

% A feladat beadása előtt ellenőrizze (*Validate*), hogy minden cella kiértékelése hiba nélkül lefut. Érdemes először **újraindítani a kernelt** (a menüből *Kernel*→*Restart*) és utána **az összes cellát futtatni** (a menüből *Cell*→*Run All*). (A feladat hibás értékekkel is beadható...)

A feladat beadása után (*Assignments*→*Downloaded assignments: Submit*) készítsen egy PDF-et (*File*→*Save and Export Notebook as*→*PDF*) és ezt a PDF-et töltse fel a Moodle-be a megfelelő szorgalmi feladat beadásához. **Ügyeljen a határidők betartására!**

Ellenőrizze, hogy minden olyan cellát megfelelően kitöltött, ahol **YOUR CODE HERE**, **YOUR ANSWER HERE** vagy "..." szerepel, különben a feladat **hiányos**, és az ilyen *Házi feladat nem értékelhetőnek* minősül! (A *No answer given* figyelmeztetések eltávolíthatók értelemszerűen.)

*Kijelentem, hogy az ebben a dolgozatban közölt, az eredetit, segédletűl szolgálót kiegészítő gondolatmenet a saját megértésemet tükrözi, meg nem engedett eszközök, segítség használata nélkül készített önálló munka.*

Készítette: Rohrsetzer Róbert

In [1]: NEPTUN = "VN48J4";

warning: using the gnuplot graphics toolkit is discouraged

The gnuplot graphics toolkit is not actively maintained and has a number of limitations that are unlikely to be fixed. Communication with gnuplot uses a one-directional pipe and limited information is passed back to the Octave interpreter so most changes made interactively in the plot window will not be reflected in the graphics properties managed by Octave. For example, if the plot window is closed with a mouse click, Octave will not be notified and will not update its internal list of open figure windows. We recommend using the qt toolkit instead.

---

% Mechanikai rendszerek és robotok dinamikája - BMEGEMMNMMR

% Házi feladat - !!!Több verzió esetén Csak a legutóbbi verziót kell kitölteni!!!

## Feladatleírás{-}

Vizsgáljuk a 6 tengelyű IRB 120 típusú ABB robotot.



A színezett részek az egyes tagok modellezésére használt egyszerűsített geometriát jelölik:

- az 1-es, 3-as, 5-ös tagok (piros, sárga, barna) hengeres test ( $D_i, L_i$  paraméterek),
- a 2-es és 4-es tagok (kék és zöld) téglatest ( $L_i, a_i, b_i$  paraméterek),
- a feladatban a végberendezést jelentő 6. tagot csak az  $E$ -vel jelölt TCP koordinátarendszerén keresztül vesszük figyelembe az egyszerűség kedvéért.

## Adatok{-}

A robot műszaki adatlapja alapján az ábrán jelölt méretek táblázatosan összefoglalva:

ABB típus	$h_1$	$D_1$	$L_1$	$h_2$	$L_2$	$a_2$	$b_2$	$h_3$	$D_3$	$h_4$	$L_{35}$	$L_{34}$	$h_5$	$L_5$	$a_4$	$b_4$	$D_5$
IRB 120	187	140	100	103	270	100	40	70	140	134	463	302	72	110	120	90	70

továbbá  $L_3$  és  $L_4$ , a 3-as és 4-es tagot alkotó testek hossza, az alábbi képletek szerint számítható:

$$L_3 = L_{35} - L_{34} - h_5 + h_4 = 223\text{mm}, \quad L_4 = L_{35} - L_3 - h_5 = 168\text{mm}$$

In [2]: % a geometriai méretek globális változói:

```
global h1 D1 L1 L2 a2 b2 h3 D3 L3 h4 h5 L35 L34 L4 L5 a4 b4 D5
```

```
h1=187;
D1=140;
L1=100;
h2=103;
L2=270;
a2=100;
b2=40;
h3=70;
D3=140;
h4=134;
L35=463;
L34=302;
h5=72;
L5=110;
a4=120;
b4=90;
D5=70;
```

```
L3=L35-L34-h5+h4
L4=L35-L3-h5
```

```
L3 = 223
L4 = 168
```

In [3]: libhash;

```
assert( h1*D1*L1 + h2*L2*a2*b2 + h3*D3*L3 + h4*L4*a4*b4 + h5*L5*D5 == 359727400, "HIBÁS adatok!")
assert( L3*L4 == 37464, "L3 és/vagy L4 hibás!" )
```

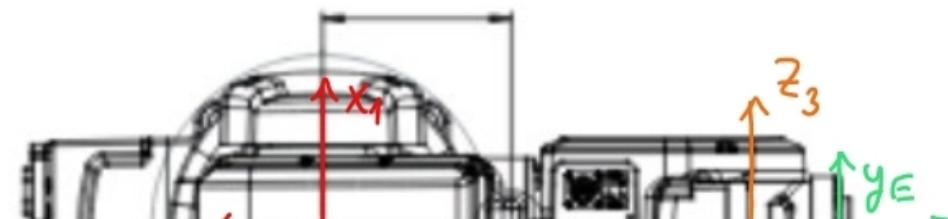
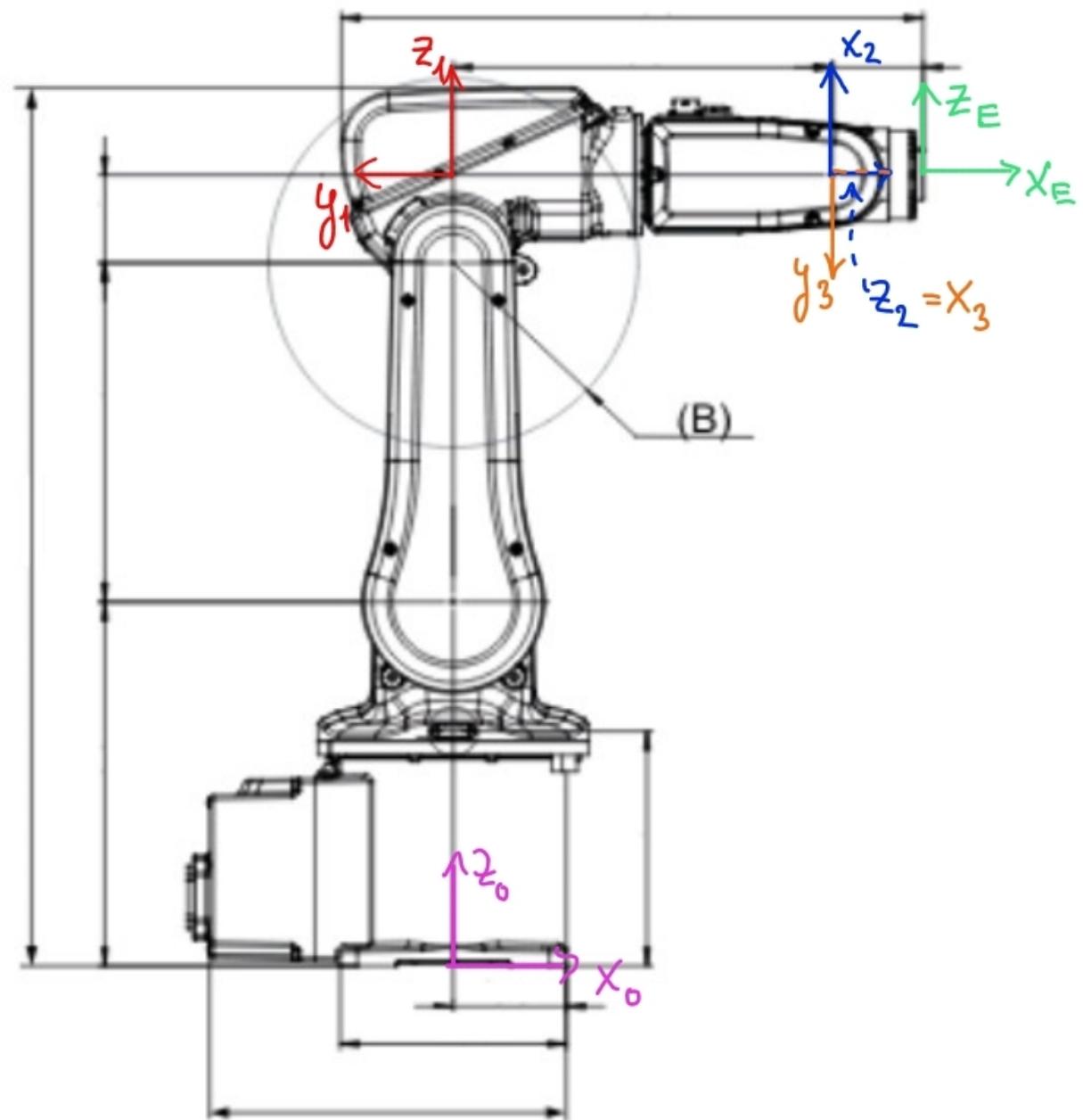
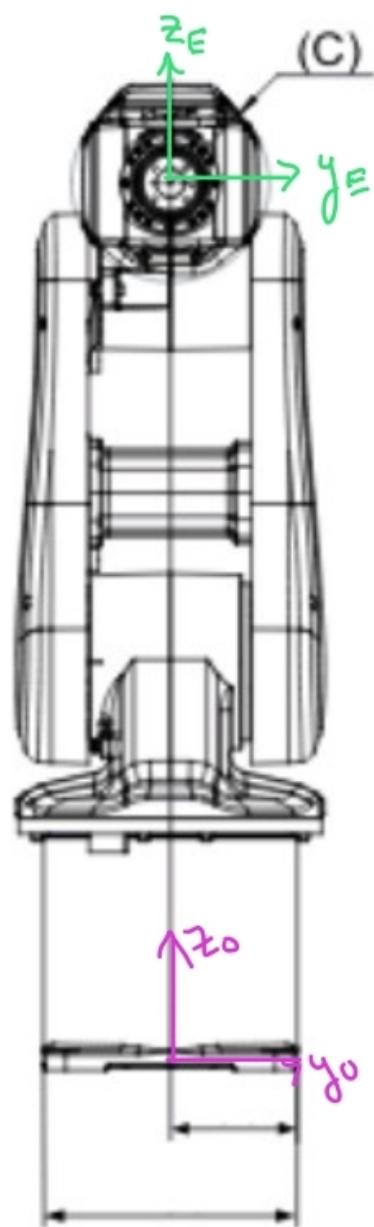
## Geometriai vizsgálat

Jelölje  $\hat{q}_i$  a robot eredeti csuklóváltozót ( $i = 1, \dots, 6$ ), melyek közül  $\hat{q}_2$ -t és  $\hat{q}_3$ -at a HOME pozícióban rögzítjük, és a továbbiakban az alábbi jelöléseket használjuk:

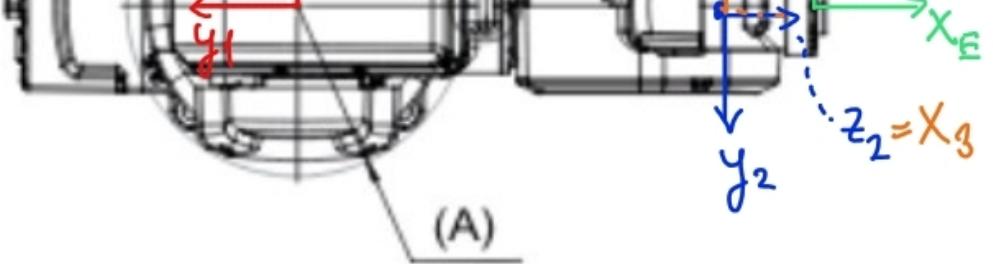
$$q_1 = \hat{q}_1, \quad q_2 = \hat{q}_4, \quad q_3 = \hat{q}_5, \quad (\mathbf{q} = [q_i] \in \mathbb{R}^3)$$

Tehát az 1-es, 2-es és 3-as (piros, kék illetve sárga) tagok egymáshoz képesti, relatív helyzete az előző ábrán vázolt módon rögzített, ami által egy 3 DoF soros robotot kaptunk. Továbbá mostantól ezeket az együtt mozgó tagokat jelöljük 1-gyel (piros-kék-sárga tag), 2-vel és 3-mal (zöld illetve barna színű tagok).

Az így kapott három mozgó taghoz rögzített lokális koordinátarendszereket (KR) a Denavit-Hartenberg paraméterezés Khalil-féle módosított konvenciója szerint felvéve az alábbi ábra jelöli. (Az  $x$  és  $z$  tengelyek értelme úgy lett megválasztva, hogy a méreteket tartalmazó korábbi ábrán pirossal bejelölt irányok valamelyikével legyenek párhuzamosak és megegyező értelműek, akárcsak az állvány KR<sub>0</sub> és a TCP-hez tartozó KR<sub>E</sub> tengelyeinek esetében.)



*Dobosi*



A  $q_i$  csuklókoordináták értékét a fenti ábrán vázolt helyzetben zérusnak tekintve (HOME pozíció) a **D-H paramétereket** az alábbi táblázat foglalja össze, ami a robot adatlapja alapján tartalmazza a csuklóváltozók értelmezési tartományának határait is:

Csukló ( $i$ )	$a_i$	$\alpha_i$	$d_i$	$\vartheta_i$	$q_{i,min}$	$q_{i,max}$
1: ${}^0T_1$	0	0	$h1 + h2 + L2 + h3$	$q(1) + 90$	-165	165
2: ${}^0T_2$	0	$\pi/2$	$L34$	$q(2) + 90$	-160	160
3: ${}^0T_3$	0	$\pi/2$	0	$q(3) + 90$	-120	120
E: ${}^3T_E$	$h5$	$\pi/2$	0	0	-	-

```
In [4]: function DH = DH_tab(q)
%
% A Denavit-Hartenberg PARAMÉTEREK FENTI TÁBLÁZATÁNAK MEGFELELŐ 4x6-os TÖMB
% a bemenő csukló változók értéke mm-ben és fokban
% a kimenő DH tábla elemei m-ben és rad-ban
%
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

DH=[0,0,(h1+h2+L2+h3)/1000,deg2rad(q(1)+90),-165,165;
    0,pi/2,L34/1000,deg2rad(q(2)+90),-160,160;
    0,pi/2,0,deg2rad(q(3)+90),-120,120;
    h5,pi/2,0,0,0,0];
DH(4,5:6) = [ -999 999 ];
end;
```

```
In [5]: _q20_ = [ 20 20 20 ];
assert_v( "%.4g:DH1", DH_tab(_q20_)(1,1:4), "DH tábla 1. sora HIBÁS!" );
assert_v( "%.4g:DH2", DH_tab(_q20_)(2,1:4), "DH tábla 2. sora HIBÁS!" );
assert_v( "%.4g:DH3", DH_tab(_q20_)(3,1:4), "DH tábla 3. sora HIBÁS!" );
assert_v( "%.4g:DH5", DH_tab(_q20_)(1:3,5:6), "qi-k min-max tartománya HIBÁS!" );
```

## Homogén transzformációk

Az alábbi cellák mutatják a Denavit-Hartenberg paraméterekkel általánosan felírt homogén transzformáció mátrixát, valamint az egyes KR-ek közötti transzformációkat az *eredeti geometriai paraméterekkel*, a trigonometrikus függvényekből az eltolást eltüntetve, a következő egyszerűsítő jelöléseket használva:

$$c_i = \cos q_i, \quad s_i = \sin q_i.$$

A homogén transzformáció mátrixának általános alakja:  $\theta_i$

$${}^j\mathbf{T}_i = S_x(a_i, \alpha_i)S_z(d_i, \vartheta_i) = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} & 0 & a_i \\ c_{\alpha_i}s_{\vartheta_i} & c_{\alpha_i}c_{\vartheta_i} & -s_{\alpha_i} & -d_i s_{\alpha_i} \\ s_{\alpha_i}s_{\vartheta_i} & s_{\alpha_i}c_{\vartheta_i} & c_{\alpha_i} & d_i c_{\alpha_i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

és az egyes transzformációs mátrixok:

$$\begin{bmatrix} a_1 \\ \alpha_1 \\ d_1 \\ \vartheta_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ h_1 + h_2 + L_2 + h_3 \\ q_1 + \pi/2 \end{bmatrix} \implies {}^0\mathbf{T}_1 = \begin{bmatrix} \cos(q_1 + \pi/2) & -\sin(q_1 + \pi/2) & 0 & 0 \\ \sin(q_1 + \pi/2) & \cos(q_1 + \pi/2) & 0 & 0 \\ 0 & 0 & 1 & \frac{h_1 + h_2 + L_2 + h_3}{1000} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} a_2 \\ \alpha_2 \\ d_2 \\ \vartheta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \pi/2 \\ L_{34} \\ q_2 + \pi/2 \end{bmatrix} \implies {}^1\mathbf{T}_2 = \begin{bmatrix} \cos(q_2 + \pi/2) & -\sin(q_2 + \pi/2) & 0 & 0 \\ 0 & 0 & -1 & \frac{-L_{34}}{1000} \\ \sin(q_2 + \pi/2) & \cos(q_2 + \pi/2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

valamint

$$\begin{bmatrix} a_3 \\ \alpha_3 \\ d_3 \\ \vartheta_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \pi/2 \\ 0 \\ q_3 + \pi/2 \end{bmatrix} \implies {}^2\mathbf{T}_3 = \begin{bmatrix} \cos(q_3 + \pi/2) & -\sin(q_3 + \pi/2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(q_3 + \pi/2) & \cos(q_3 + \pi/2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_4 \\ \alpha_4 \\ d_4 \\ \vartheta_4 \end{bmatrix} = \begin{bmatrix} h_5 \\ \pi/2 \\ 0 \\ 0 \end{bmatrix} \implies {}^3\mathbf{T}_E = \begin{bmatrix} 1 & 0 & 0 & \frac{h_5}{1000} \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A fenti összefüggések alapján a **homogén transzformációk mátrixait** megadó függvények kódja:

```
In [6]: function mx = Tr01(q1)

% A KÉRDÉSES HOMOGÉN TRANSZFORMÁCIÓ MÁTRIXA AZ ALÁBBI
% GLOBÁLIS VÁLTOZÓK ÉS A CSUKLÓPARAMÉTER SEGÍTSÉGÉVEL
% (a bemenő változók értéke mm-ben ill. fokban értendő)
%
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

q1=deg2rad(q1);
mx=[cos(q1+pi/2),-sin(q1+pi/2),0,0;
    sin(q1+pi/2),cos(q1+pi/2),0,0;
    0,0,1,(h1+h2+L2+h3)/1000;
    0,0,0,1];
assert( sprintf("%dx%d",rows(mx),columns(mx))=="4x4", "mx mérete hibás!" )
end;

function mx = Tr12(q2)
%
% A KÉRDÉSES HOMOGÉN TRANSZFORMÁCIÓ MÁTRIXA AZ ALÁBBI
% GLOBÁLIS VÁLTOZÓK ÉS A CSUKLÓPARAMÉTER SEGÍTSÉGÉVEL
% (a bemenő változók értéke mm-ben ill. fokban értendő)
%
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

q2=deg2rad(q2);
```

```

mx=[cos(q2+pi/2),-sin(q2+pi/2),0,0;
    0,0,-1,-L34/1000;
    sin(q2+pi/2),cos(q2+pi/2),0,0;
    0,0,0,1];
end;

function mx = Tr23(q3)
%
% A KÉRDÉSES HOMOGÉN TRANSZFORMÁCIÓ MÁTRIXA AZ ALÁBBI
% GLOBÁLIS VÁLTOZÓK ÉS A CSUKLÓPARAMÉTER SEGÍTSÉGÉVEL
% (a bemenő változók értéke mm-ben ill. fokban értendő)
%
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

q3=deg2rad(q3);
mx=[cos(q3+pi/2),-sin(q3+pi/2),0,0;
    0,0,-1,0;
    sin(q3+pi/2),cos(q3+pi/2),0,0;
    0,0,0,1];
end;

function mx = Tr3E()
%
% A KÉRDÉSES HOMOGÉN TRANSZFORMÁCIÓ MÁTRIXA AZ ALÁBBI
% GLOBÁLIS VÁLTOZÓK ÉS A CSUKLÓPARAMÉTER SEGÍTSÉGÉVEL
% (a bemenő változók értéke mm-ben ill. fokban értendő)
%
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

mx=[1,0,0,h5/1000;
    0,0,-1,0;
    0,1,0,0;
    0,0,0,1];
end;

```

```
In [7]: assert_v( "%.4g:Tr01(20)" );
assert_v( "%.4g:Tr12(20)" );
assert_v( "%.4g:Tr23(20)" );
assert_v( "%.4g:Tr3E(20)" );
```

## Direkt kinematikai feladat

A TCP  ${}^0\mathbf{r}_E$  helyvektorát megadó összefüggés a homogén transzformációkkal kifejezve:

$${}^0\mathbf{r}_E(\mathbf{q}) = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^3\mathbf{T}_2 {}^3\mathbf{T}_E {}^E\mathbf{r}_E$$

Ez alapján a **direkt kinematikai feladat (FKIN)** kódja és a TCP helyvektora

$$\mathbf{q}^* = \left[ \frac{\pi}{3} \quad \frac{\pi}{6} \quad -\frac{\pi}{6} \right]^\top$$

esetén:

```
In [8]: function r0E = FKin(q)
%
```

```
% a TCP helyvektora a homogén transzformációk függvényeivel kifejezve
% q: csuklóváltozók vektora fok-ban illetve mm-ben
% r0E: a TCP x,y,z koordinátáinak vektora m-ben a globális KR-ben (KR0)
%
```

```
rEE=[0;0;0;1];
r0=Tr01(q(1))*Tr12(q(2))*Tr23(q(3))*Tr3E()*rEE;
r0E=r0(1:3,1);
end;
```

```
% a q* vektor megadása:
qstar = [pi/3;pi/6;-pi/6];
qstar=rad2deg(qstar);
```

```
% a TCP kérdéses helyvektora:
r0E=FKIN(qstar)
```

r0E =

```
0.1978
0.3065
0.6612
```

In [9]: assert\_v( "FKIN(\_q20\_)", FKin(\_q20\_) )

A TCP kérdéses helyvektora tehát:

$${}^0\mathbf{r}_E(\mathbf{q}^*) = [ 0.1978 \quad 0.3065 \quad 0.6612 ]^T \text{mm.}$$

## Tengely/szög számítás

Az  ${}^0\mathbf{R}_3$  mátrixnak megfelelő forgatás előállítható az egyes homogén transzformációk forgatásainak sorozataként, azaz:

$${}^0\mathbf{R}_3 = {}^0\mathbf{R}_1^{-1} {}^1\mathbf{R}_2 {}^2\mathbf{R}_3$$

Egy adott  $\mathbf{n}$  tengely körül  $\varphi$  szögű forgatási transzformáció mátrixát megadó Rodriguez-formula:

$$\mathbf{R} = \mathbf{I} + (1 - \cos(\varphi))[\mathbf{n}\mathbf{x}]^2 + [\mathbf{n}\mathbf{x}] \sin(\varphi), \quad [\mathbf{n} \times] = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}$$

illetve a forgatási mátrixból meghatározható forgatás tengelye és annak szöge:

$$\cos \varphi = \frac{\mathbf{R}_{11} - 1}{2}, \quad [\mathbf{n} \times] = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$$

Az  ${}^0\mathbf{R}_E$  forgatási mátrixot illetve a tengely/szög paramétereit megadó függvények kódjai és azok értéke a csuklóvektor fent megadott  $\mathbf{q}^*$  értéke esetén:

In [10]: function mx = Rot0E(q)

```
% az R_0E forgatási mátrix a homogén transzformációk függvényeivel kifejezve
```

```

% q: csuklóváltozók vektora mm-ben ill. fokban
%
T0E=Tr01(q(1))*Tr12(q(2))*Tr23(q(3))*Tr3E();
mx=T0E(1:3,1:3);
end;
%-----
function nx = v2nx( v ) %vetktor to matrix
%
% KERESZTSZORZÁS TENSOR: Length(v)=3, size(nx)= [3 3]
%
nx=[ 0,-v(3),v(2);
      v(3),0,-v(1);
      -v(2),v(1),0];
end;
%-----
function v = nx2v( nx ) %matrix to vektor
%
% KERESZTSZORZÁS TENSORBÓL VEKTOR
% Length(v)=3, size(nx)= [3 3]
%
assert( nx+nx'==zeros(3), "nx: nem keresztszorzás tensor!" )

v=[nx(3,2);nx(1,3);nx(2,1)];
end;
%-----
function rr = Rodriguez(fi,n)
%
% AZ n VÉKTOR KÖRÜLI fi[rad] SZÖGŰ FORGATÁS MÁTRIXA
%
assert( sumsq(n)==1, "n: nem egységvektor!" );

rr=eye+(1-cos(fi))*v2nx(n)*v2nx(n)+v2nx(n)*sin(fi);
end;
%-----
function [ fi, nvec ] = invRodriguez(rr)
%
% AZ rr MÁTRIXNAK MEGFELELŐ FORGATÁS TENGELY/SZÖG REPREZENTÁCIÓJA
%   fi: FORGATÁSI SZÖG
%   nvec: FORGATÁS TENGELYÉNEK IRÁNYVEKTORA
%
fi=(acos((trace(rr)-1)/2));

nvec=nx2v((rr-transpose(rr))/(2*sin(fi)));
end;
%-----
% AZ R_0E FORGATÁSI MÁTRIX ÉS TENGELY/SZÖG PARAMÉTEREI q* ESETÉN:
qstar
R_0E=Rot0E(qstar)
[theta,n]=invRodriguez(R_0E)

```

```

qstar =
 60.000
 30.000
-30.000

R_0E =
 0.6495 -0.7500  0.1250
 0.6250  0.4330 -0.6495
 0.4330  0.5000  0.7500

```

```
theta = 1.1415
```

```
n =
```

```
 0.6321
-0.1694
 0.7561
```

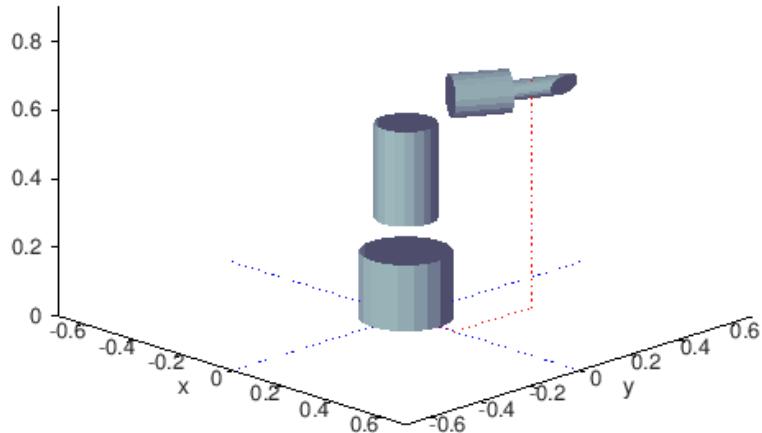
```
In [11]: assert_v( "invRodriguez( Rot0E(_q20_) )", invRodriguez( Rot0E(_q20_) ) )
```

Azaz a  $\mathbf{q}^*$  csuklóvektorhoz tartozó  ${}^0\mathbf{R}_E$  forgatási mátrix és a neki megfelelő tengely/szög paraméterek:

$${}^0\mathbf{R}_E = \begin{bmatrix} 0.6495 & -0.7500 & 0.1250 \\ 0.6250 & 0.4330 & -0.6495 \\ 0.4330 & 0.5000 & 0.7500 \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} 0.6321 \\ -0.1694 \\ 0.7561 \end{bmatrix}, \quad \varphi = 65.401^\circ$$

## A robotkarok helyzetének ábrázolása

```
In [12]: MRD_lib
plot_robot(qstar,[-0.7,0.7],[-0.7,0.7],[0,.9]);
grid; set(gca,"gridlinestyle","");
view([ 1 -1 .7 ])
```

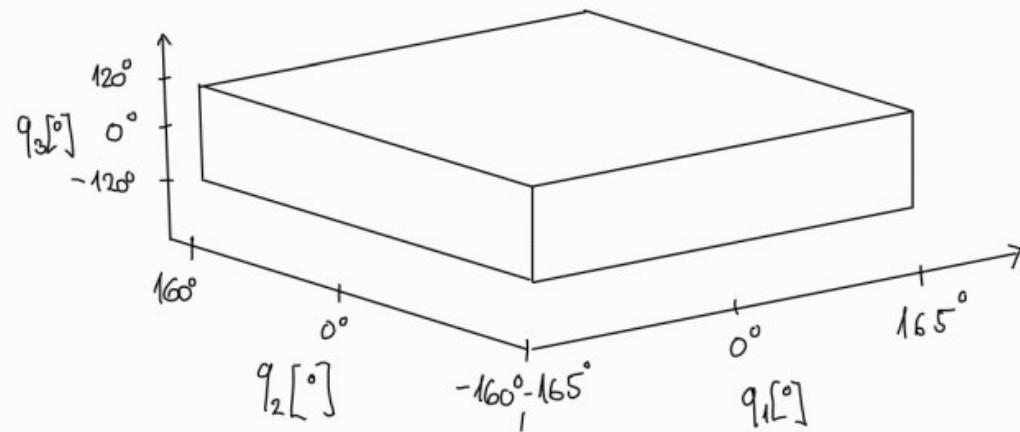


## Kinematikai vizsgálat (3 DoF)

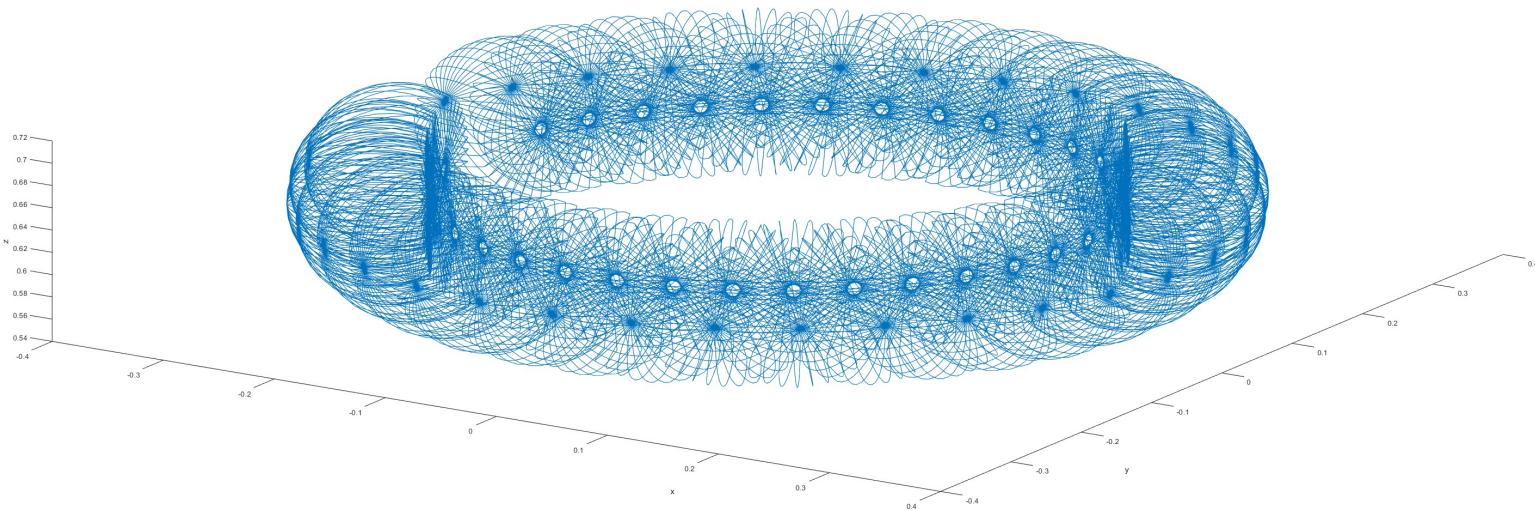
A korábban leírt módon korlátozott robot csukló- és munkaterének ábrája:

A munkatér mint geometriai alakzat tehát egy Tórusz.

Csuklóter



rTCP munkatér



A TCP helyvektora és Jacobi-mátrixa a  $q_1$ ,  $q_2$  és  $q_3$  paraméterekkel:

$${}^0\mathbf{r}_E(\mathbf{q}) = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_E {}^E\mathbf{r}_E = \begin{bmatrix} h_5 \frac{\cos(q_2)\cos(q_3) + \sin(q_1)\sin(q_2)\sin(q_3) + L_{34}\cos(q_2)}{1000} \\ -h_5 \frac{\cos(q_3)\sin(q_1) - \cos(q_1)\sin(q_2)\sin(q_3) + L_{34}\sin(q_1)}{1000} \\ \frac{L_2 + h_1 + h_2 + h_3 + (h_5\cos(q_2)\sin(q_3))}{1000} \end{bmatrix} \implies {}^0\mathbf{J}_E(q_1, q_2, q_3) = \begin{bmatrix} \frac{d^0\mathbf{r}_{Ex}}{dq_1} & \frac{d^0\mathbf{r}_{Ex}}{dq_2} & \frac{d^0\mathbf{r}_{Ex}}{dq_3} \\ \frac{d^0\mathbf{r}_{Ey}}{dq_1} & \frac{d^0\mathbf{r}_{Ey}}{dq_2} & \frac{d^0\mathbf{r}_{Ey}}{dq_3} \\ \frac{d^0\mathbf{r}_{Ez}}{dq_1} & \frac{d^0\mathbf{r}_{Ez}}{dq_2} & \frac{d^0\mathbf{r}_{Ez}}{dq_3} \end{bmatrix}$$

A TCP  ${}^0\mathbf{v}_E$  sebességvektora és a fenti Jacobi-mátrix közötti *elméleti* összefüggés:

$${}^0\mathbf{v}_E(\mathbf{q}, \dot{\mathbf{q}}) = {}^0\mathbf{J}_E(q_1, q_2, q_3)\dot{\mathbf{q}} = \begin{bmatrix} \frac{d^0\mathbf{r}_{Ex}}{dq_1} & \frac{d^0\mathbf{r}_{Ex}}{dq_2} & \frac{d^0\mathbf{r}_{Ex}}{dq_3} \\ \frac{d^0\mathbf{r}_{Ey}}{dq_1} & \frac{d^0\mathbf{r}_{Ey}}{dq_2} & \frac{d^0\mathbf{r}_{Ey}}{dq_3} \\ \frac{d^0\mathbf{r}_{Ez}}{dq_1} & \frac{d^0\mathbf{r}_{Ez}}{dq_2} & \frac{d^0\mathbf{r}_{Ez}}{dq_3} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

```
In [13]: function Jacobian = JvE(q)
%
% A ROBOT FENT KISZÁMOLT Jacobi MÁTRIXA:
%
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

q1 = deg2rad(q(1));
q2 = deg2rad(q(2));
q3 = deg2rad(q(3));

J1 = [- (L34*sin(q1))/1000 - (h5*cos(q3)*sin(q1))/1000 - (h5*cos(q1)*sin(q2)*sin(q3))/1000, -(h5*cos(q2)*sin(q1)*sin(q3))/1000, -(h5*(cos(q1)*sin(q3) + cos(q3)*sin(q1)))/1000;
J2 = [ (L34*cos(q1))/1000 + (h5*cos(q1)*cos(q3))/1000 - (h5*sin(q1)*sin(q2)*sin(q3))/1000, (h5*cos(q1)*cos(q2)*sin(q3))/1000, -(h5*(sin(q1)*sin(q3) - cos(q1)*cos(q3)))/1000];
J3 = [0, (h5*sin(q2)*sin(q3))/1000, -(h5*cos(q2)*cos(q3))/1000];

Jacobian = [J1;J2;J3];

end;

In [14]: assert_v( "JvE(_q20_)", JvE(_q20_) )
```

## Az inverz kinematika numerikus megoldása

Adott a TCP előírt pályájának  $\mathbf{r}(t)$  mozgástörvénye. Az inverz kinematikai feladat algoritmus a csuklósebességekre:

$$\dot{\mathbf{q}} = \mathbf{q}_E^{-1}(\mathbf{v}_d - K_\epsilon(\mathbf{r}_E(\mathbf{q}) - \mathbf{r}_d))$$

$$\mathbf{q}^{(n+1)} = \mathbf{q} + \dot{\mathbf{q}}\Delta t$$

```
In [15]: function qn = IKIN(rm,rn,qm,dt)
%
% qn: MEGHATÁROZANDÓ CSUKLÓVEKTOR (2DOF)
% rm: AKTUALÍS ELŐÍRT POZÍCIÓ HELYVEKTORA
% rn: KÖVETKEZŐ ELŐÍRT POZÍCIÓ
% qm: AKTUALÍS CSUKLÓVEKTOR (2DOF)
% dt: KÖVETKEZŐ IDŐLÉPÉS (MINTAVÉTEL) HOSSZA
%
% AZ ALÁBBI VÁLTOZÓK BEVEZETÉSÉVEL ADJA MEG qdot KIFEJEZÉSÉT:
```

```

% vd: AKTUÁLIS ELŐÍRT SEBESSÉGVEKTOR KÖZELÍTÉSE
% rq: AKTUÁLIS TÉNYLEGES POZÍCIÓ A CSUKLÓVÁLTOZÓKBÓL
% Jpi: A JACOBI MÁTRIX PSZEUDO-INVERZE
%
global cv % CSILLAPÍTÁSI (KONVERGENCIA) PARAMÉTER

J = JvE(qm); % Függvényt kell ami kiszámolja a Jacobi mátrixot

% Aktuális pozíció meghatározása a csuklóvektorokból
rq = FKIN(qm); % Függvényt kell ami kiszámolja az aktuális pozíciót

% Kívánt sebességek meghatározása az aktuális és a kívánt pozíció különbsége alapján
vd = (rn - rm)/dt;

% Csuklósebességek számítása

Jpi = pinv(J);
qdot = Jpi*(vd-cv*(rq-rn));

% Következő időlépéstre vonatkozó csuklópozíciók meghatározása
qn = qm + qdot * dt;

end;

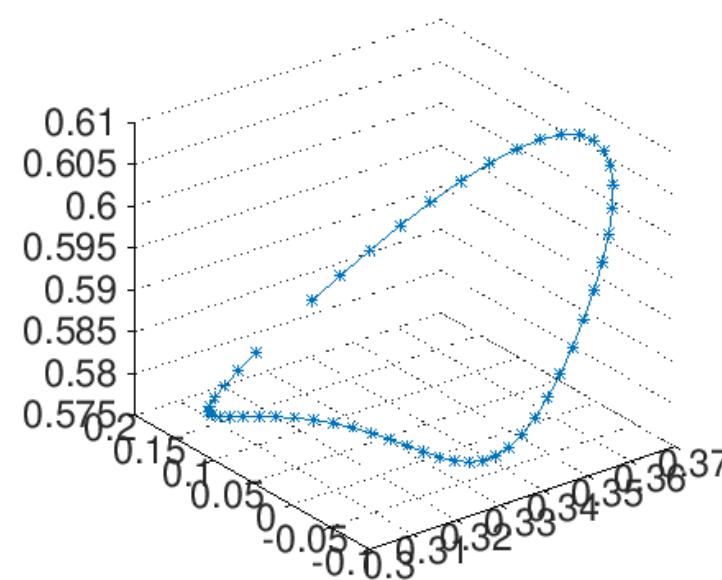
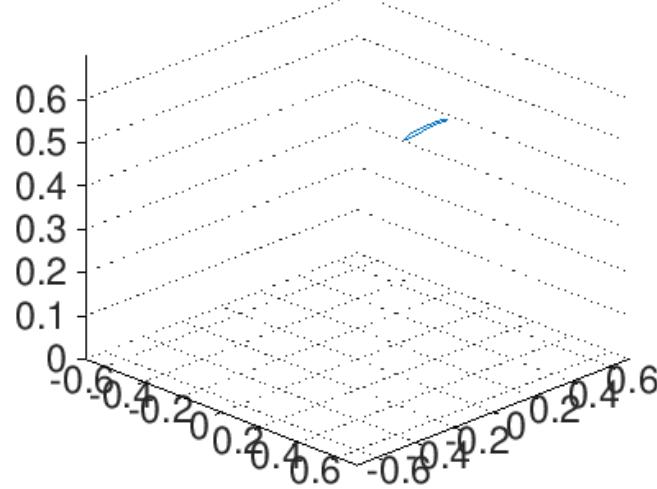
```

In [16]:

```

%
% AZ ELŐÍRT r(t) PÁLYA PONTJAIT AZ rDes TÖMBBEN TÁROLJUK:
%
t = linspace(0,1.2,51)';
rDes=[];
for i=1:length(t)
    rDes = [ rDes; FKIN([20*cos(5*t(i)),30+15*sin(5*t(i)),50+30*sin(5*t(i))])' ];
end;
%rDes = [ 200+40*cos(15*t) 200+50*sin(15*t) 630+30*sin(5*t) ]/1000;
figure(1,'position',[0,0,1000,400])
subplot(1,2,1)
plot3( rDes(:,1), rDes(:,2), rDes(:,3) )
xlim([-0.7,0.7]); ylim([-0.7,0.7])
zlim([0,.7]); grid; set(gca,"gridlinestyle",":")
view([ 1 -1 .7 ])
subplot(1,2,2)
plot3( rDes(:,1), rDes(:,2), rDes(:,3), '-*' )
grid; set(gca,"gridlinestyle",":")

```



```
In [17]: global cv
cv = 1850;

rn=[0 0 0];

qAct = [10 20 50];
rAct = FKIN(qAct)';

for i=1:length(t)-1
% Kiszámítani az új csuklóvektorokat a következő időlépésre (qn)
    qn = IKIN(rAct(i,:)', rDes(i,:)', qAct(i,:)', t(i+1)-t(i))';
    qAct = [qAct; qn];

    rAct = [rAct; FKIN(qn)'];

end;

ra = rAct(end,:);
qa = qAct(end,:);

while (abs(sum(rDes(end,:)-rn))>1e-5)
    qn = IKIN(ra',rDes(end,:)',qa',t(end)-t(end-1))';
    qa = qn;

    rn = FKIN(qn)';
    ra = rn;
end;

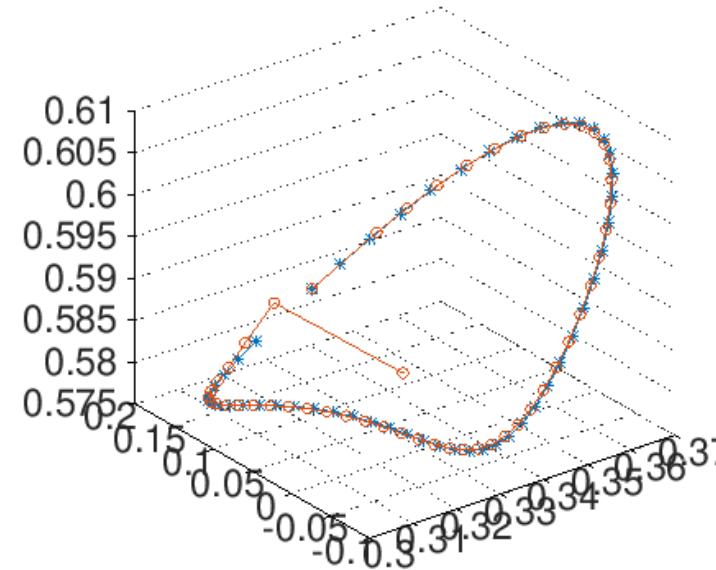
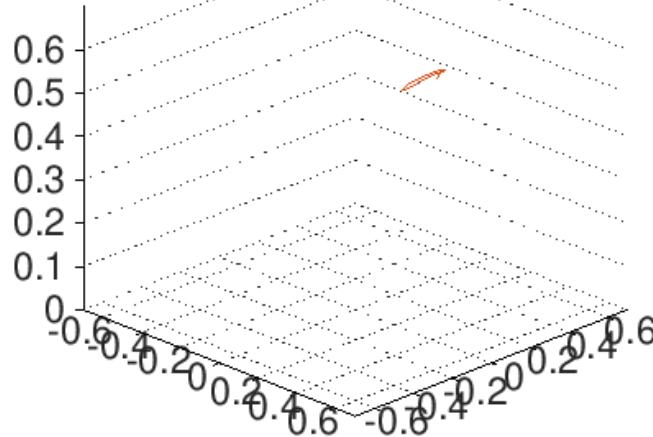
qAct(end,:) = qn;
rAct(end,:) = rn;

figure(1,'position',[0,0,1000,400])
subplot(1,2,1)
```

```

plot3( rDes(:,1), rDes(:,2), rDes(:,3) )
hold
plot3( rAct(:,1), rAct(:,2), rAct(:,3) )
xlim([-0.7,0.7]); ylim([-0.7,0.7])
zlim([0,.7]); grid; set(gca,"gridlinestyle",":")
view([ 1 -1 .7 ])
subplot(1,2,2)
plot3( rDes(:,1), rDes(:,2), rDes(:,3), '-*' )
hold
plot3( rAct(:,1), rAct(:,2), rAct(:,3), '-o' )
grid; set(gca,"gridlinestyle",":")

```



```
In [18]: assert( sumsq( ((rAct-rDes)(end,:)./rDes(end,:)) )<1e-5, "NEM ELÉG PONTOS!")
assert( norm( (rAct-rDes)(5,:) )/norm(rDes(5,:))<1e-3, "NEM ELÉG GYORS!" )
```

```
error: NEM ELÉG GYORS!
error: called from
    assert at line 101 column 11
```

## Dinamikai vizsgálat (3 DoF)

Modelezük az eredeti 1-es, 3-as, 5-ös tagokat  $\emptyset D_i \times L_i$  homogén hengeres testekkel, a 2-es és 4-es tagokat  $a_i \times b_i \times L_i$  méretű hasábokkal, melyeknek a sűrűsége azonos,  $\rho = 2850 \text{ kg/m}^3$ .

Az alábbi függvények az egyes tagok  $\hat{m}_i$  tömegét illetve  $\hat{\Theta}_i$  tehetetlenségi nyomatéki mátrixát határozzák meg:

```
In [19]: function m_i = Mass(i)
    global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

    % AZ i-edik TAG TÖMEGE [kg]
    % (HA i INDEXEK TÖMBJE, AKKOR A MEGFELELŐ TÖMEGEK TÖMBJE)
```

```

% (a 2-es tag tömege a két hasáb együttes tömege!)
%Henger
    ro=2850;
    m1=D1^2*pi/4*L1*ro;
    m3=D3^2*pi/4*L3*ro;
    m5=D5^2*pi/4*L5*ro;
%Hasáb
    m2=2*a2*b2*L2*ro;
    m4=a4*b4*L4*ro;
    mv=[m1;m2;m3;m4;m5]/1e9;

    m_i=mv(i);

end;

function th_i = Theta(i)
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

% AZ i-edik TAG TEHETETLENSÉGI NYOMATÉKI MÁTRIXA A MEGFELELŐ LOKÁLIS KR
% TENGELYEIVEL PÁRHUZAMOS SP-I TENGELYEKRE SZÁMÍTVA [kg m^2]
%Henger
m_1=Mass(1);
m_2=Mass(2);
m_3=Mass(3);
m_4=Mass(4);
m_5=Mass(5);

if i==1
    th_i=[ 1/2 * m_1 * (D1/2)^2, 0, 0;
            0, 1/12 * m_1 * (3*(D1/2)^2+L1^2), 0;
            0, 0, 1/12 * m_1 * (3*(D1/2)^2+L1^2)]/1e6;
end
if i==3
    th_i=[1/12*m_3*(3*(D3/2)^2 + L3^2), 0, 0;
            0, 1/2 * m_3 * (D3/2)^2, 0;
            0,0,1/12*m_3*(3*(D3/2)^2 + L3^2)]/1e6;
end
if i==5
    th_i=[1/2 * m_5* (D5/2)^2, 0, 0;
            0, 1/12*m_5*(3*(D5/2)^2 + L5^2), 0;
            0,0,1/12*m_5*(3*(D5/2)^2 + L5^2)]/1e6;
end
%Téglatest
if i==2
    m1=2850*L1/1000*a2/1000*L2/1000;
    mk=m_2+m1;

    th_1=[1/12*mk*(L2^2+a2^2),0,0;
            0,1/12*mk*(L2^2+(L1+2*b2)^2),0;
            0,0,1/12*mk*(a2^2+(L1+2*b2)^2)]/1e6;

    th_2=[1/12*m1*(L2^2+a2^2),0,0;
            0,1/12*m1*(L2^2+L1^2),0;
            0,0,1/12*m1*(L1^2+a2^2)]/1e6;

    th_i=th_1-th_2;

```

```

end
if i==4
    th_i=[1/12*m_4*(L4^2 + b4^2),0,0; %sztem itt b4 helyett a4 kéne
        0,1/12*m_4*(L4^2+a4^2),0; % sztem itt a4 helyett b4 kéne
        0,0,1/12*m_4*(b4^2+a4^2)]/1e6;
end

end;

```

In [20]:

```

assert_v( "%.4g:Mass(1:5)" )
assert_v( "%.4g:Theta(1)([1,5,9])" )
assert_v( "%.4g:Theta(2)([1,5,9])" )
assert_v( "%.4g:Theta(3)([1,5,9])" )
assert_v( "%.4g:Theta(4)([1,5,9])" )

```

## A 3 DoF modell tagjainak súlypontjai, tömegei

Jelölje  $m_1$ ,  $m_2$  és  $m_3$  az együtt mozgó merev részek tömegeit:

$$m_1 = \sum_{i=1}^3 \hat{m}_i, \quad m_2 = \hat{m}_4, \quad m_3 = \hat{m}_5$$

Ezek  ${}^1\mathbf{r}_1$ ,  ${}^2\mathbf{r}_2$  és  ${}^3\mathbf{r}_3$  súlyponti vektorai,

$${}^1\mathbf{r}_1 = \frac{m_1 \cdot {}^1\hat{\mathbf{r}}_1 + m_2 \cdot {}^1\hat{\mathbf{r}}_1 + m_3 \cdot {}^1\hat{\mathbf{r}}_1}{m_1 + m_2 + m_3}, \quad {}^2\mathbf{r}_2 = {}^2\hat{\mathbf{r}}_4, \quad {}^3\mathbf{r}_3 = {}^3\hat{\mathbf{r}}_5$$

és ez és ezek értékei a lokális KR-ekben:

$${}^1\mathbf{r}_1 = \begin{bmatrix} 0 \\ -0.0108 \\ -0.1355 \end{bmatrix} [\text{m}], \quad {}^2\mathbf{r}_2 = \begin{bmatrix} 0 \\ 0 \\ -0.084 \end{bmatrix} [\text{m}], \quad {}^3\mathbf{r}_3 = \begin{bmatrix} 0.017 \\ 0 \\ 0 \end{bmatrix} [\text{m}],$$

(az alábbi cellában elvégzett számítások alapján)

In [21]:

```

function r_i = CoG(i)
%
% AZ i-edik (i=1..5) TAG SP-i KOORDINÁTÁI A MEGFELELŐ (1-es, 2-es, vagy 3-as) LOKÁLIS KR-BEN [m]
% (HA i n ELEMŰ INDEX TÖMB, AKkor A SP-i KOORDINÁTÁK 3xn-es TÖMBJE)
%
global h1 D1 L1 h2 L2 a2 b2 h3 D3 L3 h4 h5 L34 L35 L4 L5 a4 b4 D5

r11=[0;0;-L2-h3]/1000;
r12=[0;0;-h3-L2/2]/1000;
r13=[0;-(h4-L3/2);0]/1000;
r24=[0;0;-L4/2]/1000;
r35=[h5-L5/2;0;0]/1000;
r=[r11,r12,r13,r24,r35];

r_i=r(:,i);

end;

```

```

function r_i = rSP(i)

% AZ m_i (i=1,2,3) TÖMEGŰ RÉSZ SP-i KOORDINÁTÁI KR_i-BEN [m]
% (HA i n ELEMŰ INDEX TÖMB, AKkor A SP-i KOORDINÁTÁK 3xn-es TÖMBJE)
%
m1=Mass(1);
m2=Mass(2);
m3=Mass(3);
m4=Mass(4);
m5=Mass(5);

rsp1=(m1*CoG(1)+m2*CoG(2)+m3*CoG(3))/(m1+m2+m3);
rsp2=CoG(4);
rsp3=CoG(5);
rsp=[rsp1,rsp2,rsp3];

r_i=rsp(:,i);

end;

% AZ r1, r2 és r3 SÚLYPONTI HELYVEKTOROK TEHÁT:
r1=rSP(1)
r2=rSP(2)
r3=rSP(3)

```

r1 =

```

0
-0.0108
-0.1355

```

r2 =

```

0
0
-0.084000

```

r3 =

```

0.017000
0
0

```

In [22]:

```

assert_v( "%.4g:rSP(1)" )
assert_v( "%.4g:rSP(2)" )
assert_v( "%.4g:rSP(3)" )

```

In [23]:

```

pkg load symbolic

% CSUKLÓVÁLTÓZÓK
syms q1 q2 q3

% D-H PARAMÉTEREK A SZIMBOLIKUS SZÁMÍTÁSOKHOZ
syms a1 a3 d1 d2 d3 alpha1 alpha2 alpha3 theta1 theta2 theta3
aa2 = sym('a2'); % az a2 numerikus változóval való ütközés miatt
Pi = sym(pi);

```

```

global ai di ali thi

% AZ ALÁBBI VEKTOROKBAN (ai,di,...,r1,r2,r3) A NULLA ILLETVE Pi-VEL KIFEJEZHETŐ
% ÉRTÉKÜ PARAMÉTEREK HELYÉRE ÍRJUNK 0-t ILLETVE MEGFELELŐ Pi HÁNYADOST;
% A d_i ILLETVE theta_i HELYÉRE PEDIG A DH_tab-BAN KORÁBBAN MEGÁLLAPÍTOTT q_i FÜGGÉST
% (Pl. q1-Pi )

%ai = [ a1 aa2 a3 ];
%di = [ d1 d2 d3 ];
%ali= [ alpha1 alpha2 alpha3 ];
%thi= [ theta1 theta2 theta3 ];

ai = [ 0 0 0 ];
di = [ d1 d2 0 ];
ali= [ 0 Pi/2 Pi/2];
thi= [ q1+Pi/2 q2+Pi/2 q3+Pi/2];

% LOKÁLIS SÚLYPONTI KOORDINÁTÁK
syms x1 x2 x3 y1 y2 y3 z1 z2 z3
%r1 = [ x1; y1; z1; 1 ];
%r2 = [ x2; y2; z2; 1 ];
%r3 = [ x3; y3; z3; 1 ];

r1 = [ 0; y1; z1; 1 ];
r2 = [ 0; 0; z2; 1 ];
r3 = [ x3; 0; 0; 1 ];

% SZIMBOLIKUS HOMOGÉN TR.
function ti = Ti(i,q)
global ai di ali thi

    ti= [ cos(thi(i)), -sin(thi(i)), 0, ai(i);
          cos(alii(i))*sin(thi(i)), cos(alii(i))*cos(thi(i)), -sin(alii(i)), -di(i)*sin(alii(i));
          sin(alii(i))*sin(thi(i)), sin(alii(i))*cos(thi(i)), cos(alii(i)), di(i)*cos(alii(i));
          0,0,0,sym(1)];

end;

% A SÚLYPONTI HELYVEKTOROK KR_0-BAN
r01 = Ti(1,0)*r1;
r02 = Ti(1,0)*Ti(2,0)*r2;
r03 = Ti(1,0)*Ti(2,0)*Ti(3,0)*r3;
% A SZÖGFÜGGVÉNYEK RÖVIDÍTÉSEI A SZEBB KIMENETHEZ:
syms c1 s1 c2 s2 c3 s3
csq = { cos(q1) sin(q1) cos(q2) sin(q2) cos(q3) sin(q3) };
csi = { c1 s1 c2 s2 c3 s3 };

subs(r01,csq,csi)
subs(r02,csq,csi)
subs(r03,csq,csi)

```

```
Symbolic pkg v3.1.1: Python communication link active, SymPy v1.12.  
ans = (sym 4x1 matrix)
```

$$\begin{bmatrix} -c_1 \cdot y_1 \\ -s_1 \cdot y_1 \\ d_1 + z_1 \\ 1 \end{bmatrix}$$

```
ans = (sym 4x1 matrix)
```

$$\begin{bmatrix} c_1 \cdot d_2 + c_1 \cdot z_2 \\ d_2 \cdot s_1 + s_1 \cdot z_2 \\ d_1 \\ 1 \end{bmatrix}$$

```
ans = (sym 4x1 matrix)
```

$$\begin{bmatrix} c_1 \cdot d_2 + x_3 \cdot (c_1 \cdot c_3 - s_1 \cdot s_2 \cdot s_3) \\ d_2 \cdot s_1 + x_3 \cdot (c_1 \cdot s_2 \cdot s_3 + c_3 \cdot s_1) \\ -c_2 \cdot s_3 \cdot x_3 + d_1 \\ 1 \end{bmatrix}$$

```
In [24]:  
assert_v( "s:r01", subs(r01,csq,csi) )  
assert_v( "s:r02", subs(r02,csq,csi) )  
assert_v( "s:r03", subs(r03,csq,csi) )
```

A súlyponti helyvektorok tehát a globális KR-ben:

$${}^0\mathbf{r}_1 = {}^0\mathbf{T}_1^{-1}\mathbf{r}_1 = \begin{bmatrix} -c_1 y_1 \\ -s_1 y_1 \\ d_1 + z_1 \\ 1 \end{bmatrix},$$

$${}^0\mathbf{r}_2 = {}^0\mathbf{T}_1^{-1}{}^1\mathbf{T}_2^{-2}\mathbf{r}_2 = \begin{bmatrix} c_1 d_2 + c_1 z_2 \\ d_2 s_1 + s_1 z_2 \\ d_1 \\ 1 \end{bmatrix}$$

$${}^0\mathbf{r}_3 = {}^0\mathbf{T}_1^{-1}{}^1\mathbf{T}_2^{-2}{}^2\mathbf{T}_3^{-3}\mathbf{r}_3 = \begin{bmatrix} c_1 d_2 + x_3(c_1 c_3 - s_1 s_2 s_3) \\ d_2 s_1 + x_3(c_1 s_2 s_3 + c_3 s_1) \\ -c_2 s_3 x_3 + d_1 \\ 1 \end{bmatrix}$$

# Jacobi-mátrixok

## Sebességvektorok

A súlyponti sebességek illetve a megfelelő Jacobi-mátrixok:

$${}^0\mathbf{v}_i = {}^0\dot{\mathbf{r}}_i = {}^0\mathbf{J}_{v_i}\dot{\mathbf{q}}, \quad (i = 1, 2, 3)$$

ahol

$${}^0\mathbf{J}_{v_i} = \frac{d^0\mathbf{r}_i}{d\mathbf{q}}$$

```
In [25]: function J = jacobian(v,q)
%
% v: SYMBOLIC VECTOR
% q: CELL ARRAY OF SYMBOLIC VARS
%
    assert( columns(v(:))==1, "1st argument is not a vector!" )
    J = v(:)*(1:length(q));
    for i=1:length(v)
        J(i,:) = gradient( v(i), q );
    end;
end;

Jv1 = jacobian(r01(1:3),[q1;q2;q3]);
Jv2 = jacobian(r02(1:3),[q1;q2;q3]);
Jv3 = jacobian(r03(1:3),[q1;q2;q3]);

subs(Jv2,csq,csi)
subs(Jv3,csq,csi)
```

ans = (sym 3x3 matrix)

$$\begin{bmatrix} -d_2 \cdot s_1 - s_1 \cdot z_2 & 0 & 0 \\ c_1 \cdot d_2 + c_1 \cdot z_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

ans = (sym 3x3 matrix)

$$\begin{bmatrix} -d_2 \cdot s_1 + x_3 \cdot (-c_1 \cdot s_2 \cdot s_3 - c_3 \cdot s_1) & -c_2 \cdot s_1 \cdot s_3 \cdot x_3 & x_3 \cdot (-c_1 \cdot s_3 - c_3 \cdot s_1 \cdot s_2) \\ c_1 \cdot d_2 + x_3 \cdot (c_1 \cdot c_3 - s_1 \cdot s_2 \cdot s_3) & c_1 \cdot c_2 \cdot s_3 \cdot x_3 & x_3 \cdot (c_1 \cdot c_3 \cdot s_2 - s_1 \cdot s_3) \\ 0 & s_2 \cdot s_3 \cdot x_3 & -c_2 \cdot c_3 \cdot x_3 \end{bmatrix}$$

```
In [26]: assert_v( "s:Jv1", subs(Jv1,csq,csi) )
assert_v( "s:Jv2", subs(Jv2,csq,csi) )
assert_v( "s:Jv3", subs(Jv3,csq,csi) )
```

## Szögsebesség vektorok

Az egyes tagok (relatív) szögsebessége az adott tag KR-ében:

$${}^i\omega_{ji} = \sum_{j=1}^i {}^i\mathbf{R}_j \dot{q}_j \quad , \quad i = (1, 2, 3)$$

Az  $m_1$ ,  $m_2$  és  $m_3$  tömegű részek abszolút szögsebességei a *lokális KR-ekben*:

$${}^1\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \omega_1 \end{bmatrix}, \quad {}^2\omega_2 = \begin{bmatrix} c_2\omega_1 \\ -\omega_1 s_2 \\ \omega_2 \end{bmatrix}, \quad {}^3\omega_3 = \begin{bmatrix} -c_2\omega_1 s_3 + c_3\omega_2 \\ -c_2 c_3 \omega_1 - \omega_2 s_3 \\ \omega_1 s_2 + \omega_3 \end{bmatrix}$$

és a megfelelő Jacobi-mátrixok:

$${}^1\mathbf{J}_{\omega_1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad {}^2\mathbf{J}_{\omega_2} = \begin{bmatrix} c_2 & 0 & 0 \\ -s_2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad {}^3\mathbf{J}_{\omega_3} = \begin{bmatrix} -c_2 s_3 & c_3 & 0 \\ -c_2 c_3 & -s_3 & 0 \\ s_2 & 0 & 1 \end{bmatrix}.$$

(az alábbi cellában elvégzett szimbolikus számítások alapján)

```
In [27]: % AZ EGYES TAGOK RELATÍV SZÖGSEBESSÉGEI (\dot{q}_i)
om1 = sym('omega1');
om2 = sym('omega2');
om3 = sym('omega3');

% AZ EGYES KR-ek KÖZÖTTI FORGATÁSI MÁTRIXOK:
R01 = Ti(1,q1)(1:3,1:3);
R21 = transpose( Ti(2,q2)(1:3,1:3) );
R32 = transpose( Ti(3,q3)(1:3,1:3) );

% AZ ABSZOLÚT SZÖGSEBESSÉGVETGOROK A LOKÁLIS KR-BEN:
w1 = [ 0; 0; om1 ]
w2 = R21*w1+[ 0; 0; om2 ];
w3 = R32*w2+[ 0; 0; om3 ];

subs(w2,csq,csi)
subs(w3,csq,csi)
```

```
w1 = (sym 3x1 matrix)
```

$$\begin{bmatrix} 0 \\ 0 \\ \omega_1 \end{bmatrix}$$

```
ans = (sym 3x1 matrix)
```

$$\begin{bmatrix} c_2 \cdot \omega_1 \\ -\omega_1 \cdot s_2 \\ \omega_2 \end{bmatrix}$$

```
ans = (sym 3x1 matrix)
```

$$\begin{bmatrix} -c_2 \cdot \omega_1 \cdot s_3 + c_3 \cdot \omega_2 \\ -c_2 \cdot c_3 \cdot \omega_1 - \omega_2 \cdot s_3 \\ \omega_1 \cdot s_2 + \omega_3 \end{bmatrix}$$

```
In [28]: assert_v( "s:w1", subs(w1,csq,csi) )
assert_v( "s:w2", subs(w2,csq,csi) )
assert_v( "s:w3", subs(w3,csq,csi) )
```

```
In [29]: % A SZÖGSEBESSÉGEK JACOBI MÁTRIXAI A LOKÁLIS KR-EKBEN:
Jw1 = jacobian(w1,[om1;om2;om3]);
Jw2 = jacobian(w2,[om1;om2;om3]);
Jw3 = jacobian(w3,[om1;om2;om3]);

subs(Jw1,csq,csi)
subs(Jw2,csq,csi)
subs(Jw3,csq,csi)
```

```
ans = (sym 3x3 matrix)
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

```
ans = (sym 3x3 matrix)
```

$$\begin{bmatrix} c_2 & 0 & 0 \\ -s_2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

```
ans = (sym 3x3 matrix)
```

$$\begin{bmatrix} -c_2 \cdot s_3 & c_3 & 0 \\ -c_2 \cdot c_3 & -s_3 & 0 \\ s_2 & 0 & 1 \end{bmatrix}$$

```
In [30]: assert_v( "s:Jw1", subs(Jw1,csq,csi) )
assert_v( "s:Jw2", subs(Jw2,csq,csi) )
assert_v( "s:Jw3", subs(Jw3,csq,csi) )
```

## Mozgás egyenletek

A robot mozgás egyenletei a holonóm kényszerek miatt a másodfajú Lagrange-egyenletekből származtathatók,

$$\frac{d}{dt} \frac{\partial \mathcal{T}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{T}}{\partial \mathbf{q}} + \frac{\partial U}{\partial \mathbf{q}} = Q$$

amihez tehát szükség van a mozgási és potenciális energia kifejezéseire.

### Mozgási energia

Az  $m_1$ ,  $m_2$  és  $m_3$  mozgó részekből álló robot mozgási energiája:

$$\mathcal{T} = \frac{1}{2} \sum_{i=1}^3 \left( m_i v_i^2 + \omega_i^T \mathbf{R}_i^i \boldsymbol{\Theta}_i^0 \mathbf{R}_i^\top \omega_i \right) = \frac{1}{2} \dot{\mathbf{q}}^\top (\mathbf{H}_{v_1} + \mathbf{H}_{v_2} + \mathbf{H}_{v_3} + \mathbf{H}_{\omega_1} + \mathbf{H}_{\omega_2} + \mathbf{H}_{\omega_3}) \dot{\mathbf{q}}$$

ahol

$$\mathbf{H}_{v_i} = m_i^0 \mathbf{J}_{v_i}^\top \mathbf{J}_{v_i} \quad ; \quad \mathbf{H}_{\omega_i} = {}^i \mathbf{J}_{\omega_i}^\top \boldsymbol{\Theta}_i^i \mathbf{J}_{\omega_i}$$

és (számításokat l. később)

$$\mathbf{H}_{v_1} = \begin{bmatrix} c_1^2 m_1 y_1^2 + m_1 s_1^2 y_1^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{H}_{v_2} = \begin{bmatrix} m_2(d_2 + z_2)^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{H}_{v_3} = \begin{bmatrix} m_3(2c_3 d_2 x_3 + d_2^2 + s_2^2 s_3^2 x_3^2 - s_3^2 x_3^2 + x_3)^2 & c_2 m_3 s_3 x_3(c_3 x_3 + d_2) & m_3 s_2 x_3(c_3 d_2 + x_3) \\ c_2 m_3 s_3 x_3(c_3 x_3 + d_2) & m_3 s_3^2 x_3^2 & 0 \\ m_3 s_2 x_3(c_3 d_2 + x_3) & 0 & m_3 x_3^2 \end{bmatrix}$$

de  $\mathbf{H}_{\omega_i}$  előállításához meg kell határozni az  $m_1$  és  $m_2$  tömegű részek tehetetlenségi nyomatéki mátrixait is.

Az  $m_1$ ,  $m_2$  és  $m_3$  tömegű részek tehetetlenségi nyomatéki mátrixai a rögzített tagokból ( ${}^i\hat{\Theta}_i$ ,  $i = 1, \dots, 5$ ) rakható össze:

$${}^1\Theta_1 = \sum_{i=1}^3 \left( {}^1\mathbf{R}_i {}^i\hat{\Theta}_i {}^i\mathbf{R}_1 + \hat{m}_i \mathbf{D}({}^1\Delta\hat{\mathbf{r}}_i) \right), \quad {}^2\Theta_2 = {}^2\hat{\Theta}_4 + \hat{m}_4 \mathbf{D}({}^2\mathbf{r}_4 - {}^2\hat{\mathbf{r}}_4), \quad {}^3\Theta_3 = {}^3\Theta_5$$

és

$$\mathbf{D}(\mathbf{r}) = \mathbf{r}^2 \mathbf{I} - \mathbf{r} \mathbf{r}^\top, \quad {}^1\Delta\hat{\mathbf{r}}_i = {}^1\mathbf{r}_1 - {}^1\mathbf{T}_i {}^i\hat{\mathbf{r}}_i, \quad (i = 1, \dots, 4).$$

In [31]: % A FENTI ILLETVE KORÁBBI ÖSSZEFÜGGÉSEK ALAPJÁN  $m_1$ ,  $m_2$ ,  $m_3$  ÉS

% TEHETETLENSÉGI NYOMATÉKI MÁTRIXAIK NUMERIKUS ÉRTÉKEI

%

**global** Mass1 Mass2 Mass3 Theta1 Theta2 Theta3

% YOUR CODE HERE

```
Mass1 = Mass(1)+Mass(2)+Mass(3);
Mass2 = Mass(4);
Mass3 = Mass(5);
```

[ Mass1 Mass2 Mass3 ]

```
Dr1 = dot(rSP(1)-CoG(1),rSP(1)-CoG(1))*eye(3)-(rSP(1)-CoG(1))*transpose(rSP(1)-CoG(1));
Dr2 = dot(rSP(1)-CoG(2),rSP(1)-CoG(2))*eye(3)-(rSP(1)-CoG(2))*transpose(rSP(1)-CoG(2));
Dr3 = dot(rSP(1)-CoG(3),rSP(1)-CoG(3))*eye(3)-(rSP(1)-CoG(3))*transpose(rSP(1)-CoG(3));
```

```
Theta1 = (Theta(1)+Mass(1)*Dr1)+(Theta(2)+Mass(2)*Dr2)+(Theta(3)+Mass(3)*Dr3);
```

```
Theta2 = Theta(4);
```

```
Theta3 = Theta(5);
```

ans =

20.3268 5.1710 1.2065

In [32]: assert\_v( "%4g:[Mass1,Mass2,Mass3]", [Mass1,Mass2,Mass3] )

assert\_v( "Theta1", Theta1 )

assert\_v( "Theta2", Theta2 )

assert\_v( "Theta3", Theta3 )

In [33]: % TÖMEGEK SZIMBOLIKUS VÁLTOZÓI

syms m1 m2 m3

% TEHETETLENSÉGI MÁTRIX ELEMÉK

syms X1 Y1 Z1 Dxy1 Dxz1 Dyz1

syms X2 Y2 Z2 Dxy2 Dxz2 Dyz2

syms X3 Y3 Z3 Dxy3 Dxz3 Dyz3

% A Th1, Th2, Th3 TEHETETLENSÉGI MÁTRIXOK ELEMÉKET HELYETTESÍTSE 0-val:

```
% TEHETETLENSÉGI MÁTRIXOK
Th1 = [ X1 0 0; 0 Y1 -Dyz1; 0 -Dyz1 Z1 ]
Th2 = [ X2 0 0; 0 Y2 0; 0 0 Z2 ]
Th3 = [ X3 0 0; 0 Y3 0; 0 0 Z3 ]
```

Th1 = (sym 3x3 matrix)

$$\begin{bmatrix} x_1 & 0 & 0 \\ 0 & y_1 & -Dyz_1 \\ 0 & -Dyz_1 & z_1 \end{bmatrix}$$

Th2 = (sym 3x3 matrix)

$$\begin{bmatrix} x_2 & 0 & 0 \\ 0 & y_2 & 0 \\ 0 & 0 & z_2 \end{bmatrix}$$

Th3 = (sym 3x3 matrix)

$$\begin{bmatrix} x_3 & 0 & 0 \\ 0 & y_3 & 0 \\ 0 & 0 & z_3 \end{bmatrix}$$

```
In [34]: assert_v( "s:Theta1", Th1 )
assert_v( "s:Theta2", Th2 )
assert_v( "s:Theta3", Th3 )
```

```
% A SEBESSÉGEKBŐL SZÁRMAZÓ TÖMEGMÁTRIXOK SZÁMÍTÁSA
% (HASZNÁLJUK A simplify() FV-t A KIFEJEZÉS EGYSZERŰSÍTÉSÉRE!)
Jv1 = jacobian(r01(1:3),[q1;q2;q3]);
Jv2 = jacobian(r02(1:3),[q1;q2;q3]);
Jv3 = jacobian(r03(1:3),[q1;q2;q3]);

Hv1 = m1*transpose(Jv1)*Jv1;
Hv2 = simplify(m2*transpose(Jv2)*Jv2);
Hv3 = simplify(simplify(m3*transpose(Jv3)*Jv3));

subs( Hv1,csq,csi );
subs( Hv2,csq,csi );
subs( Hv3,csq,csi );
```

```
In [36]: assert_v( "s:Hv1", subs( Hv1,csq,csi ) )
assert_v( "s:Hv2", subs( Hv2,csq,csi ) )
assert_v( "s:Hv3", subs( Hv3,csq,csi ) )
```

```
% A SZÖGSEBESSÉGEKBŐL SZÁRMAZÓ TÖMEGMÁTRIXOK SZÁMÍTÁSA
% (ITT A simplify NEM BIZTOS, HOGY SEGÍT...)
Jw1 = jacobian(w1,[om1;om2;om3]);
Jw2 = jacobian(w2,[om1;om2;om3]);
Jw3 = jacobian(w3,[om1;om2;om3]);
```

```

Hw1 = transpose(Jw1)*Th1*Jw1;
Hw2 = transpose(Jw2)*Th2*Jw2;
Hw3 = transpose(Jw3)*Th3*Jw3;

% HASZNOS HELYETTESÍTÉS: SX3 = (X3+Y3)/2; DX3 = (X3-Y3)/2
syms SX3 DX3
Hw3 = expand( subs( Hw3, {X3,Y3,cos(q3)^2,sin(q3)^2}, {SX3+DX3,SX3-DX3,(1+cos(2*q3))/2,(1-cos(2*q3))/2} ) );
subs( Hw1,csq,csi );
subs( Hw2,csq,csi );
subs( Hw3,csq,csi );

```

In [38]:

```

assert_v( "s:Hw1", subs( Hw1,csq,csi ) )
assert_v( "s:Hw2", subs( Hw2,csq,csi ) )
assert_v( "s:Hw3", subs( Hw3,csq,csi ) )

```

In [39]:

```

% AZ EREDŐ TÖMEGMÁTRIX:

H = Hv1+Hv2+Hv3+Hw1+Hw2+Hw3;

% ÉS A MOZGÁSI ENERGIA
dqi = [ om1; om2; om3 ];

Ekin = sym(1)/2*transpose(dqi)*H*dqi;

subs( Ekin, csq, csi );

```

## A potenciális energia

A nehézségi erő hatását figyelembe véve:

$$\mathcal{U} = m_1g(d_1 + z_1) + m_2gd_1 + m_3g(d_1 - x_3\sin(q_3)\cos(q_2))$$

In [40]:

```

% POTENCIÁLIS ENERGIA SZIMBOLIKUS KIFEJEZÉSE

syms g % 9,81

z1=(r01(3))
z2=(r02(3))
z3=(r03(3))
Upot = m1*g*z1+m2*g*z2+m3*g*z3;
% YOUR CODE HERE

subs(Upot,csq,csi)

% ÉS AZ EBBÓL SZÁRMAZÓ ÁLT. ERŐ OSZLOPVEKTORA:
Qpot = -gradient(Upot,[q1;q2;q3]);

subs( Qpot, csq, csi )

```

```

z1 = (sym) d1 + z1
z2 = (sym) d1
z3 = (sym) d1 - x3·sin(q3)·cos(q2)
ans = (sym) d1·g·m2 + g·m1·(d1 + z1) + g·m3·(-c2·s3·x3 + d1)
ans = (sym 3x1 matrix)

```

$$\begin{bmatrix} 0 \\ -g \cdot m_3 \cdot s_2 \cdot s_3 \cdot x_3 \\ c_2 \cdot c_3 \cdot g \cdot m_3 \cdot x_3 \end{bmatrix}$$

```
In [41]: assert_v( "s:Upot", subs(Upot,csq,csi) )
assert_v( "s:Qpot", subs(Qpot,csq,csi) )
```

## Az aktuátorok virtuális teljesítménye

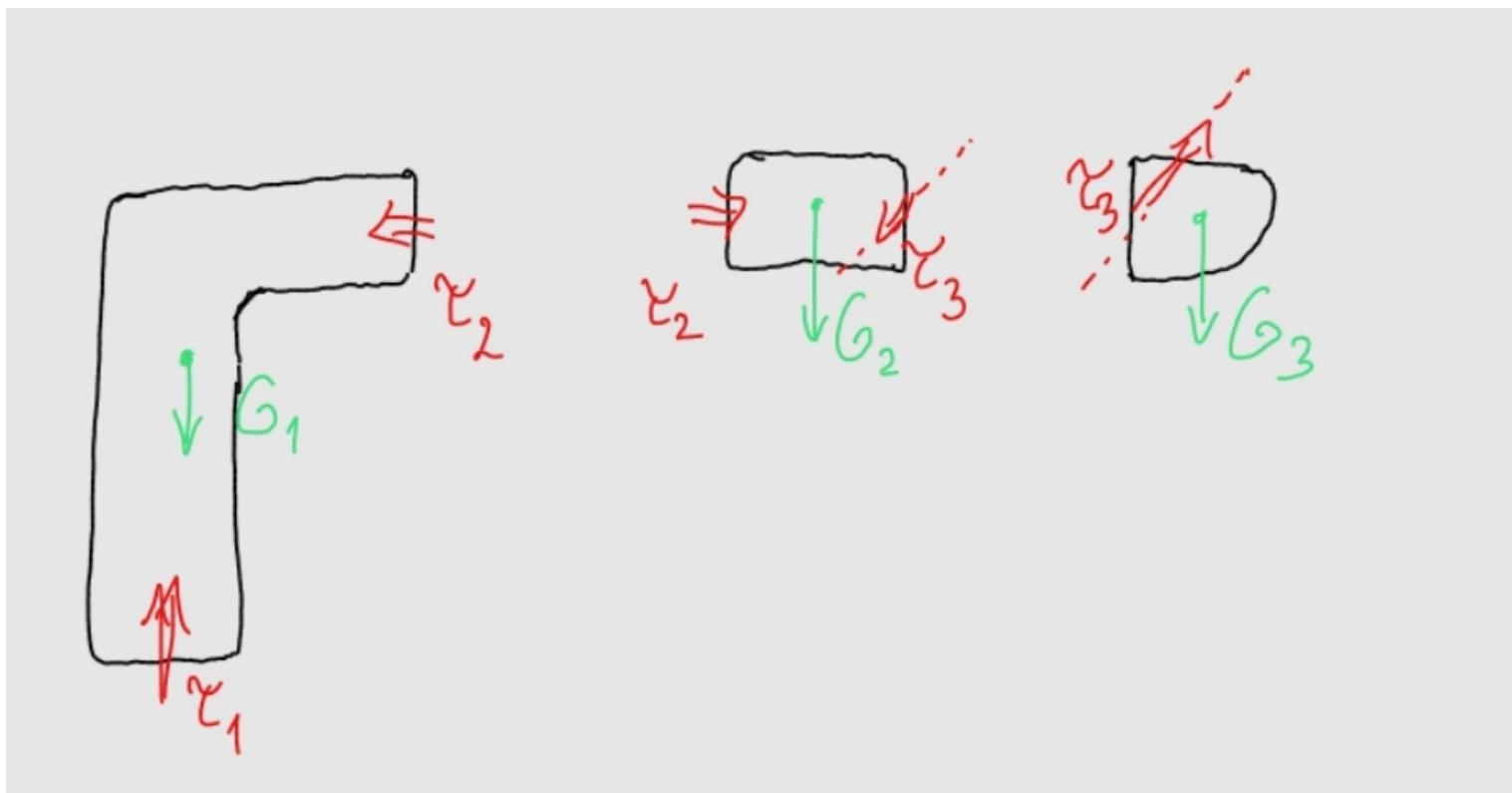
Az 1-es, 4-es és 5-ös csuklókban működő  $\tau_1$  és  $\tau_4$  és  $\tau_5$  aktuátornyomatékok mozgó tagokra gyakorolt hatásainak (szabad test) ábrája:

Ezek alapján a virtuális teljesítményük:

$$\begin{aligned} \delta\mathcal{P} &= {}^1\boldsymbol{\tau}_1 \cdot {}^1\delta\boldsymbol{\omega}_1 + {}^2\boldsymbol{\tau}_4 \cdot {}^2\delta\boldsymbol{\omega}_2 + {}^3\boldsymbol{\tau}_5 \cdot {}^3\delta\boldsymbol{\omega}_3 \\ &= \sum_{k=1,2} Q_k \delta \dot{q}_k \end{aligned}$$

az általános erőkomponensek pedig:

$$\mathbf{Q}(\boldsymbol{\tau}) = \begin{bmatrix} \tau_1 \\ \tau_4 \\ \tau_5 \end{bmatrix}$$



## A származtatott mozgásegyenletek

A

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{h}(\dot{\mathbf{q}}, \mathbf{q}) = \mathbf{Q}_{pot} + \mathbf{Q}(\tau)$$

alakú mozgásegyenlet tehetslenségi erőkből származó nemlineáris vektora és a tömegmátrix közötti kapcsolat

$$\mathbf{h}(\dot{\mathbf{q}}, \mathbf{q}) = \dot{\mathbf{H}}\dot{\mathbf{q}} - \frac{\partial \mathcal{T}}{\partial \mathbf{q}},$$

a potenciális általános erő vektora pedig a potenciálfüggvénnyel kifejezve:

$$\mathbf{Q}_{pot} = -\frac{\partial \mathcal{U}}{\partial \mathbf{q}}$$

```
In [42]: % A NEMLINEÁRIS TEHETETLENSÉGI ERŐK VECTORA:
qv = [q1; q2; q3];
```

```
Hdq=H*dqi;
dH=jacobian(Hdq,qv);
h = dH*dqi-sym(1)/2*transpose(dH)*dqi;
% YOUR CODE HERE
```

In [43]:

```
% A h VEKTOR ELEMEINEK KIÍRÁTÁSA ÁTTEKINTETŐ ALAKRA EGYSZERŰSÍTVE
%
% HASZNOS HELYETTESÍTÉSEK A simplify TÁMOGATÁSÁRA
cs32 = { s2^2, c3^2, s3^2, SX3 };
c2q3 = { 1-c2^2, (1+cos(2*q3))/2, (1-cos(2*q3))/2, Y3+DX3 };
%SD2XY= { (X3+Y3)/2, (X3-Y3)/2 };

% AZ ELSŐ ELEMBEN A SZÖGSEBESSÉGEK EGYÜTTTHATÓINAK LEVÁLOGATÁSA
[c,ct] = coeffs( expand( subs(h(1),csq,csi) ), {om1,om2,om3} );
% ÉS VISSZASZORZÁSA
h_1 = subs( simplify( subs(c,cs32,c2q3) )*transpose(ct), csq, csi )

% A MÁSODIK ELEMBEN A SZÖGSEBESSÉGEK EGYÜTTTHATÓINAK LEVÁLOGATÁSA
[c,ct] = coeffs( expand( subs(h(2),csq,csi) ), {om1,om2,om3} );
% ÉS VISSZASZORZÁSA
h_2 = subs( simplify( subs(c,cs32,c2q3) )*transpose(ct), csq, csi )

% A HARMADIK ELEMBEN A SZÖGSEBESSÉGEK EGYÜTTTHATÓINAK LEVÁLOGATÁSA
[c,ct] = coeffs( expand( subs(h(3),csq,csi) ), {om1,om2,om3} );
% ÉS VISSZASZORZÁSA
h_3 = subs( simplify( subs(c,cs32,c2q3) )*transpose(ct), csq, csi )
```

$$h_1 = \text{sym} \left( \frac{\left( \begin{array}{l} 2 \\ -2\cdot DX_3 \cdot S_3 - X_2 + Y_2 - Y_3 + Z_3 + m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2}{2 \cdot c_2 \cdot w_1 \cdot w_2 \cdot S_2} + c_2 \cdot w_2 \cdot w_3 \cdot \left( \begin{array}{l} -4 \cdot D \\ X_3 \cdot c_3 + 2 \cdot DX_3 + Z_3 + 2 \cdot c_3 \cdot m_3 \cdot X_3 + 2 \cdot c_3 \cdot d_2 \cdot m_3 \cdot X_3 \end{array} \right)^2 - d_2 \cdot m_3 \cdot w_3 \cdot S_2 \cdot S_3 \cdot X_3 + w_1 \cdot w_3 \cdot \left( \begin{array}{l} 2 \\ -2 \cdot DX_3 \cdot c_2 \cdot \sin(2 \cdot q_3) - 2 \cdot c_2 \cdot c_3 \cdot m_3 \cdot S_3 \cdot X_3 - 2 \cdot d_2 \cdot m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2 + w_2 \cdot S_2 \cdot S_3 \cdot \left( \begin{array}{l} 2 \\ -2 \cdot D \\ X_3 \cdot c_3 - c_3 \cdot m_3 \cdot X_3 - d_2 \cdot m_3 \cdot X_3 \end{array} \right)^2 \right)$$

$$h_2 = \text{sym} \left( \frac{\left( \begin{array}{l} 2 \\ -2 \cdot DX_3 \cdot S_3 + X_2 - Y_2 + Y_3 - Z_3 - m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2}{c_2 \cdot w_1 \cdot S_2} + c_2 \cdot w_1 \cdot w_3 \cdot \left( \begin{array}{l} 4 \cdot DX_3 \cdot S_3 \\ -2 \cdot DX_3 - Z_3 - 2 \cdot m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2 + w_2 \cdot w_3 \cdot \left( \begin{array}{l} 2 \\ -2 \cdot DX_3 \cdot \sin(2 \cdot q_3) + 2 \cdot c_3 \cdot m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2 \right)$$

$$h_3 = \text{sym} \left( \frac{\left( \begin{array}{l} 2 \\ -4 \cdot DX_3 \cdot S_3 + 2 \cdot DX_3 + Z_3 + 2 \cdot m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2}{c_2 \cdot w_1 \cdot w_2 \cdot \left( \begin{array}{l} 2 \\ -4 \cdot DX_3 \cdot S_3 + 2 \cdot DX_3 + Z_3 + 2 \cdot m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2 + c_2 \cdot c_3 \cdot m_3 \cdot S_3 \cdot X_3 + d_2 \cdot m_3 \cdot S_3 \cdot X_3} + w_1 \cdot \left( \begin{array}{l} 2 \\ -DX_3 \cdot c_2 \cdot \sin(2 \cdot q_3) \end{array} \right)^2 + w_2 \cdot \left( \begin{array}{l} 2 \\ -DX_3 \cdot \sin(2 \cdot q_3) - c_3 \cdot m_3 \cdot S_3 \cdot X_3 \end{array} \right)^2 \right)$$

```
In [44]: assert_v( "s:h_1", h_1 )
assert_v( "s:h_2", h_2 )
assert_v( "s:h_3", h_3 )
```

## A nemlineáris rész kifejezésének egyszerűsítése

(a további cellák kitöltése nem kötelező, a *Validate* ezzel kapcsolatos figyelmeztetése figyelmen kívül hagyható)

(A  $\mathbf{h}$  vektor alábbi kifejezését a ...-ok helyén egészítsük ki a  $\dot{q}_k$ -k megfelelő kvadratikus kifejezésével, majd adjuk meg a  $H_i$  paraméterek numerikus értékét a következő cellában elvégzett számítások alapján.)

$$\mathbf{h} = \begin{bmatrix} 2(H_{24} + H_{14} \cos(2q_3))s_2c_2 \\ 0 \\ (H_{34} + 2H_{14} \cos(2q_3))c_2 \end{bmatrix} \dots + \begin{bmatrix} (H_{34} - 2H_{14} \cos(2q_3) + 2H_5c_3)c_2 \\ -2H_{14} \sin(2q_3) \\ 0 \end{bmatrix} \dots \\ - \begin{bmatrix} 0 \\ (H_{24} + H_{14} \cos(2q_3))s_2c_2 \\ 2H_{14}c_2^2s_3c_3 - H_5s_3 \end{bmatrix} \dots + \begin{bmatrix} 2H_{14}c_2^2 \sin(2q_3) - 2H_5s_3 \\ (H_{34} + 2H_{14} \cos(2q_3))c_2 \\ 0 \end{bmatrix} \dots + \begin{bmatrix} -H_5s_2s_3 \dots \\ 0 \\ 2H_{14}s_3c_3 \dots \end{bmatrix}$$

ahol

$$H_{14} = \text{YOUR ANSWER HERE}, \dots$$

```
In [45]: % HELYETTESÍTENDŐ SZIMBOLIKUS PARAMÉTEREK cella tömbje:
```

```
pars = { X2, Y2, ...
% YOUR CODE HERE
error('No Answer Given!')
};

D_H= DH_tab([0,0,0]);
% A FENTI PARAMÉTEREK NUMERIKUS ÉRTÉKEINEK vektora MEGFELELŐ SORRENDBEN:
vals = [ Theta2(1,1), Theta2(2,2), ...
% YOUR CODE HERE
error('No Answer Given!')
];

function v = double2sym( mx )
% LEBEGŐPONTOS ÉRTÉKEK MÁTRIXÁBÓL SZIMBOLIKUST CSINÁL
% WARNING NÉLKÜL...
v = sym( zeros(size(mx)) );
for i=1:rows(mx)
    for j=1:columns(mx)
        v(i,j) = sym(mx(i,j), 'r');
    end;
end;
vals = double2sym(vals);

% A H_i ÉRTÉKEK SZÁMÍTÁSA A SZIMBOLIKUS FORMULÁK HELYETTESÍTÉSÉVEL
H14 = eval( subs( ... % H2, H3, ...
% YOUR CODE HERE
error('No Answer Given!')
) )
```

```
error: No Answer Given!
error: No Answer Given!
error: 'vals' undefined near line 1, column 1
error: No Answer Given!
```

```
In [46]: assert_v( "H14", H14 )
assert_v( "H24", H24 )
assert_v( "H34", H34 )
assert_v( "H5", H5 )
```

```
error: 'H14' undefined near line 1, column 1
error: 'H24' undefined near line 1, column 1
error: 'H34' undefined near line 1, column 1
error: 'H5' undefined near line 1, column 1
```