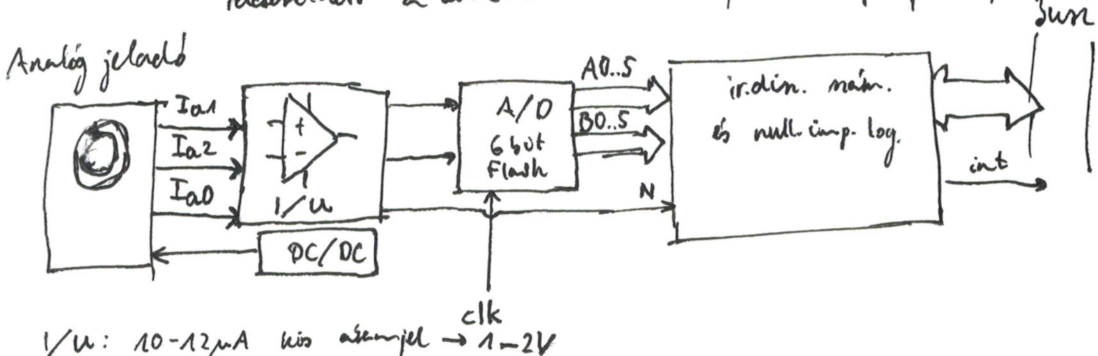
Elég az első nullimpulzusig fordulni

+ kis mozgás — drága poti



galv. levál.: - sebesség
- Fázishelyzetet tartsa!

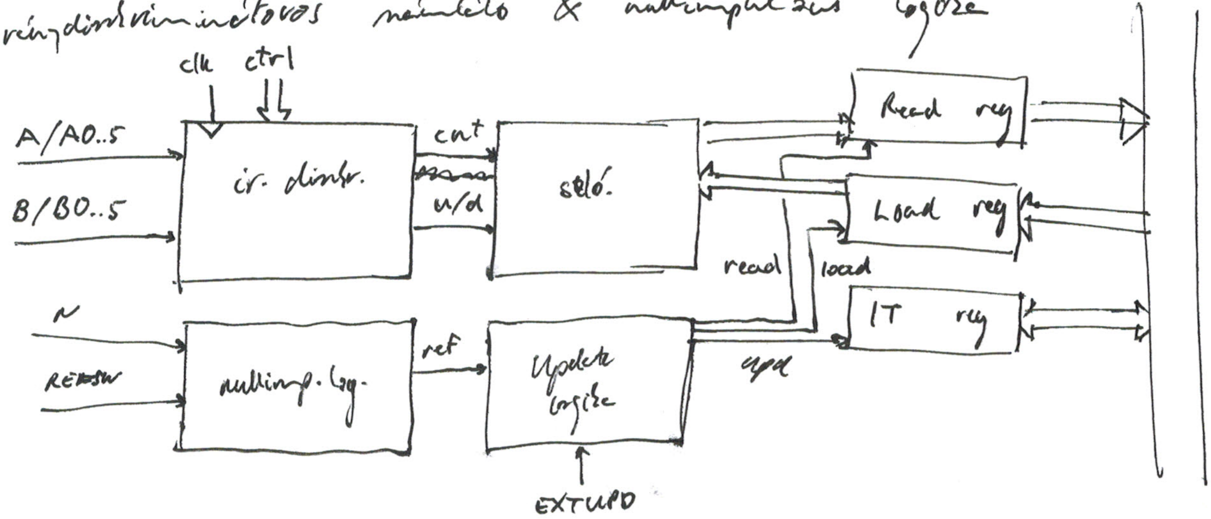
Diff. vevő: pl. RS-485/422
felcserélhető a leválasztással: fázishelyzet szempontjából jobb, de drágább.



I/U: 10-12 μA kis áramjel → 1–2V
nullimp → logikai jel
kis áram → nem lehet leválasztani, de legalább áramjel

Flash AD:

Iránydiszkriminátoros számláló & nullimpulzus logika



iránydiszkriminátor: szűrés lehet, ha tud túlmintavételezni

1/2/4-szeres kiértékelésnél megfelelően cnt + u/d jelek

számláló: ált 16 v 32 bit

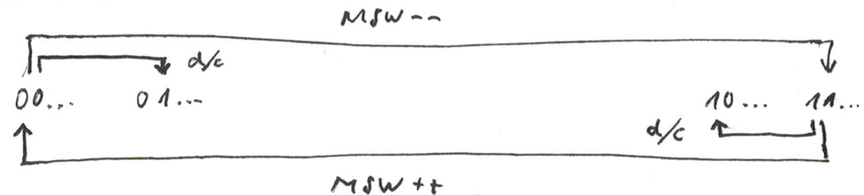
↳ SW kiterjesztés

nullimpulzus logika → ref impulzus

kellhet kalibráláshoz (refsw aktív) vagy

minden nullimpulzuson áthaladáskor hihetőségvizsgálat (előző érték + $\frac{inkr}{ford} \cdot \{-1, 0, 1\} \approx$ új érték)

SW kiterjesztés



Két kiolvasás közt $T_{max} < \frac{2^{14}}{f_{max}}$ idő telhet el

Sebesség:

$$v \cong \frac{\Delta s}{\Delta t} = \frac{\Delta N q}{T_v}$$

q: felbontás $[\frac{m}{inkr}]$ ΔN: inkr T_v: mintavételi idő

Pontosság: $\frac{1}{\Delta N}$ így T_v nem lehet túl kicsi!

Kis sebességnél:

$$v \cong \frac{\Delta s}{\Delta t} = \frac{kq \cdot N}{f_s}$$

q: $[\frac{m}{inkr}]$ f_s: $[\frac{1}{s}]$ segédfr. N: segédfr. számláló állása $= N \cdot T_s = \frac{N}{f_s} = \Delta t$

k: [inkr] amennyi inkrementet számláltunk két időmérés közt (const)

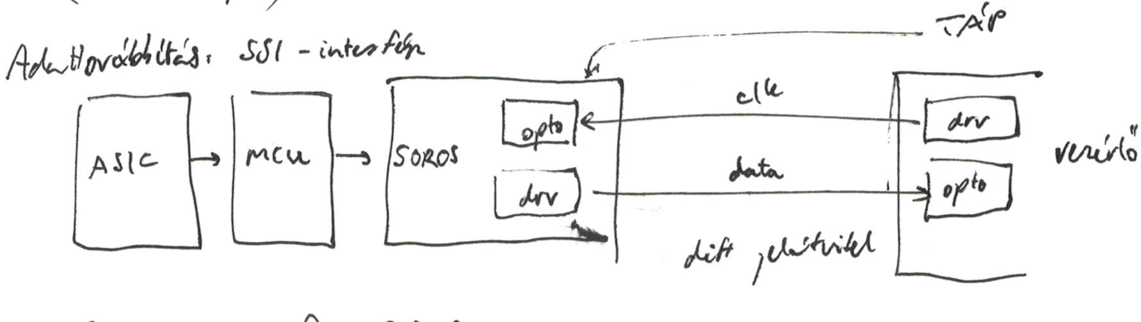
2.2. Abszolút szöghelyzet adók

Ált 12–18 bit

Gray-kód $\Delta bit\# = 1$

Multiturn: elektronikus kiegészítés több (pl. 4096) fordulaton keresztül abszolút kódolásra (+12 bit pl.)

Adattovábbítás: SSI-interfész

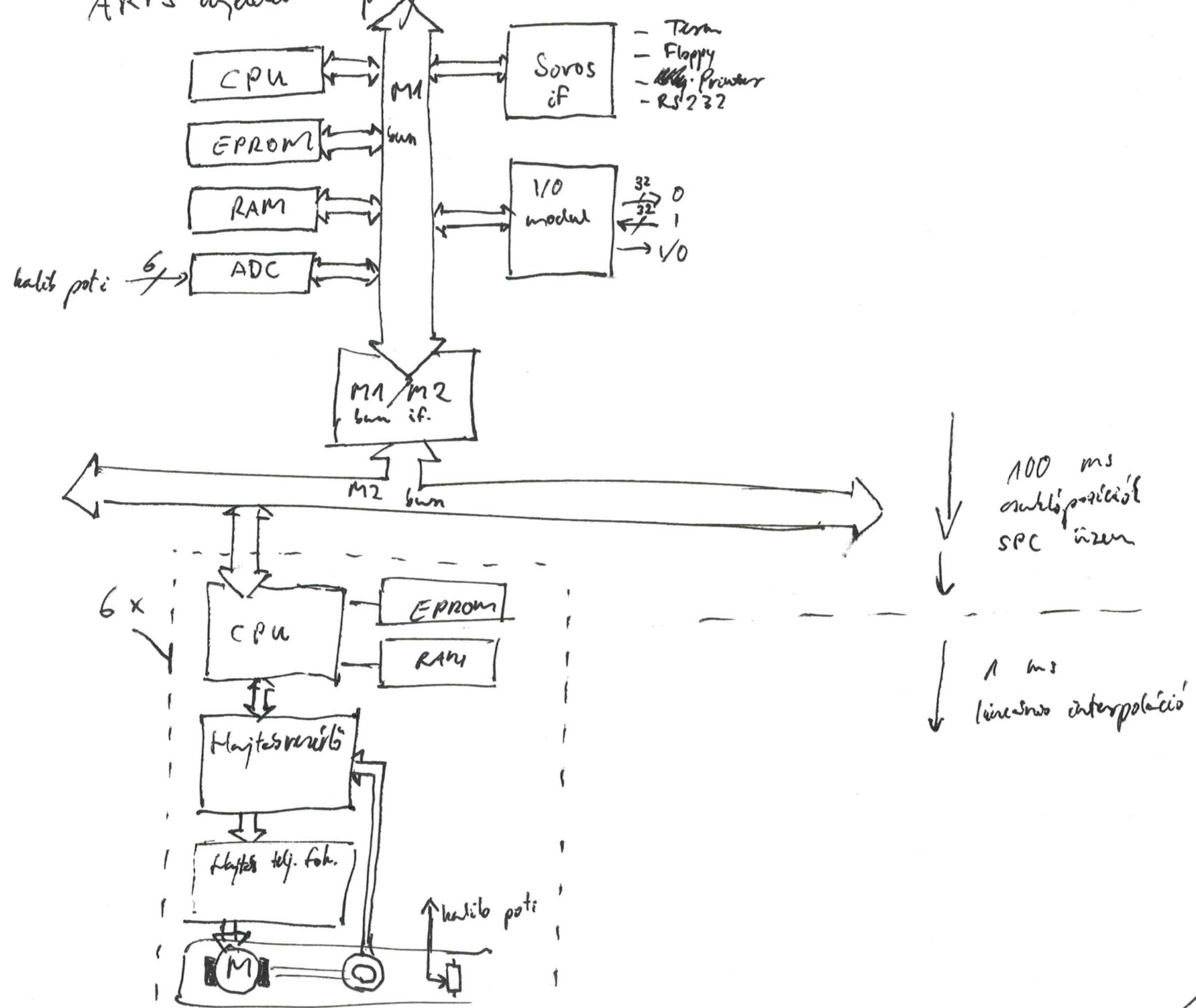


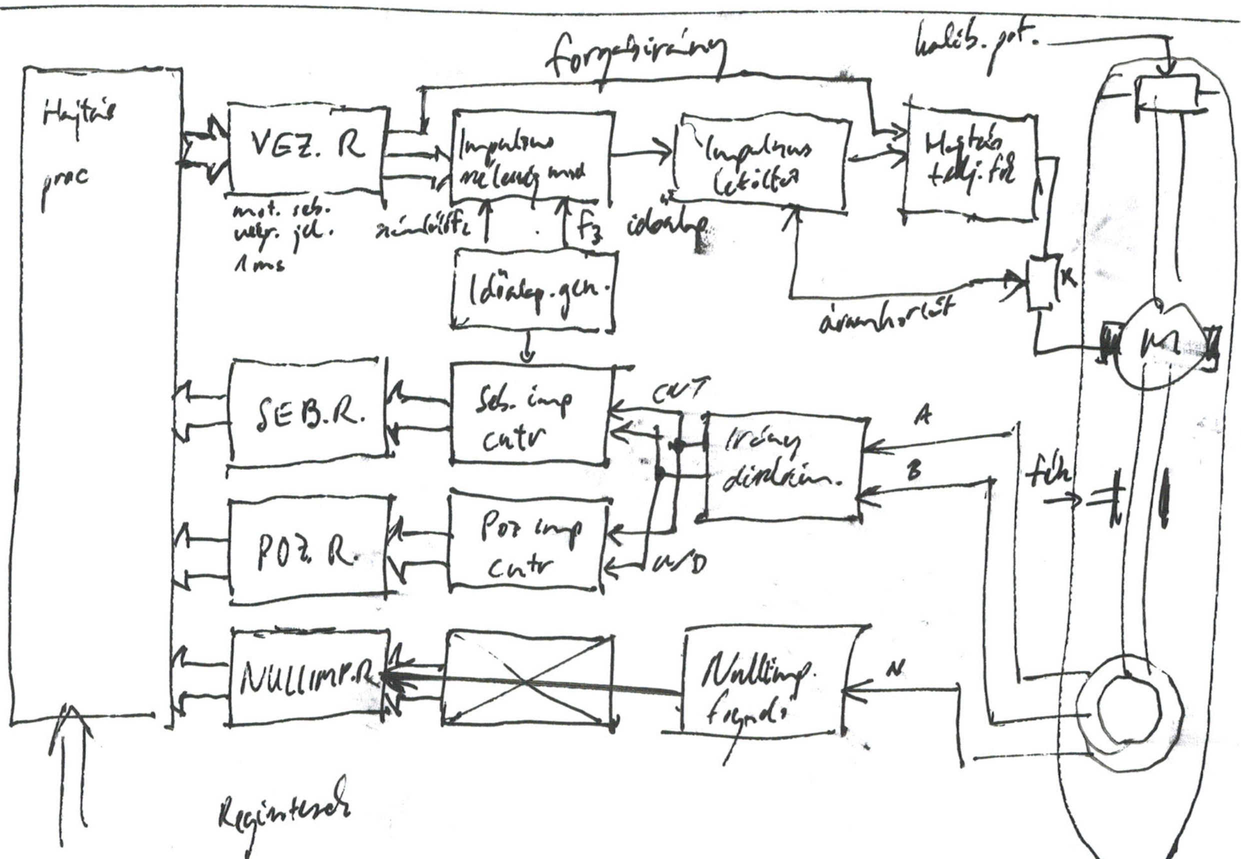
Pl. EnDat if. RS485

2.3 X

2.4 Szfera-35 irányítórendszer

ARPS nyelven programozható





Pályapontok (abszolút)
Felügyelőtől
100 ms

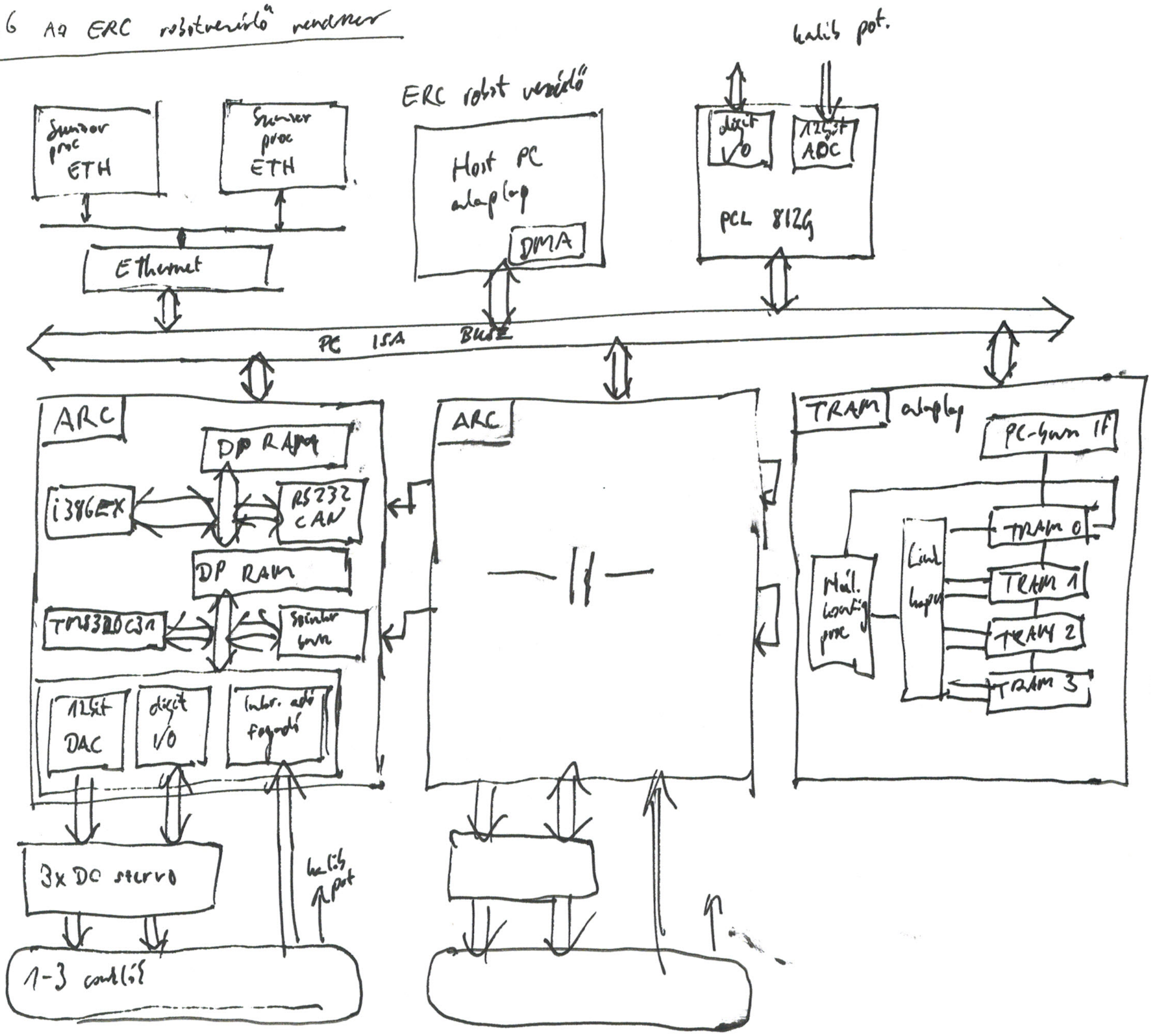
2.5 Fejlődés iránya

összefüggések miatt csuklók nem kezelhetők külön

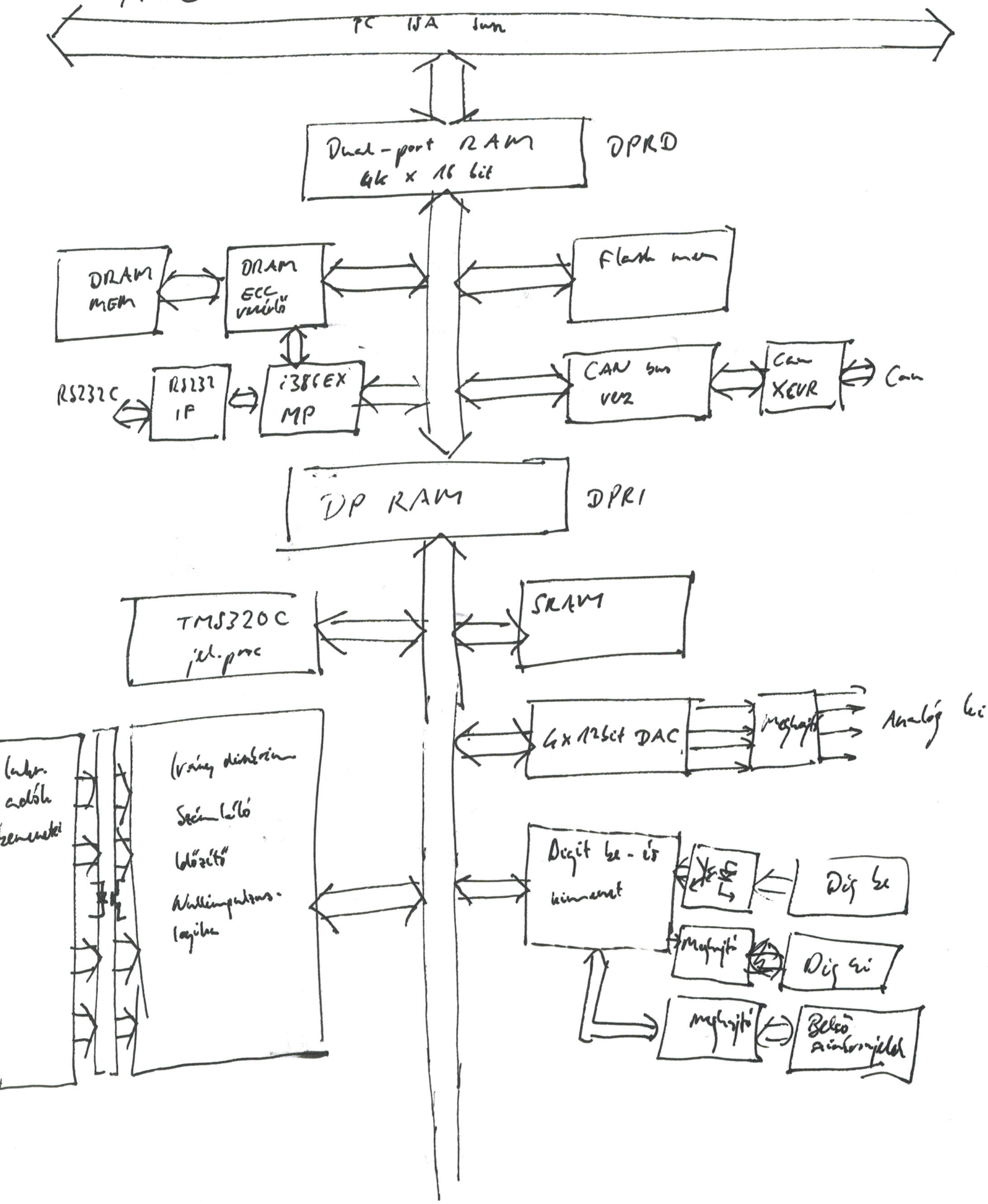
3 szintű architektúra

1. – pályatervezés
2. – dinamikus modell
3. – megfogó nyomatékok számítása

2.6 Az ERC robotvezérlő rendszer



ARC


PC ISA bus
Dual-port RAM
4k x 16 bit
DPRD
DRAM MEM
DRAM ECC vezérlő
Flash mem
RS232C
RS232 IF
i386EX MP
CAN bus vez
Can XCVR
Can
DP RAM
DPRI
TMS320C jel.proc
SRAM
4x12bit DAC
Meghajtó
Analóg ki
Irány diszkrim
Számláló
Időzítő
Nullimpulzus-logika
Digit be- és kimenet
Meghajtó
Dig ki
Meghajtó
Belső státuszjelek
Dig be

3. SZABÁLYOZÓK ÉS PROGRAMOZÁSUK

3.1. Alapfogalmak

Szabályozás: negatív vcs kimenő jelről

LTI rsz → Feladatai:
- Értéktartás: zavaró jellemzők ellenére
- Követés: változó alapjel. robotoknál tipikus

NLTI rsz → munkaponti lin.

3.1.1. A szabályozók típusai

Állásos szabályozó:

- Kimenő jel diszkrét értékű.
- Gyakran 2/3/5 állású
  - motoroknál 0 + névleges · 2 + lassú · 2
- \+ kapcsolóüzemű kimenő teljesítményfokozat
- − nem szabályoz 0 maradó hibára (állandósult lengés)

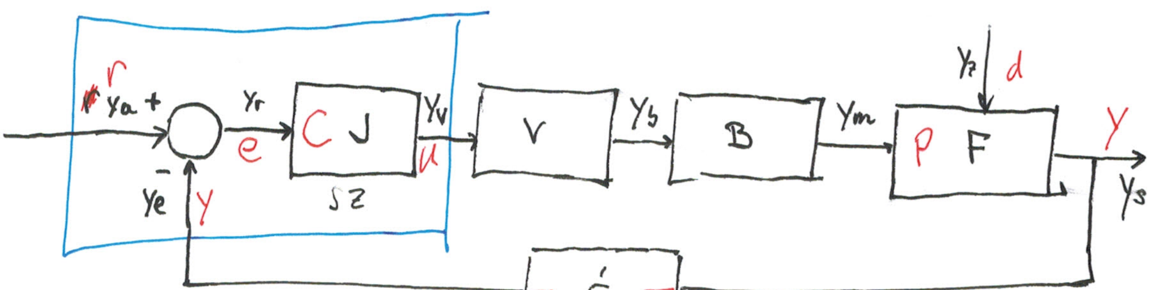
Időarányos szabályozó

- Ki: PWM
- \+ Kapcsolóüzemű

Folytonos szabályozók

- DT vagy CT (analóg vs digitális)
- ~~Utóbbi~~ Előbbi is lehet CT-ként kezelhető, ha időállandója $< \frac{1}{10} \cdot$ rsz időállandója.

3.1.2 Szabályozók funkcionális egységei és jelei



P F: folyamat
y ys: szabályozott jellemző
ym: módosított jellemző
d yz: zavaró jellemző

É: érzékelő
ye: ellenőrző jel
C Sz: szabályozó jel
r ya: alapjel

e yr: hibajel
J: jelformáló
u yv: végrehajtó jel

3.1.3. A szabályozásokkal szemben támasztott elvárások

- stabilitás
- zérus maradó hiba

Függések
alapjelkör., zavarelh. ← típusszám
maradó hiba ← hurokerősítés

- szabályozási idő
  ↳ ω_c vágási frek → $\frac{3}{\omega_c} < t_c < \frac{10}{\omega_c}$
  Ahol a hurokerősítés $= 1$

- lengési hajlam
  ↳ φ_m fázistartalék: nyitott kör fázistolása ω_c-n $-180°$-tól felfelé
  $\varphi_m = 60° \rightarrow$ lengés $< 10\%$

3.2 Szabályozók kiválasztása

PID egyéltergedtesl

$$\frac{u(s)}{e(s)} = k_c \cdot \left(1 + \frac{1}{sT_I} + \frac{sT_D}{1 + sT}\right)$$: párh. ipari PID

Tervezéshez soros PIPD

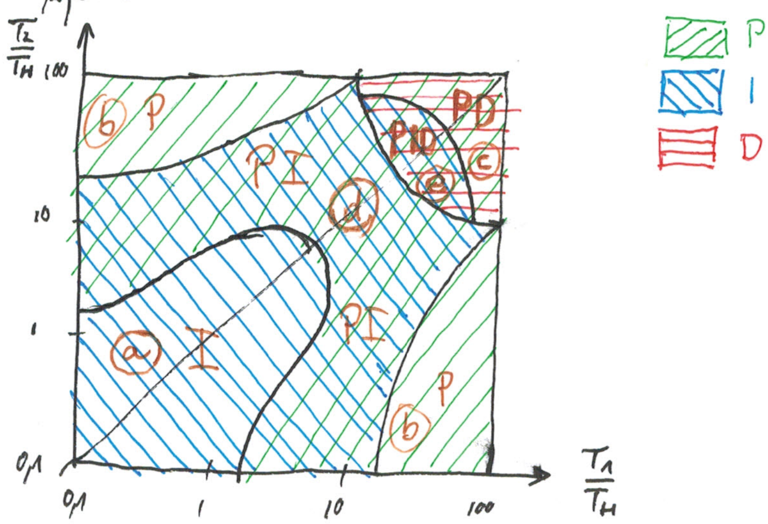
$$\frac{u(s)}{e(s)} = k_c \cdot \frac{1 + sT_I}{sT_I} \cdot \frac{1 + sT_D}{1 + sT}$$ (nem azonos T értékek!)

3.2.1. Szabályozók arányos folyamathoz.

Arányos folyamat ≈ 2-tárolós holtidős tag

$$P(s) = \frac{e^{-sT_H}}{(1 + sT_1)(1 + sT_2)}$$

Ajánlás:



a) tároló nélküli holtidős tag
Nyquist-diag: kör → 1-nél nagyobb k_c-re labilis
I-vel stabilizálható

b) egytárolós folyamat
P-vel stabilis, nagy k_c eltünteti a hibát

c) kéttárolós azonos időállandóval
Bode → nincs $-20\,\frac{dB}{D}$ szakasz
PD-vel áthelyezzük egy pólusát, hogy legyen
Itt lesz az ω_c, de tervezzük!
Strukturálisan stabilis kör, de PD → nagy túlvezérlés, kis statikus hiba

d) legelterjedtebb ált. eset
PI-vel egyező alakú jelre hiba nélkül szabályoz
I-nél gyorsabb, nincs nagy túlvezérlés

e) ritka
PID akkor indokolt, ha két közel azonos időállandó és nem elhanyagolható holtidő
k_c nem lehet nagy (TH), ezt javítja az PI (stat. hiba)
Időállandókat PD megosztja, de nagy lesz a túlvezérlés

3.2.2. Szabályozók integráló folyamathoz

- P vagy PD: zérus hiba
  PD: legnagyobb időállandó túlvezérlés árán gyorsítható
- PI kell, ha a zavarás az integráló hatás előtt hat.
  −40 és −60 dB/d-os szakasz van
  $-20\,\frac{dB}{D}$-os szakasz: integrátor időállandóját a legnagyobb időállandónál min 1D.-al balra (tizenegyszere) (Felt: 2. időáll. nincs túl közel.)
- PI²D:
  legnagyobb időáll → nagyobb frek.
  integ. idő → 1 D.-al nagyobbra
  PI-hez képest kisebb integrálási idő.
  nagy túlvezérlés árán gyorsít.

3.3 Szervomotorok

3.3.1. A legfontosabb szervomotorok

- Állandó mágneses egyenáramú
  - kb. lineáris
  - egyszerű vezérlés
  - ált kommutátoros, tönkre tud menni
- Állandó mágneses szinkron
  - kefenélküli igénytelen
  - drága mágnes
  - bonyolult vezérlés: inverter elven PWM frekvenciaváltoztatás
- 3-fázisú aszinkron
  - előző előnyei
  - kalickás: olcsó, kis tehetetlenségi nyomaték
  - azonos vezérlési elv
- Léptető motor
  - szinkron elvű vagy lágyvas forgórészű
  - kis telj.
  - ok. pozícionálható, de tévedhet és nyomatékban lüktet
- Switched reluctance motor
  - = lágyvas forgórészű léptető + vezérlő elektronika
  - Álló és forgórész más mértékben fogazott (kiálló pólusú)
  - nagy fogszámmal áttétel nélküli hajtás
  - illesztett táplálás az állandó nyomaték érdekében
  - szögelfordulás – áram függvényt EPROM tárolja
  - forgórész pozícióját Hall-szenzor vagy optoszenzor érzékeli
- kefenélküli egyenáramú brushless DC motor
  - Állandó mágneses forgórészű szinkron motor
  - Forgórész pozíciójától függő táplálás
  - Kb. lineáris
  - Vezérlő elektronikával együtt
  - Drágább

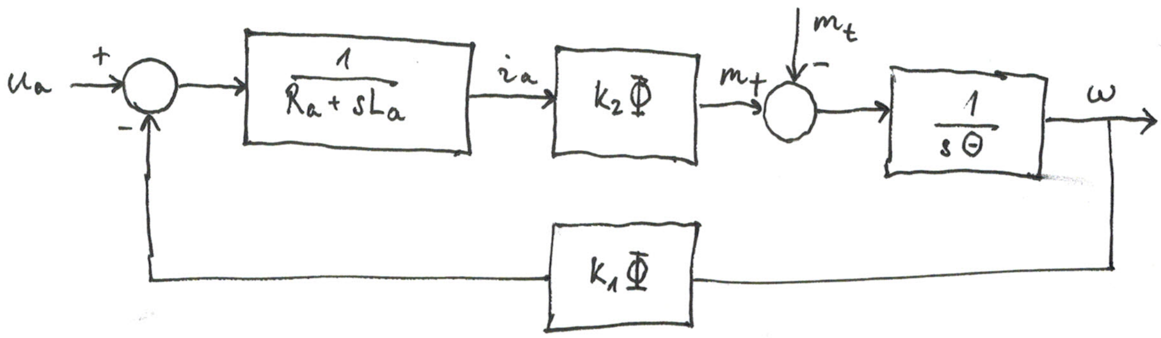
3.3.2. Az egyenáramú szervomotor

Armatúrakör: $u_a = i_a \cdot R_a + L_a \cdot \frac{di_a}{dt}$

Indukált fesz: $u_i = \underbrace{k_1}_{\text{gépállandó}} \cdot \underbrace{\Phi}_{\text{Fluxus}} \cdot \underbrace{\omega}_{\text{forgórész mozgássebessége}}$

Motor EM nyomatéka: $m = k_2 \cdot \Phi \cdot i_a$

Gyorsító nyomaték: $\underbrace{m - m_t}_{m_t\text{: terhelő nyom.}} = \underbrace{\Theta}_{\text{forgórész + terhelés tehetetlenségi nyomatéka}} \cdot \frac{d\omega}{dt}$



$$\frac{\omega(s)}{u_a(s)} = \frac{A_M}{1 + sT_M + s^2 T_V T_M}$$

$A_M = \frac{1}{k_1 \Phi}$ motor átvit. tényező

$T_M = \frac{\Theta R_a}{k_1 k_2 \Phi^2}$ mech időáll.

$T_V = \frac{L_a}{R_a}$ vill időáll.

$$\frac{\omega(s)}{m_t(s)} = \frac{A_T(1 + sT_V)}{1 + sT_M + s^2 T_V T_M}$$

$A_T = \frac{-R_a}{k_1 k_2 \Phi^2}$

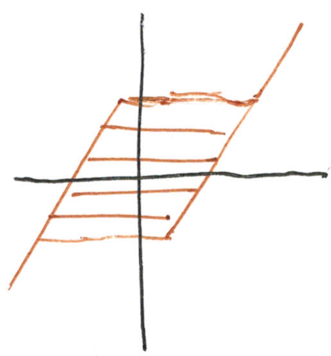
$\zeta = \frac{1}{2}\sqrt{\frac{T_M}{T_V}}$ csillapítási tényező ($T_M < 4T_V$ esetén lengő)

3.4 Nemlinearitások

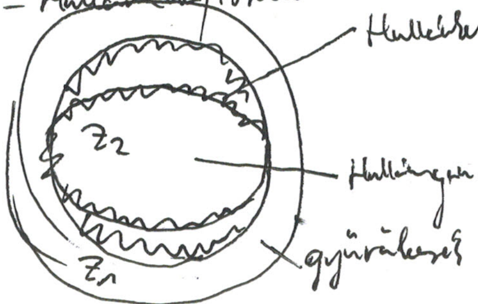
3.4.1. Kotyogás

Holtjáték

Statikus hiba akkor is, ha lineárisan 0 hibára szabályzunk



Megoldás
- Előfeszített fogaskerekek
- Hullámhajtómű



Hullámkerék: hullámgenerátor-ral ellentétes irányba indul

$\gamma = \frac{z_2}{z_1 - z_2}$ (lassító, ha $\gamma > 1$ ∵ gépjármű) áttétel utáni nyomaték

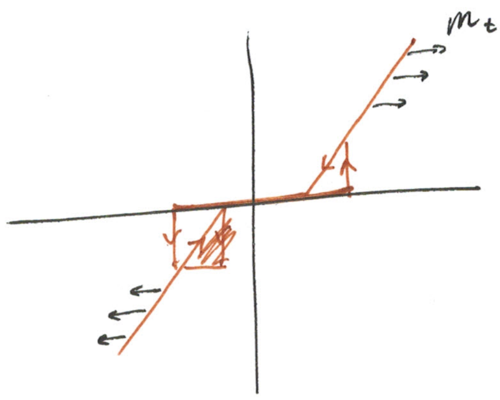
$$\tau = \gamma \cdot m \cdot \eta$$

veszteséges áttételnél áttétel hatásfok.

fékezésnél

$$\tau = \frac{\gamma \cdot m}{\eta}$$

3.4.2. Érzéketlenségi sáv



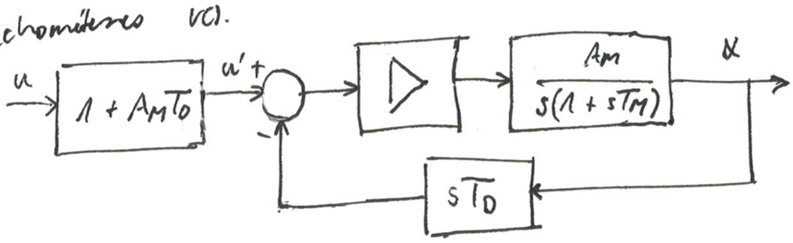
Dead zone, holtsáv

maradó hiba

súrlódás miatt

mo.:
- kis súrlódású áttétel
- kefe nélküli motor
- sebesség vcs.
- poz vcs.
- sw kompenzálás inverz karakterisztikával

Tachométeres vcs.



motor integráló marad

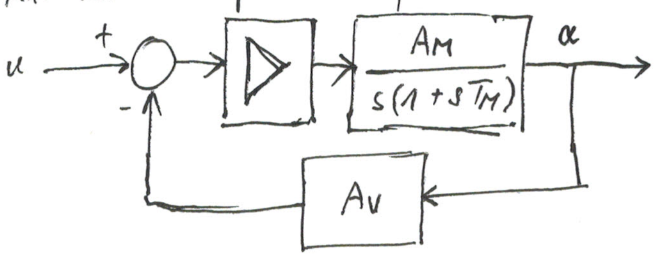
időállandója csökken

átviteli tényező csökken; vcs előtte kompenzáljuk ezért

↳ induláskor $(1 + A_M T_D)$-mor nagyobb jelet kap a motor (→ vcs.)

Helyzetbeállító: integráló típusú (kimenet: elmozdulás) arányos visszacsatolása

Ált. elrendelt pozícióérzékelő szerint



- arányosabb
- gyorsabb
- hibát nagyrészt kiszabályozza

3.4.3 Telítődés

Szaturáció, korlátozás

Hatásai:

D-hatás nem eléggé érvényesül

elintegrálódás

Kiküszöbölése lehet: alapjel meredekség korlátozás – megvezetés

Jobban közelíti pl a helyzetbeállító lin. modelljét.

3.5. Az elintegrálódás és kiküszöbölése

3.5.1. Az elintegrálódás

Integrator windup → Elintegrált állapotban a szabályozási kör nyitott

↓

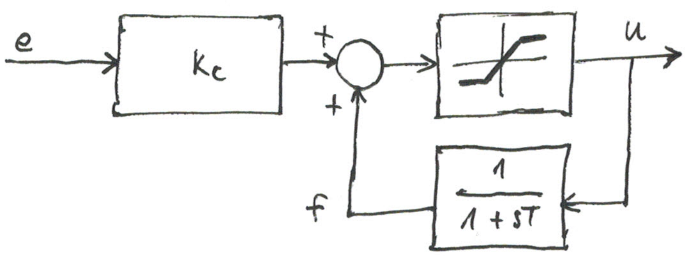
Időkiesés

→ Anti-windup kapcsolás

→ Korlátozások összehangolása (méretezés)

3.5.2. A FOXBORO szabályozás

Belső korlátozás beiktatása és illesztése



$$C(s) = k_c \cdot \frac{1}{1 - \frac{1}{1+sT}} = k_c\left(1 + \frac{1}{sT}\right)$$

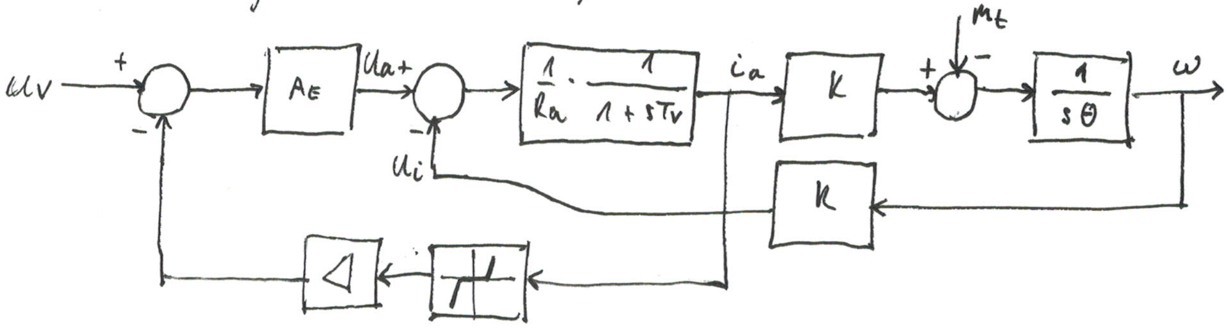
3.6 Szabályozási feladatok robotokban

Csuklószabályozás mintájára

3.6.1. Áramkorlátozás

Feszvezérelt motornál kell, mert kiég (tekercsmelegedés)

Áram érzéketlenségi sávon kívül negatívan visszacsatolva

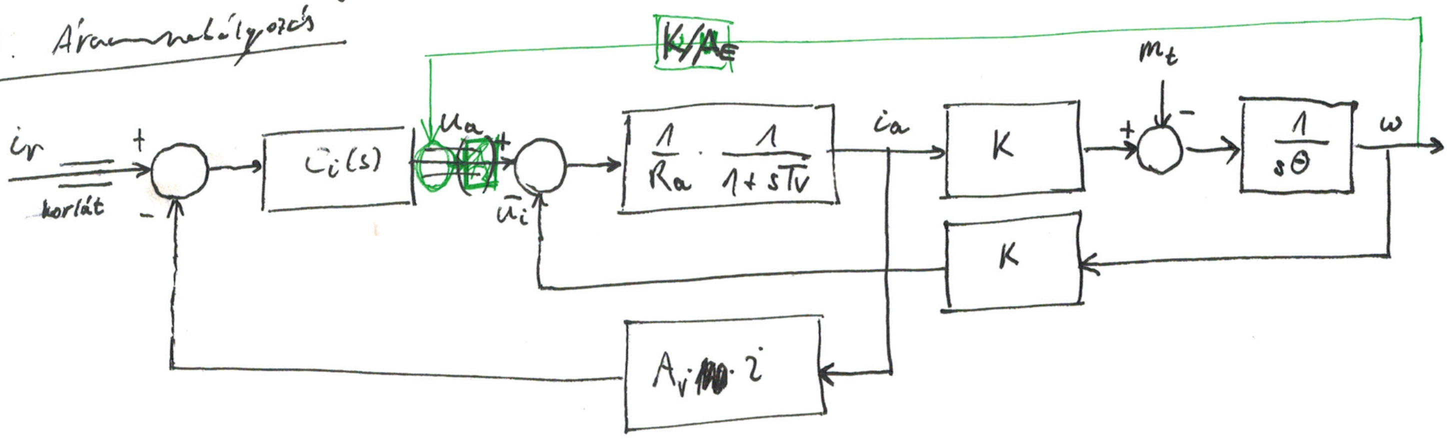


3.6.2 Nyomatékszabályozás

3 eset:

- ~~esetek~~ hatkomp. nyomatékm. → teljes robot szabályozására
- előreszámított nyomaték alapján beavatkozás (l. nemlineáris szétcsatolás)
- motor elektromágneses nyomatékának szabályozása: áramszabályozás

3.6.3. Áramszabályozás



$$L(s) = k_c \cdot \frac{1+sT_I}{sT_I} \cdot \frac{1}{1+sT_V} = k_c \cdot \frac{1}{sT_I}$$ hurok a villamos időállandóhoz méretezve

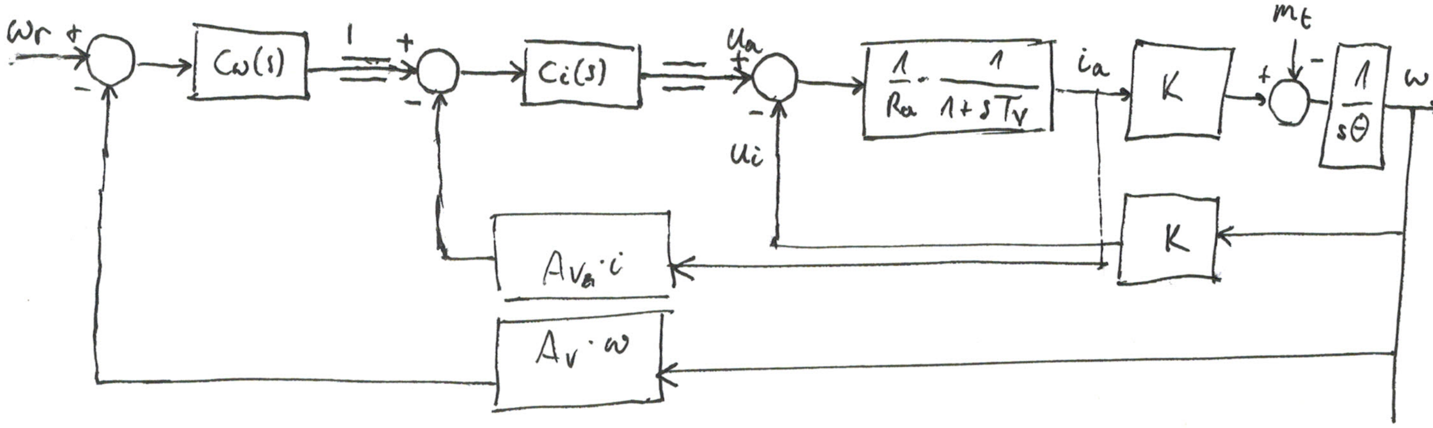
PI szab. időállandót $\frac{1}{k_c}$ arányban csökkenti

- Felgyorsul
- Túllendül
- Indukált fesz → zavaró hatás (statikus hiba, amíg gyorsul a motor) → zavarkompenzáció: indukált fesz nem mérhető, de vele arányos szögsebesség igen
- Elég az alapjelet korlátozni

3.6.4. Szögsebesség (ford. szám) szabályozás

Egyhurkos szabályozás és PI v. PID
Kell áramkorlátozás

Gyakoribb a kaszkád szabályozás
Alárendelt áramszabályozás
PI szab → villamos időállandót gyorsítja



Áramszabályozás eredője egytárolós arányos
Ezt látja a külső szabályozás

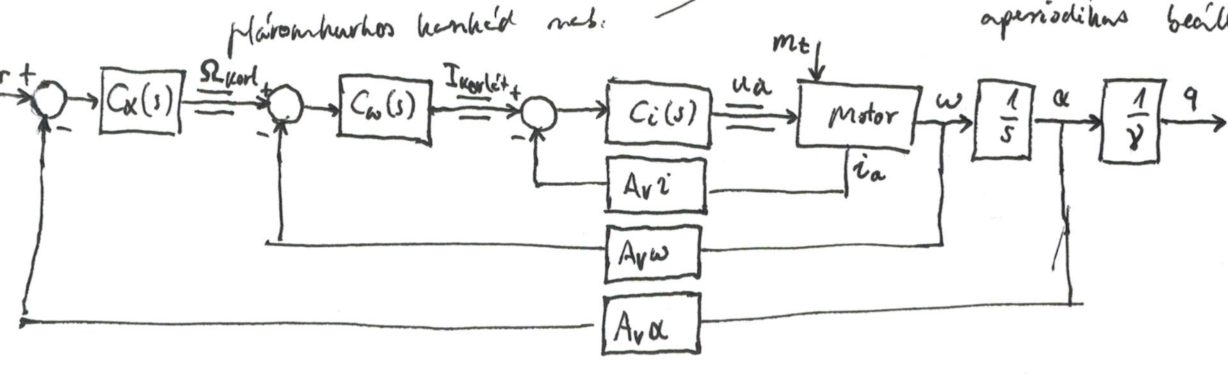
3.6.5 Pozíciószabályozás

Egyhurkos pozíciószab → folyamat kéttárolós integráló

Lehetőség:
Pozíciószabályozás alárendelt áramszabályozással
Kéthurkos kaszkád szabályozás alárendelt szögsebesség szabályozással
Szögsebesség szabályozó PI vagy PID
Belső kör eredője = kéttárolós arányos tag
↳ PD v. PI szab.
Kell áramkorlát külön

Belső PI: vill. időáll. gyorsítása
Áramkorlát: Ci alapjel korlát
Cα: ált PD, mert követelmény a csekély aperiodikus beállása

Háromhurkos kaszkád szab.



3.7 Szabályozó tervezés egyszerűen

Kísérleti módszerek

Ziegler-Nichols módszer

Arányos szabályozóval lengetés
Hurokerősítést változtatva keressük a kritikus körerősítést
Hátrány: állandósult lengés

Oppelt-módszer

Ugrójelre átmeneti fv
↳ Egytárolós közelítés
Arányos folyamatnál erre tervezhető szabályozó

3.7.1. Példa arányos folyamatra

$T_H = 3 \qquad T = 14 \qquad K_p = \frac{\Delta y}{\Delta u} = 2$

↓
Ideális holtidős folyamat PI-vel
↓
Nyitt kör: $L(s) = \frac{e^{-sT_H}}{sT_I}$

$T_I = 2T_H$ kb $60°$ fázistartalékot ad

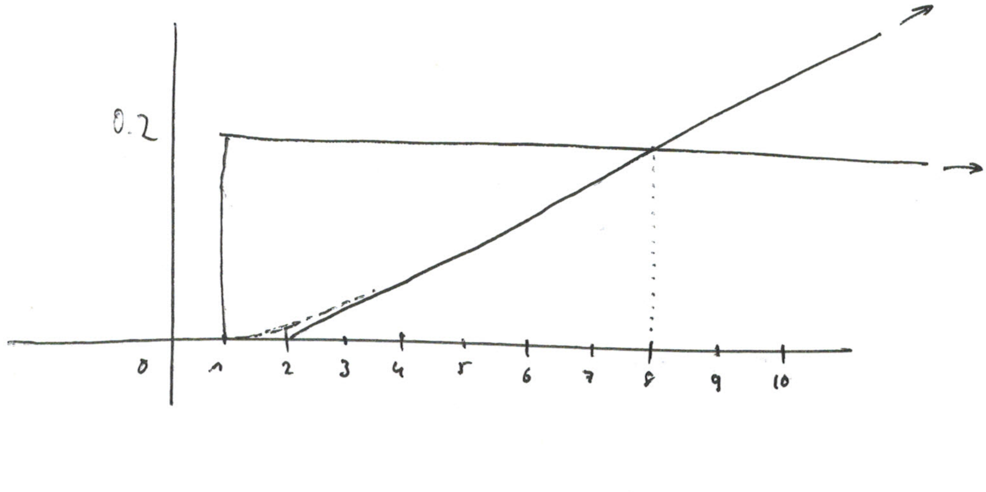
$$P(s) = K_p \cdot \frac{e^{-sT_H}}{1+sT} \longrightarrow PI: \quad PI(s) = K_c \cdot \frac{1+sT_I}{sT_I}$$

$$L(s) = P(s)PI(s) = K_c K_p \frac{e^{-sT_H}}{sT_I} \longrightarrow K_c = \frac{T_I}{2K_p T_H}$$

3.7.2. Példa integráló folyamatra

$$P(s) = \frac{1}{sT_I(1+sT)} \qquad T = 1 \quad T_I = 6$$

ehhez
$PI(s) = K_c \frac{1+10s}{10s}$



Papír-ceruza módszer

$\frac{1}{T_I} = \frac{1}{6}$ körfrekvencián egységnyi az erősítés

3.2. diasor szabályozó program része

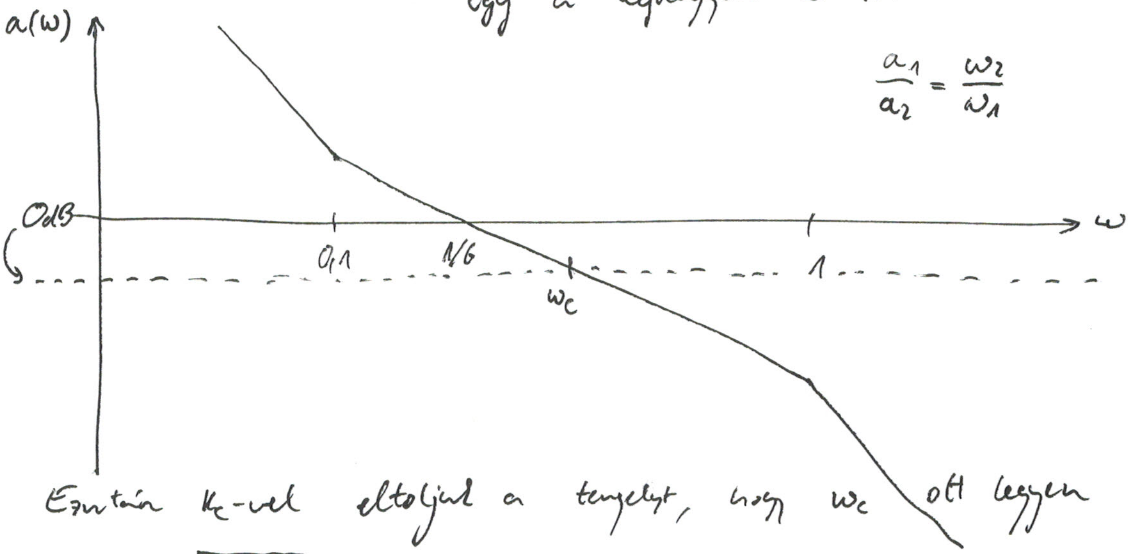
↳ A tároló töréspontja $\frac{1}{T} = 1$

↳ PI töréspontja $\frac{1}{10} = 0{,}1$ körfrekvencián egységnyi K_c-nél

↳ ω_c vágási körfrekvencia a $-20\frac{dB}{D}$-os szakasz közepére

így a legnagyobb a fázistartalék

$$\frac{a_1}{a_2} = \frac{\omega_2}{\omega_1}$$



Ezután K_c-vel eltolják a tengelyt, hogy ω_c ott legyen

$\omega_c = \sqrt{0{,}1 \cdot 1}$ (mért. közép)

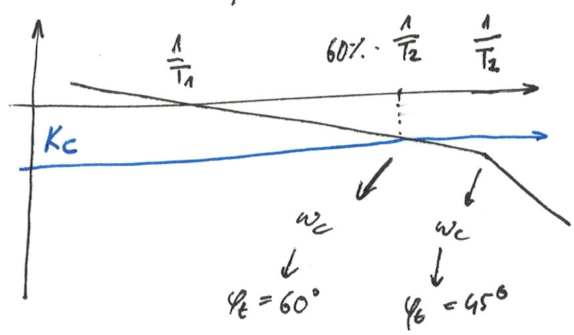
új erősítés $\frac{1}{6}$-nál K_c, ω_c-nél 1

$$\frac{\omega_c}{\frac{1}{6}} = \frac{K_c}{1} \rightarrow K_c = 6\omega_c = 6\sqrt{0{,}1} \approx 1{,}897$$

3.7.3 Bode-diagram alapján történő tervezés 1 típusú szabályozás esetén

Pl. PI hurok: $L = K_c \cdot PI \cdot P$

$$P = \frac{1}{(1+sT_1)(1+sT_2)} \rightarrow PI = \frac{1+sT_1}{sT_1}$$



Innen: $\frac{K_c}{1} = \frac{0{,}6 \cdot \frac{1}{T_2}}{\frac{1}{T_1}}$

$K_{c60°} = 0{,}6\frac{T_1}{T_2}$

$K_{c45°} = \frac{T_1}{T_2}$

$$\frac{L}{1+L} = \frac{K_c}{T_1 T_2 s^2 + T_1 s + K_c}$$

Aperiodikus esetben: $T_1^2 - 4T_1T_2K_c = 0$

tehát $K_{cap} = 0{,}25\frac{T_1}{T_2} \cong 0{,}4\, K_{c60}$

Sebesség - algoritmus

$$u_k - u_{k-1} = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} = \Delta u_k$$

$\downarrow$

Rekurzív pozíció algoritmus

$$u_k = u_{k-1} + b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2}$$

Telítődés kezelése

```
if(uk < Umin) uk = Umin;
if(uk > Umax) uk = Umax;
```

Integráló rekeszes PI

$$u_k = K_c \cdot e_k + K_i \cdot \sum_{j=0}^{k} e_j = K_c \cdot e_k + i_k \qquad \left(K_i = K_c \cdot \frac{T_s}{T_I}\right)$$

```
ek = rk - yk ;
ik = ik + Ki*ek ;
Vk = Kc*ek + ik ;
if (Vk < Umin) uk = Umin ;
else if(Umax < Vk) uk = Umax ;
else uk = Vk ;
ik = ik + (uk - Vk) ; // integrátor visszaállítás
```

FOXBORO PI szabályozó

$$\frac{f(z)}{u(z)} = \frac{(1-\beta) z^{-1}}{1-\beta z^{-1}} \qquad \beta = e^{-\frac{T_s}{T_I}}$$

$$f(z) = \beta z^{-1} f(z) + u(z) \cdot (1-\beta) z^{-1}$$

```
ek = rk - yk
uk = Kc*ek + Fk
if(uk < Umin) uk = Umin
if(Umax < uk) uk = Umax
fk = beta · fk + (1 - beta) * uk
```

3.8. Szabályozók programozása

3.8.1. Folytonos idejű szab.

a) $C(s) = k_c\left(1 + \frac{1}{sT_I} + sT_D\right)$ Folytonos PID

$$C(z) = \frac{u(z)}{e(z)} = k_c\left(1 + \frac{T_s}{T_I(1-z^{-1})} + \frac{T_D}{T_s}(1-z^{-1})\right) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1}}$$

$$s \to \frac{1-z^{-1}}{T_s}$$

Sebesség: $u_k - u_{k-1} = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} = \Delta u_k$

Pozíció: $u_k = u_{k+1} + b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2}$ rekurzív pozíció algoritmus

b) integráló rekurzív PI

$$u_k = K_c e_k + k_i \sum_{j=0}^{k} e_j = k e_k + i_k \qquad \left(k_i = k_c \cdot \frac{T_s}{T_I}\right)$$

nem rekurzív alg

$e_k = r_k - y_k$

$i_k = i_k + k_i \cdot e_k$

$v_k = k_c \cdot e_k + i_k$

$u_k = sat(u_{min}, v_k, u_{max})$

$i_k = i_k + (u_k - v_k)$ ← integrátor visszaállítás

c) Foxboro-san a visszacsatoló egytárolós tag SRE diszkrét alakja

$$\frac{f(z)}{u(z)} = \frac{(1-\beta)z^{-1}}{1-\beta z^{-1}} \quad \text{ahol } \beta = e^{-\frac{T_s}{T_I}}$$

$T_I \gg T_s$ esetén $e^x \simeq 1 + x$

rekurzív alg:

$e_k = r_k - y_k$

$u_k = k_c \cdot e_k + f_k$

$u_k = sat(u_{min}, u_k, u_{max})$

$f_k = beta \cdot f_k + (1-beta) \cdot u_k$

rekurzív, így a végén kezeli az elintegrálódást

3.8.2. Diszkrét idejű szabályozás

Ha a mintavétel frekvenciája közel van a vágási frekvenciához → DI szabályozó.

ideális PID → közelítő PID: elöjöhet lehet, nem két mintavételből deriválja

Rekurzív pozíció algoritmussal működik

3.8.3. Lökésmentes indítás

Automata szabályozásnál kimenetje milyen alapjelre nem változna a kézi irányítójel. Indulásnál határozatlan változó értéke beállítható.

Robotokban nincs ilyen → fékek elengedése után a leggyorsabb szabályozás indul

Egyhurkos

Szögsebesség

$$\frac{1}{1+sT_M+s^2T_MT_V} \quad \text{motor átv. időfv-e} = \frac{\omega(s)}{u_a(s)}$$

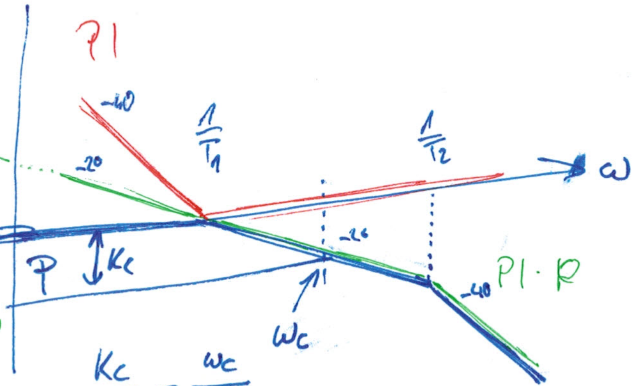
$\rightarrow \simeq \dfrac{1}{1+sT_m}$ durva közelítés

$\rightarrow \dfrac{1}{(1+sT_1)(1+sT_2)}$

$T_1 \approx T_m - T_v \qquad T_2 \approx T_v$

$PI = \dfrac{1+sT_1}{sT_1}$

$C = K_c \cdot PI$

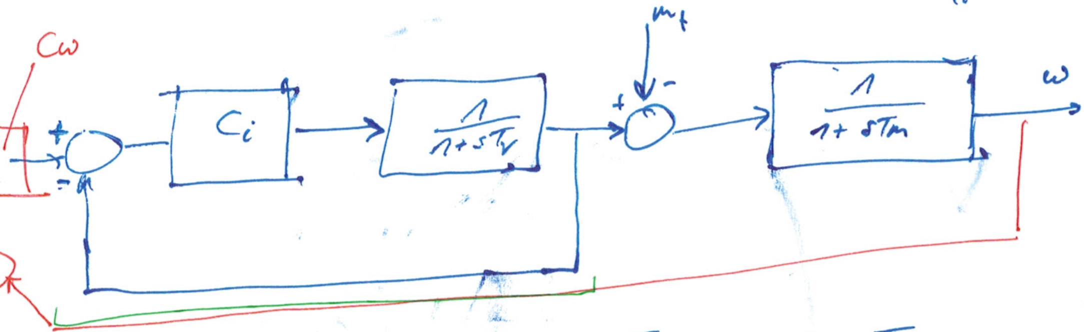


Ez a végtelen biztosítja a 0 statikus hibát

$\dfrac{K_c}{1} = \dfrac{\omega_c}{1/T_1}$

$\omega_c = 0{,}6 \cdot \dfrac{T_1}{T_2} \rightarrow \varphi_t = 60°$ és 10% túllendülés

$0{,}25 \cdot \dfrac{T_1}{T_2} \rightarrow$ aperiodikus beállás $\varphi_t \approx 70..80\%$



$C_I = K_{cI} \cdot PI = K_{ci} \cdot \dfrac{1+sT_I}{sT_I}$; $T_I = T_V$

$C_I(s) = K_{cI} \dfrac{1+sT_I}{sT_I} \cdot \dfrac{1}{1+sT_V} = \dfrac{K_{cI}}{sT_V} \rightarrow W_{zárt} = \dfrac{1}{1+\dfrac{sT_V}{K_{cI}}}$

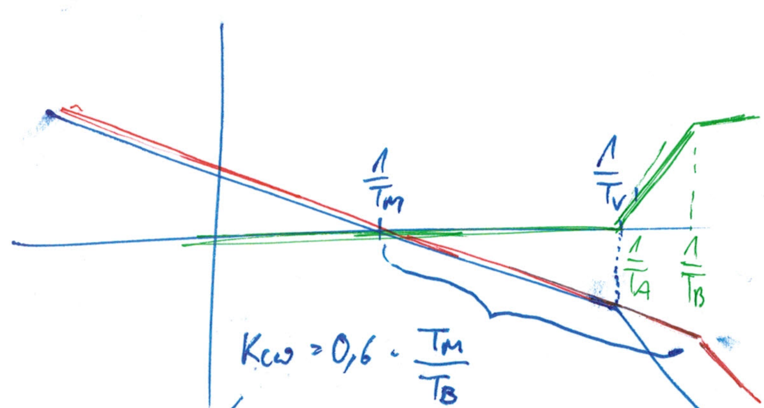
$P_I = \dfrac{1}{1+sT_V'}$

Részletesen Cω külső szabályozást

$P_\omega = \dfrac{1}{1+sT_V'} \cdot \dfrac{1}{sT_M}$

a) $C_\omega = K_{c\omega} \cdot PD$; $PD = \dfrac{1+sT_A}{1+sT_B}$

$\dfrac{1}{T_A} \quad \dfrac{1}{T_B}$



$K_{c\omega} = 0{,}6 \cdot \dfrac{T_M}{T_B}$

60° φ_t-hez 0,25

↘ Aperiodikus beállás

$T_A = T_V'$

$T_B = \dfrac{T_V'}{5}$ ($\frac{1}{5}$ tipikus)

$\_\_\!\!\lceil\_\_ \rightarrow err = 0$

$\_\_/ \rightarrow err \neq 0$

De az 1-es típusú szabályozónak a terhelő nyomatéka az integráló hatás előtt hat (nyomaték, és nem ω) → maradó hiba lesz

↓

2-es típusú szabályozással megoldható

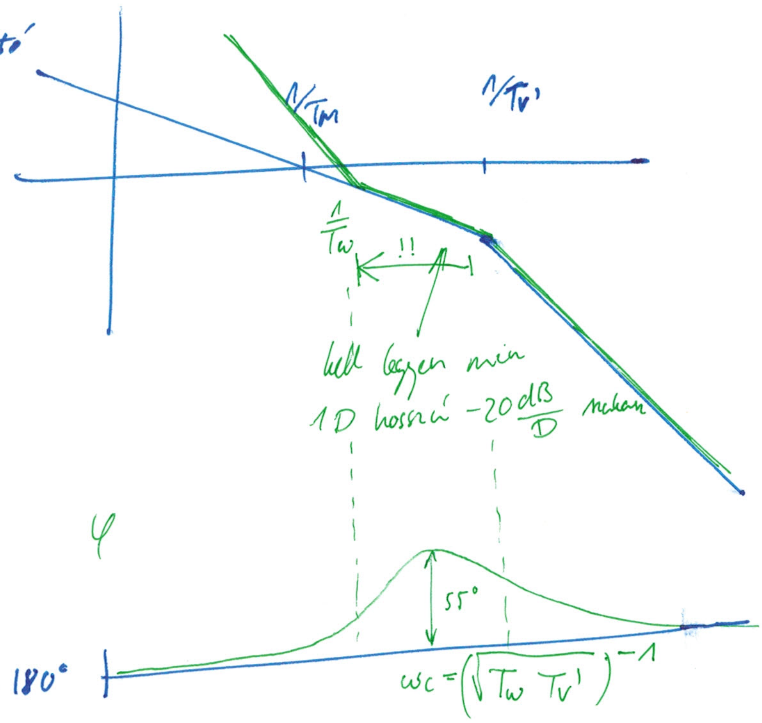
$C_\omega = K_{C\omega} \cdot PI$

$PI = \frac{1 + sT_\omega}{sT_\omega}$ $1/T_{I\omega}$

$T_\omega > T_v'$

$T_\omega \cong T_v' \cdot 10$

$\varphi_m = 55°$



$\_\_\!\!\lceil\_\_$ $\varepsilon = 0$

$\_\_/$ $\varepsilon = 0$

mt { $\_\_\!\!\lceil\_\_$ $\varepsilon = 0$; $\_\_/$ $\varepsilon \neq 0$ }

Hátrányok: – Lassú
– Jócskán túllendül, hacsak nem lassítsuk le nagyon-nagyon
Sokkor integrálóval célzottal már nem praktikus

TODO integrátorrekeszes algoritmus

4. ROBOTPROGRAMOZÁSI NYELVEK

4.1. A NOKIA-PUMA robot programozása

4.1.1. A programrendszer

ARPS advanced robot programming system

1. Programozási nyelv
   compiler + interpreter
2. Operációs rendszer
   real time
   foreground vor
      high prio → robot program
      low prio → másik megszakítható
   - program futtatás
   - komm
   - periféria-kezelés
   - self-check
3. Editor
4. Debugger

Decentralizált csuklóvezérlés feletti pozícióirányítás
Pálya: pontokkal időfüggés: sebességgel megadva

4.1.2. Az ARPS használata

Indítás
Betanítás
  Pult/monitor parancs/Editor
Program... írás
  futtatás
  leáll/folytat/újraindít

Kikapcsolás

Különböző koordinátákban mozgatható

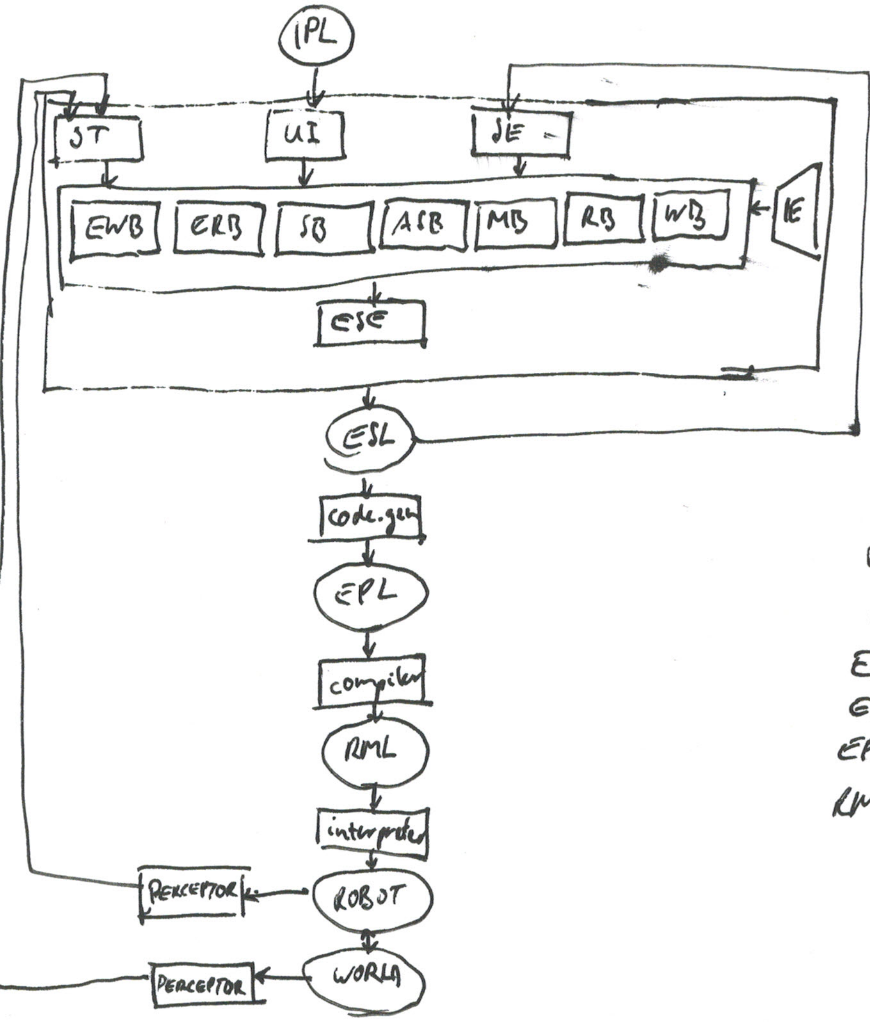
4.1.3. A programozási nyelv elemei

monitor parancs v. program utasítás

argumentumok az előbbiekhez

- egész állandó vagy változó
- decimális állandó (∄ float)
- tárolság
- szög
- csuklóindex
- digitális i/o
- sebesség
- programkapcsolás (ENA.../DISA...)
  BREAK, DIST10, ...
- operátorok
  arit./logi.
- szöveg
- címke
- pontok

4.1.4. Programozási példák



IPL implicit programming language
UI user interface
ST symbol transformer
SE skeleton extractor
EWB error world base
ERB error robot base
SB skeleton base
ASB action sequence base
MB motion base
RB robot base
WB world base
IE inference engine
ESE explicit solution extractor
ESL explicit solution language
EPL explicit programming language
RML robot motion language

4.2. ARPS továbbfejlesztése

4.2.1. A fejlesztés sarokpontjai

1. Szabad mozgás → decent. csuklónak.
   Áramgenerátoros szervomotorok
   Betöltés QNX fájlkezelővel
2. Szabad mozgás → intelligens algoritmusokkal
   Jacobi és H inverziónak számításával
3. QNX4 → QNX6
4. Hatlomp. erő- és nyomaték érz.
5. Hibrid pozíció- és erőir.
   Csak a geometriát kell megadni

4.2.2. HARPS programrendszer

Eredeti utasítások
+ Erő- nyomatékkorlátos utasítások
  ↳ Átkapcsol hibrid irányításra
Hibrid erőir. utasítások
Korlátozásstruktúrát frissítő utasítások
Erő/nyomatékbeolvasás
Hibrid koord.rsz. és term. korl. meghatározására utasítások

4.3. Robotprogramozási nyelvek osztályozása

On-line
+ kerekek egyszerűen megadhatók
+ egyszerű kezelés
− termeléskiesés

Off-line
+ nincs termeléskiesés
− komoly sw támogatás kell
− keretek bevitele nehéz

Pályamegadás szerint

- Explicit → betanítás/leírás (ARPS)
- Implicit → alkatrészmodell, pályadef

Explicit osztályozás

USA

1. HW/MCU szint
   ASM, C
   opc RTOS
   lehető legegyszerűbb
   ok, ha nem kell átprogramozni
2. Pont-pont ir. szint
   csuklókoord.-ban mozog
   Egyszerű nyelv
3. Egyszerű mozgásvezérlés
   Világkoordinátákban
   Pályatervezés
   Ugrás, utasítások → programstrukt.
   opc. Frame koncepció
   opc. párhuzamosság elér
4. Strukturált programozás
   struktúrák, adattípusok
   keretek, koord.-transzf.
   külső érzékelők (látó)
   Path.
   Mérnöki nehézséggel kell
5. Feladatorientált szint
   Dinamikus világmodell
   Összetett érzékelők (látórsz.)
   Bonyolult nehézségi rsz
   Implicit nyelv

EU

1. Csukló szint (USA-2) JOINT
2. Robotkar (USA-3 – USA-4)
   MANIPULATOR
3. Munkadarab. OBJECT
   Offline prog. megoldások
   ↳ letölthető prog.-k generálása
4. Feladat. TASK (USA-5)
   Robot megkeresi a tárgyat
   Bonyolult nehézségi rendszer

4.4. Fejlődés irányai

Explicit nyelvek egységesítése
   SRL structured robot lang – kevertn.
   IRDATA – hw-független if-elem illeszt

Implicit progr.

5. MELFA ⊗

6. MOBIL ROBOTOK

6.1. Mobil robotok osztályozása

Levegő UAV / űrmodul

Víz AUV

Szárazföld

- Kerék
- Sín
- Járó

6.1.1. Kerekeken mozgó robotok mobilitása

Holonom → bármerre mozdulhat ritkább, drágább mecha

Anholonom pl. autó

Pl. differenciális hajtás

$$v = \frac{V_l + V_r}{2} \qquad \omega = \frac{V_r - V_l}{w} \qquad \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} v\cos\varphi \\ v\sin\varphi \\ \omega \end{bmatrix}$$

2 szabadsági fok

egyensúlyozás v. bolygókerék

négykerekű

ackermann-formáció: közös középpont körül forog mind a 4 kerék

Szinkron hajtás

mechanikus vagy vezérlő elektronikai úton biztosítható a csúszásmentesség

Jobb manőverezőképesség

MDOF

Túlhatározott szabadsági fok

Egyenetlen terepen a legjobb a több lágy felfüggesztésű egyenként forgatott kerék

6.1.2. Járó robotok

Beépített egyensúlyozó rsz-ük lehet

Legbonyolultabb: helyváltoztatás

Állatvilágból lopnak számos ötleteket

6.1.3. Moduláris robotok

SR: self reconfiguration

Modul: mechanikus funkció és információfeldolgozás

Rács vagy láncrendszerű kapcsolódás

↓ Jobb rekonfigurálhatóság ↓ Jobban generálható mozgást

6.2. Mobil robotok helyváltoztatása

Hol?
Hova?
Hogyan?
} 3 fő kérdés

Relatív pozícióméres

dedueed reckoning → dead-reckoning: korábbi ismert helyzetből követhetetés

- odometria: kerekek elmozdulása
- inerciális: kettős integrálás gyorsulásmérő, giroszkóppal

nőnek a mérési hibák

Abszolút pozícióméres

- aktív jelzőegység (beacon)
  beérkező 2/3+ jel irányából
  fény/hang/rádió
- mesterséges tereptárgyak felismerése
  itt is 3+ kell
  komolyabb műveletigény
- természetes tereptárgy
  kb. mint az előző
  ismerni kell a környezetet előre
- modell-felismerés
  szenzorral adatgyűjtés
  adatok környezeti modellre illesztése

6.2.1. Odometria

Ált. inkr. adó (vagy néha kódadó)

k: adó és megtett út faktora
D: kerékátmérő
C_{enc}: enkóder inkr/ford
$n_{áttét}$: áttét enkóder és hajtókerék közt

$$k = \frac{D \cdot \pi}{n_{áttét} \cdot C_{enc}}$$

$\Delta s_{l/r} = k \cdot N_{l/r,i}$ megtett utak l/r kerékre

$\Delta s_i = \dfrac{\Delta s_l + \Delta s_r}{2}$ robot hajtásközéppontja által megtett út

$\Delta \varphi_i = \dfrac{\Delta s_r - \Delta s_l}{w}$ orientáció változás

$$\varphi_i = \varphi_{i-1} + \Delta\varphi_i$$ orientáció

$$x_i = x_{i-1} + \Delta s_i \cdot \cos\varphi_i$$
$$y_i = y_{i-1} + \Delta s_i \cdot \sin\varphi_i$$

↑
Közelítés: előbb fordul, aztán megy

Közelítés: félfordulat, megy, maradékfordulat

$$\varphi_i' = \varphi_{i-1} + \Delta\varphi_i \cdot \frac{1}{2}$$
$$x_i = x_{i-1} + \Delta s_i \cdot \cos\varphi_i'$$
$$y_i = y_{i-1} + \Delta s_i \cdot \sin\varphi_i'$$
$$\varphi_i = \varphi_i' + \frac{\Delta\varphi_i}{2}$$

Feltételezés így is: elfordulás megtett útté transzformálható
↑
kerék

De hibák

Szisztematikus
- más kerékátmérők
- nem névleges kerékátmérő
- nem névleges keréktávolság
- nem névleges kerékelhelyezés
- véges érzékelőfelbontás
- ensódeo mintavételezés

} kalibrációval csökkenthető hatás

Nem szisztematikus
- Talaj egyenetlenség
- kerekek csúszása
- kerék érintési pontjának bizonytalansága

UMBmark hibadefiníciók

$$E_d = \frac{D_r}{D_l} \qquad E_w = \frac{W_{act}}{W_{nom}}$$

Kalibráció: négyzet alapú pályát kell a robotnak bejárnia

6.2.2. Odometria optikai szenzorokkal

Talajeltmozdulás mérése
Ha az előző módszer túl pontatlan lenne

6.2.3. Inerciális helymeghatározás

Giroszkóp → szögelfordulás

Gyorsulásérzékelő

↳ Felületi mikromechanikai eljárással kapacitív elvű

↳ Piezorezisztív elvű

Hídban felfüggesztett tömeg

Híd ellenállásainak mérése

Giroszkóp

nagyobb tömeg, fordulatszám → jobb iránytartás

– elektromechanikus

– optikai/lézer giroszkóp

– fiber optic : fáziskülönbség $\Delta\varphi$

– ring laser : hullámhossz $\Delta\lambda$

forgás esetén $\Delta\varphi$ vagy $\Delta\lambda$ arányos a forgás gyorsaságával

előny: időmérésre visszavezethető

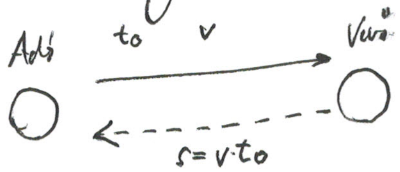
azt tudjuk a legpontosabban

– MEMS vibrációs : coriolis erőt mérik

IMU inertial measurement unit

Teljes giroszkóp + gyorsulásérzékelő egység

6.2.4. Az ultrahangos távolságmérés elve



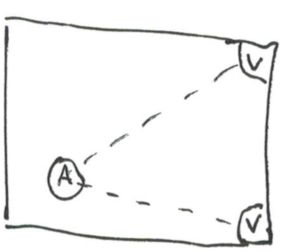
Trilateráció vagy trianguláció

Hangsebesség hőmérsékletfüggő → mérni kell

Hiba a mérési területen egyenetlen

Rádiójellel jelezzük az adás kezdetét

↳ Pontos távolságmérés (1-2 cm)



6.2.5 GPS és DGPS

Régen doppler-effektuson alapult
 Gyors járműveken nem volt jó
NAVSTAR GPS (USA) az egyetlen teljes ilyen rendszer
GLONASS (RU)
Galileo (EU)
További rendszerek vannak
 Önálló lokálisak (Kína, India)
 GPS pontosságát növelők
Kódmodulált jel két frekvencián
Idősszinkronizációt tesz lehetővé
↳ Távolságmérés rádióhullám-terjedési sebesség alapján
 Lokális óra elcsúszása miatt még egy jel szükséges
 4 [távolság + delta] alapján pozíció meghatározható

P: precision → titkos
C/A: coarse acquisition → nyitott

Vétel
 Kódmérés:
 Saját előállított kóddal hasonlítja össze a vett jelet
 Több 10 m pontosság Bitidő: időmérés kvantuma
 Legtöbb eszköz ilyen
 Fázismérés
 Vivő fázisa vs. lokálisan előállított jel fázisa
 Akár mm pontosságú

$$\Delta t' = \Delta t + \delta = \Delta t + \delta_{sat} - \delta_{recv}$$
$$D_i = c \cdot \Delta t' = d_i + c \cdot \delta = d_i + c \cdot (\delta_{sat} - \delta_{recv})$$
$$D_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} + c(\delta_{sat} + \delta_{recv})$$
d_i: valódi távolság

D-csatorna: $5 \cdot 300\,\text{bit} / 50\,\frac{\text{bit}}{\text{s}} = 30\,\text{s}$
 Műhold órapontossági paraméterei
 Durva pályaadatok (almanach) hidegindításhoz
 Fedélzeti pályaadatok (broadcast ephemeris) becsléshez
 Ionoszférikus modell paraméterei
 Műholdállapot
DGPS: differenciális: kis földrajzi régióban kb konstans hibával mérhetők a műholdtávolságok. Álló ismert helyzetű vevővel ez mérhető.

Eredetileg külön DGPS kódvétel

↳ Eddigi rendszerekhez képest valós idejű

Korrekciós szabványok

Mobiltelefon hálózat

Kereskedelmi rádió RDS segédfr.-án

Távközlő műholdon → Inmarsat

Spec. rádióadó

Több referenciaponttal javítható a rendszer

Korrekciós

Műhold pályaadatok	15 min refresh	1-2 min	broadcast
Regionális atmoszféramodell	60 min refresh	1-2 min	b/c 8 param
Egyes műholdak órajavító param.		~1 sec	b/c

↳ SA (civil pontatlanság) miatt

Végül 0,5 – 2 m pontosságot ad kódmérésnél

(SA torzítás 2000-ig 50-200 m pontosságot adott.)

Fázismérést is enged két GPS-vevővel (egyik fix) alkalmazzák

Relatív helymeghatározás

Így használható ki jól a fázisméréssel nyert pontosság

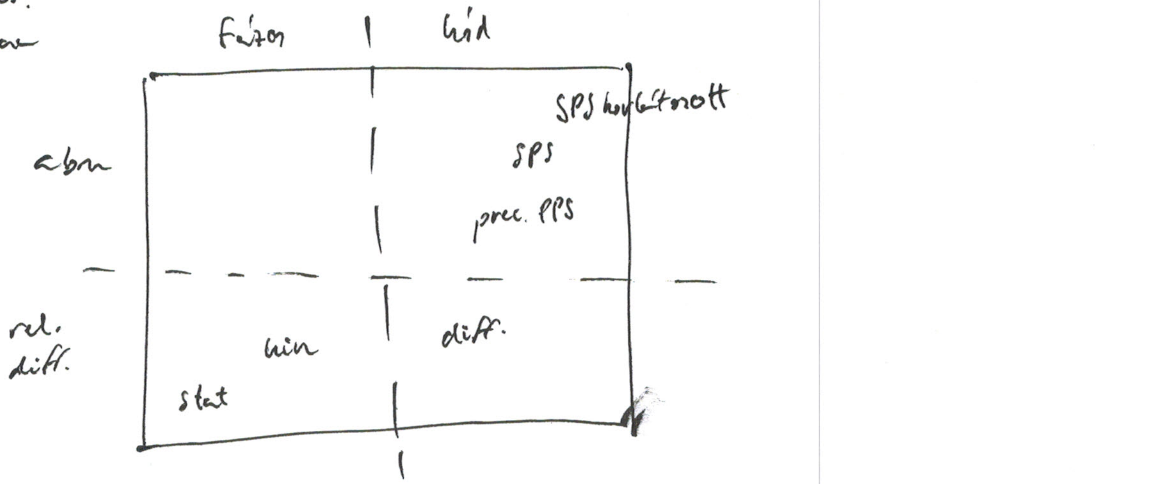
Statikus abszolút pontosság: 5 – 10 m (kommersz)

Kinematikus abszolút 10 – 100 m

Statikus relatív 0,1 – 1 ppm néhány mm / néhány km (kinematikus is pár cm)

Fázismérés!

Geodéziában



6.3 Mobil robotok navigációja akadályok között

Alapfeladatok

- Lokalizáció (Hol?)
- Célmegadás (Hová?)
- Térképalkotás (Hol van akadály?)
- Mozgástervezés (Hogyan? Merre?)
- Mozgásvégrehajtás (Szabályozás)

Autonóm mobil robot: több feladatot hajt meg a fentiek közül

0. Mozgásszabályozás. Nem autonóm.
1. Autonóm mozgástervezés. Mozgásirányítás
2. Autonóm térképalkotás. Alkalmazkodás környezetváltozáshoz. Önvezető járműfejlesztés célja ezt elérni
3. Teljes autonómia. Magasabb szintek között megfogalmazva a feladat

Mozgástervezés:

Állapottrajektória. Felbontható

- Pályatervezés → geometriai pálya
- Időparaméterezés (pályageometria és robot dinamikai tulajdonságai)

6.3.1. Konfigurációs tér

Kinematikai megközelítés: Bármikor megállítható robot

Állapot: pozíció és orientáció, de nem érdekel a deriváltjai

Szabadon mozgó robotnál (ami merev test)

elég a poz. + ori.

Robotkaroknál ez több csuklópozíciót is jelenthet és nem mindegy, melyik

n-dimenziós vektor egyértelműen jellemzi a helyzetet

(pl síkban 3)

Környezet is egyszerűbben a térben leírható

Start

Cél

Ütközésben levő/nem megengedett konfigurációk

Könnyebbség: a robot egyetlen pont a térben.

6.3.2. Mozgástervezési módszerek osztályozása

Kategorizálás
- Konfigurációs tér dimenziószáma, 2, 3, 3+
- Konfigurált akadályalakok bonyolultsága szerint (sokszög, térbeli élek, tetszőleges)
- Robot alakja (pontszerű, tetszőleges)
- Robot kinematikája (holonomikus v. nem)
- Lokalitás (lokális, globális)

Globális pályatervezés
- Ismert környezettérkép
- Tervezés/végrehajtás időben elkülönül
- Tervezés teljes konfigurációs térben
- Ált. offline alkalmazva
- Nem várt akadály ált. teljes újratervezést igényel
  - Ettől eltekintve a megoldás garantálható

Lokális
- Reaktív / „obstacle avoidance"
- Megoldás: érzékelés és gyors reakció
- Nem igényel térképet
- Tervezés + végrehajtás
- Nem várt változásokra tud reagálni vagy ismeretlen környezetet kezel
- Online
- Teljes navigációra általános esetben nem alkalmas
- Egyszerű robotmodell ált., egyszerűsíti a teljes konfig. teret.

6.3.3 Pályatervezési módszerek

- Konfigurációs teret gráffá képezi le
- Csúcsok: pontok Élek: 1 dimenziós utak
- Gráfban keresünk
- Egzakt módszer: teljes pályamodell kell, de mindig ad megoldást
- Mintavételes: elég közelítő modell, de az hiba nem elég
  - Sűrűbb vétellel javítható, de nem biztos hogy optimális
  - Nem is tudja, hogy van-e még jobb megoldás

6.3.3.1 Láthatósági gráf

Gráf akadályok geometriai modellje alapján
Akadályok közelítése sokszöggel
Csomópont: sokszögek csúcsai
Élek: láthatósági élek, hosszal súlyozva
Hagyományos gráfon belüli úthemező algoritmus pl Dijkstra
Akadályokat súrolja → előzetes akadály-növelés
Túl sok él → redukálás
Csak reflex csúcsokat nézünk (belső szög < 180°)
Élek: azonos sokszög szomszédos reflex csúcspontjai felé
különböző sokszögek reflex csúcspontjai közt, ha az
él egyenese nem metszi egyik sokszöget se (bitangenciális élek)
Ez bizonyíthatóan csak fölös éleket hagy el

6.3.3.2 Cella-dekompozíció

Szabad térrészekre koncentrál
Konfigurációs teret teljesen szabad/teljesen foglalt cellákra bontja
Cellaszomszédosság alapján topológiai gráfot ad
1. Kezdő és célkonfigurációt cellához kell rendelni
2. Topológiai gráfon útvonalkeresés
3. Útelőírás cellákról valóságba
pl középpontok összekötése

2D-ben:
Vertikális celladekompozíció
Feltételezés: akadályok határvonala sokszög
csúcspontokból függőleges határvonal az első akadályig fel és le is
konvex trapézokra osztotta a teret
Változatok főleg a cellafelosztásban különböznek

6.3.3.3 Rapidly exploring dense trees

Előző kettő teljes pontos modellt igényelt
Ez nem, de cserébe nem garantál véges idő alatt megoldást
Mintavételes

1. Vegyünk ki egyenletes mintákat
2. Próbáljuk a kiválasztott konfigurációt a korábbi gráffal összekötni egy vagy több lokális pályával (ált lin. interpol. → egyenes mentén) és itt nem kell az akadályokat figyelembe venni
3. Most vizsgáljuk a mintát. Ha egy pontja ütközik, eldobjuk

RDT:
Kísérőleg fa struktúrájú gráfot épít
Ált. egyenletes eloszlású véletlen mintákkal → rap. expl. random. trs. RRT

1. Mintavét → q_{rand}
2. Csomópont választás: a gráfban lévőkből a legközelebbi opc. élek belső pontjait is visszaadhatja → q_{near}
3. Lokális pályával összekötés. Ha ütközésmentes, bővítjük a gráfot q_{rand} ($\leftarrow q_{near}$) ponttal (éllel)
Ha ütközik, akkor az utolsó ütközésmentes pontig terjesztjük ki.

Variáns: két fa start-ból/célból növesztve
felismerni, ha összeér!
Sampling bias: néha véletlen minta helyett konkrétan a célt nézzük
Új lefutás adhat új eredményt.
Hosszabb futás adhat jobb eredményt
De általában elég gyors
Bármilyen konfig térben jó.

6.3.4. Akadályelkerülési módszerek

- Érzékelési/beavatkozási periódusidő
- Észlelt mozgás < érzékelési távolság
- Beavatkozó jel ütközésmentesen és a cél felé.

6.3.4.1. Virtuális erőtér

VFF virtual force field

akadályok taszít
cél: vonz

Eredetileg ehhez hisztogram rács
 Szenzoradatok alapján a ~~frissít~~
 Foglaltság valószínűsége ~~a~~

A rács lokális „aktív ablakát" veszi csak figyelembe

Taszítóerő minden rácspontra külön majd összegezve $(\sim \frac{1}{r^3})$

Vonzóerő a cél felé $(\sim \frac{1}{r})$

gyors és biztonságos, de
- Lokális minimum csapda
- Közeli akadályok közé nem megy be
- Oszcilláció keskeny folyosónál

} : potenciálmező, nem ampl.-függő

6.3.4.2. Vektormező hisztogram

VFH virtual field histogram

Előző hátrányai módosíttatta
 Túlredukált adat

Kétlépcsős redukció
 Környezet leírása ugyanolyan hisztogrammal és aktív ablakkal
 1. Egydimenziós poláris hisztogramot készít (poláris akadálysűrűség)
 2. Ez alapján választ irányt

$$\beta_{ij} = \mathrm{atan2}(y_j - y_0,\ x_i - x_0) \qquad m_{ij} = c_{ij}^{*2}(a - b \cdot d_{ij})$$

szög; vekt.nagys.; Foglalts.; távolság

 3. Adott mogtartományokban összegzés lineárisan csökken a távolsággal

$a - b\, d_{max} = 0$

Esélyes völgyek és völgyek helyzetének → küszöbözés

Ezután dönt, hogy merre van a cél
Szűk, széles; lehet plusz előírás a fentiek alapján

Kifinomultabb, de ez is be tud ragadni

6.3.4.3. Dinamikus ablak módszer

Sebességek terében pályatervezés irányítási célt
Gyorsulás korlátozás
Előnye a robot dinamikáját nem veszi figyelembe
DWA dynamic window approach

$$\dot{x}(t) = \cos(\varphi(t))\,v(t)$$
$$\dot{y}(t) = \sin(\varphi(t))\,v(t)$$
$$\dot{\varphi}(t) = \omega(t)$$

Eredetileg nem holonomikus szinkron hajtású robotra
Robot pályája így körívdarabokkal leírható
Cél ezek meghatározása, ehhez v_k, ω_k szög/sebességek
Ezek terében vezet be dinamikus ablakot
Ezen belül keres ütközésmentes megoldást
és szükség esetén megállást
Optimalizálás célfüggvénnyel
- Haladás a cél irányába
- Távolság akadályoktól
- Nagy sebesség