Robotok direkt geometriája

1. A gyakorlat célja

Direkt geometriai feladatot megvalósító osztály implementálása. A megvalósított függvénycsomag tesztelése egy Stanford kar végberendezése pozíciójának meghatározásához.

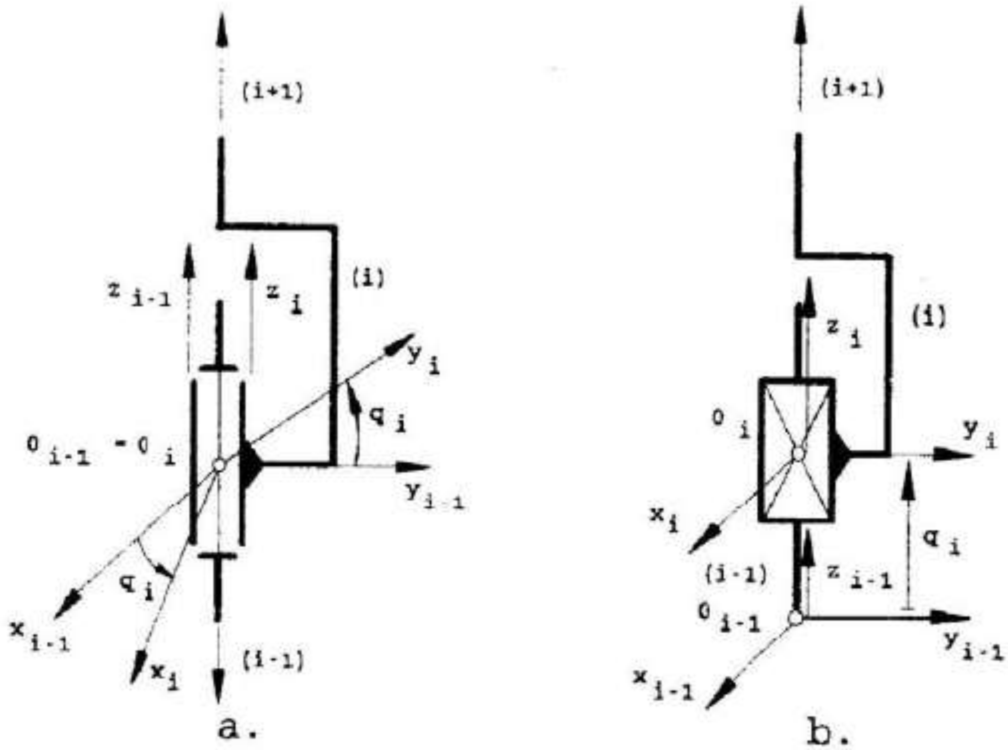
2. Elméleti bevezető

2.1 Robotok leírása

A soros robotok geometriai szempontból nyitott kinematikai láncok. A mechanikai struktúra nem más, mint egy *n+1* elemből álló mechanikai rendszer, ahol (0)-val jelölik a robot bázisát, melyhez a világkoordináta-rendszer vagy a bázisrendszer van rögzítve, melynek jelölése $O_0X_0Y_0Z_0$. A többi *n* elem képezi a robot mozgatható részét, és külön-külön ezek a struktúra kinematikai elemeinek ill. a robot karjainak is nevezendők. Jelölésük - a bázishoz hasonlóan - (i), ahol i=l→n. Minden elemhez egy saját koordináta rendszer csatolódik, $\{i\}=O_iX_iY_iZ_i$. Ezeket, a robot mozgó vagy mobil vonatkoztatási rendszereinek is nevezzük, és ezek segítségével alkothatjuk meg a robotok geometriai, kinematikai és dinamikai modellezését.

Az említett karok csuklókkal vannak összekötve, melyekhez minden esetben hajtás van társítva. A további vizsgálatok leegyszerűsítése végett a csuklók mechanikai szempontból ideális csuklóknak lesznek tekintve:

- nincs alakváltozás a csuklóban a belső érintkezésekből adódóan (merev csukló);
- véges és végtelenül kis mozgások esetén is a szabadságfokok száma megegyezik (nincs játék a csuklóban).



1. Ábra. Az R és a T típusú csuklók vázlatos ábrázolása

Ezeket a csuklókat (R és T) vázlatosan az 1. ábra szerint tudjuk jelölni. A csukló az alsó végével az {i-1} elemhez kapcsolódik, illetve tekinthető mint az {i}-edik karhoz tartozó csukló, így ehhez a csuklóhoz kapcsolható a kar vonatkoztatási rendszere. Az ábrán fel van tüntetve a q_i csukló változó is, mely természetesen csuklóhoz és karhoz kötött az i indexen keresztül. A q_i határozza meg, hogy mekkora az {i-1} és az {i} karok közötti relatív elmozdulás, és a változó dimenziója lehet szög, illetve hossz mértékegység annak függvényében, hogy R vagy T típusú csuklóról van szó. A fentieket figyelembe véve kijelenthető, hogy ha egy robotnak *n* karja van, akkor mindegyikhez hozzárendelhető egy csukló változó, melyeket összesítve robotparamétereknek is nevezhetünk. A q_i változókat egy mátrixba is foglalhatjuk, mely a robotparaméterek vektorjaként ismeretes:

$$\underset{(nx1)}{\overline{\theta}} = [q_i, i = 1 \rightarrow n]^T \qquad vagy \qquad \underset{(nx1)}{\overline{\theta}} = \begin{bmatrix} q_1 \\ \vdots \\ q_i \\ \vdots \\ q_n \end{bmatrix} \tag{1}$$

Az így meghatározott vektor a robot egy konfigurációját képviseli, a csuklóváltozókat tömöríti. Ha ezek a változók értékei zérus elemek, akkor az (1) egyenlet sajátos alakuvá válik - a robot kezdeti, zérus konfigurációját tartalmazza, egy csuklóban sincs elmozdulás:

$$\underset{(nx1)}{\overline{\theta}} = [q_i = 0, i = 1 \to n]^T \qquad vagy \qquad \underset{(nx1)}{\overline{\theta}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2}$$

Az említett konfigurációs vektorral párhuzamosan bevezethető a koordináta oszlopmátrix, mely egy test helyzetét írja le a derékszögű, 3 dimenziós vonatkoztatási rendszerbe:

$$^{0}\underset{(nx1)}{\overline{X}} = [^{0}x_j, j = 1 \to m; m \le n]^T \tag{3}$$

ahol az *m* a vonatkoztatási térben lehetséges illetve kihasznált szabadsági fokok számát képviseli. Fontos megjegyezni, hogy habár a térben hat szabadság foka lehet egy szabad testnek, ha a feladatunk úgy kívánja, akkor *m=4* lesz. A hat lehetséges mozgásból csak 4-et használunk (pl. 3 haladó és 1 forgó mozgást).

Egy robot használata során az elsődleges feladat annak meghatározása, hogy egy bizonyos robot konfigurációhoz milyen karakterisztikus pont koordináták tartoznak, illetve ennek fordítottját is meg kell határozni (főleg vezérlés esetén elengedhetetlen). Egyenletekbe öntve az előbbi kijelentést, mondhatjuk, hogy keressük az $^{0}\overline{X}$ vektort (3), ha ismert a robotparaméterek vektora, vagy ki kell számtanunk a $\overline{\theta}$ oszlopvektort (1), ha adott a mozgatandó test térbeli helyzete. Az első esetben a robot mehankai struktúrájának *direkt geometriai modellezéséről* beszélhetünk, míg a fordított irányú számítást *inverz vagy fordított geometriai modellezésnek* nevezzük.

Egy robot struktúráját három fő részre oszthatjuk:

1.a pályageneráló rész, ami nem más, mint a térbeli pozíciót megadó mechanizmus, ami állhat úgy R mint T csuklókkal összekapcsolt karokból,

2. a test irányítását megadó mechanizmus: mivel ez egy, kettő vagy három tengely körüli forgást igényel, a használt csuklók kizárólag csak R típusúak lehetnek,

3. a szerszámbefogó rész, ami rögzíti a mozgatandó terhet ill. szerszámot.

Mivel ez utóbbinak működése nem befolyásolja a robot és a teher mozgását, csak ezek relativ helyzetét rögzíti, eltekintünk a vizsgálásától. Egy un. karakterisztikus ponttal helyettesítjük, melyet **P**-vel fogunk jelölni, és ezt az *n*.-ik karhoz rözítettnek tekintjük.

Hogy az első két részt matematikailag sikeresen modellezzük, egy általános geometriai mátrixot vezetünk be, mely meghatározza a mechanikai struktúrát a zérus konfigurációban. Ez egy *(n+1)x6*-os mátrix, melynek hat oszlopa és annyi sora van ahány csuklója, ill. az utolsó sor a karakterisztikus pontot határozza meg. Az *i*-edik sor első három eleme a csukló középpontjának helyzetvektor (${}^{0}\overline{p}_i^{(0)}$) komponensein, míg az utolsó három tag a csukló tengelynek (${}^{0}\overline{k}_i^{(0)}$) irányítását határozza meg három iránykoszinusz segítségével. Természetesen ezek a vektorok a világkoordinátarendszerben értendőek, ezért a bal-felső index nullát mutat, illetve a zéros konfigurációt jellemzik (ezért jelenik meg zárójelben a zérus érték):

$$\underset{(n+1)x6}{M_g^{(0)}} = \left[\left[{}^{0}\overline{p}_i^{(0)^T} \quad {}^{0}\overline{k}_i^{(0)^T} \right] i = 1 \rightarrow n+1 \right] \tag{4}$$

A homogén transzformációs mátrix: Matemetikai szempontból egy vektort az un. homogén módon is fel lehet írni. Bevezetjük a $\underset{(4x4)}{[T]}$ jelölést, amit homogén transzformációs mátrixnak fogunk nevezni. Ennek alakja:

$$\underset{(4x4)}{[T]} = \begin{bmatrix} \underset{(3x3)}{[R]} & \underset{(3x1)}{\overline{p}} \\ \underset{(1x3)}{f} & w \end{bmatrix} \tag{5}$$

ahol R egy rotációs mátrix, p helyzetvektor illetve f egy perspektivikus transzformációs vektor, melynek vizsgálatunk során állandó értéket adunk: [*0 0 0*]. A w egy léptéktényező és w=1 értékkel lesz ellátva.
A T_T homogén transzformációs mátrix az elcsúsztatást képviseli:

$$T_T = \underset{(4x4)}{\begin{bmatrix} & I_3 & & {}^{i-1}\overline{r}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{6}$$

A T_R a rotációt megjelenítő homogén transzformációs mátrix:

$$T_R = \underset{(4x4)}{\begin{bmatrix} & {}^{i-1}_{\ \ i}[R] & & \overline{0} \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{7}$$

Tehát a homogén transzformációs mátrix, amely szorzás útján megvalósítja az átmenetet az {i} rendszerből az {i-1} rendszerbe:

$$\underset{(4x4)}{{}^{i-1}_{i}[T]} = T_T \cdot T_R = \underset{(4x4)}{\begin{bmatrix} & I_3 & & {}^{i-1}\overline{r}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}} \cdot \underset{(4x4)}{\begin{bmatrix} {}^{i-1}_{i}[R] & & & \overline{0} \\ 0 & 0 & 0 & 1 \end{bmatrix}} = \underset{(4x4)}{\begin{bmatrix} {}^{i-1}_{i}[R] & & & {}^{i-1}\overline{r}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{8}$$

A következő egyenlet segítségével meg lehet határozni a P karakterisztikus pont helyzetét a világkoordináta rendszerben ismerve az {n} rendszerbeli helyzetét, illetve z ezt követő ennek fordítottjára ad lehetőséget.

$${}^{0}[P] = {}^{0}_{1}[T] \cdot {}^{1}_{2}[T] \cdot \ldots \cdot {}^{i-1}_{i}[T] \cdot \ldots \cdot {}^{n-1}_{n}[T] \cdot {}^{n}[P] = \prod_{i=1}^{n} {}^{i-1}_{i}[T] \cdot {}^{n}[P] \tag{9}$$

$${}^{n}[P] = {}^{n}_{n-1}[T] \cdot \ldots \cdot {}^{i}_{i-1}[T] \cdot \ldots \cdot {}^{2}_{1}[T] \cdot {}^{1}_{0}[T] \cdot {}^{0}[P] = \prod_{i=n}^{1} {}^{i}_{i-1}[T] \cdot {}^{0}[P] \tag{10}$$

2.2. Direkt geometriai feladat

A geometriai modellezés a két vektor közötti összefüggés megállapítását jelenti:

$${}^{0}\overline{X} \Leftrightarrow \overline{\theta} \tag{11}$$

Ennek érvényessége két megszorítás elméleti teljesítését feltételezi:

1. statikus feltétel: a (11) egyenlet független az időtől
2. szilárdsági feltétel: minden kinematikai elem véges számú alkatrészből áll, és ezek relatív helyzete nem változik.

Ezenkívül a csuklók is ideális csuklóknak vannak tekintve.
Két féle modellezésről lehet beszélni

- direkt geometriai modellezés (DGM): ${}^{0}\overline{X} = f\left(\overline{\theta}\right)$
- inverz vagy fordított geometriai modellezés (IGM): $\overline{\theta} = f^{-1}\left({}^{0}\overline{X}\right)$

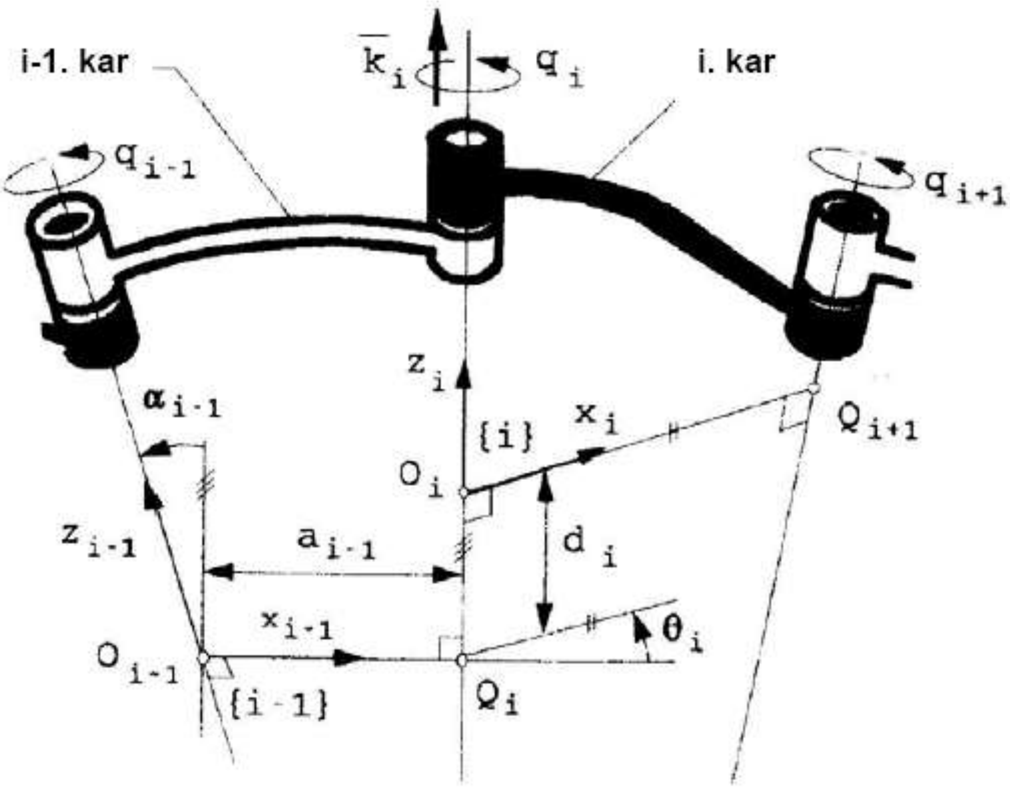
Denavit-Hartemberg paraméterek használata: Az eddigiekben hat paraméter volt használva, hogy leírható legyen egyik kar másikhoz viszonyított helyzete. A Denavit-Hartenberg (DH) paraméterek segítségével ugyanez megvalósítható, viszont ezek száma csak négy.
Legyen a 3. ábrán látható két egymást követő kar, és vizsgálva legyen szokásos módon az i. kar helyzete az i-l. -hez képest. A relatív helyzetet a négy paraméterrel adhatjuk meg:

$$[a_{i-1};\alpha_{i-1};d_i;\theta_i]$$

Mindegyik meghatározása a 3. ábráról olvasható le:

- a_{i-1} – a két tengely ($\bar{k}_{i-1}$ és $\bar{k}_i$) közös merőlegesének hossza,
- α_{i-1} – a két tengely ($\bar{k}_{i-1}$ és $\bar{k}_i$) közötti szög,
- d_i – a $\bar{k}_i$ tengely két közös merölegese közötti szög, a $\bar{k}_i$ tengelyre mérve,
- θ_i - a $\bar{k}_i$ tengely két közös merölegese közötti szög.

Annak függvényében, hogy az i. csukló R vagy T tipusú, a q_i csukló-változó más paramétert kell befolyásoljon. Ha R csukló szerepel a két kar között, akkor q_i (melyet ebben az esetben szög mértékegységben mérünk) a θ_i elemet befolyásolja.



3. Ábra. *Denavit-Haztenberg paraméterek*

Ha tarnszlációs csukló köti össze a két kart, akkor a q_i dimenziója hosszmérték lesz, és természetesen akkor a d_i paramétert befolyásolja. Általánosítva az egész robot struktúrára $i=1\to n$:

$$\begin{aligned}&[a_{i-1};\alpha_{i-1};d_i;\theta_i \pm q_i; \quad i=1\to n], \quad ha\ i=R\\&[a_{i-1};\alpha_{i-1};d_i \pm q_i;\theta_i; \quad i=1\to n], \quad ha\ i=T\end{aligned} \tag{12}$$

Hogy ezeket a paramétereket be tudjuk építeni homogén transzformációs mátrixba, a struktúra koordinátarendszereit kellő képen kell kiválasztani. Az első lépés a rendszerek középpontjainak (O_i) a megválasztása kell legyen:

$$O_i = \begin{cases} \bar{k}_i \cap \bar{k}_{i+1}\,, \quad ha\ \bar{a}_i = \bar{0} \\ C_i\ (tetszőleges\ pont)\,, \quad ha \begin{cases} \bar{a}_i = \bar{0} \\ \bar{k}_i \cdot \bar{k}_{i+1} = \pm 1 \end{cases} \\ \bar{a}_i \cap \bar{k}_i\,, \quad ha\ \bar{a}_i \neq \bar{0} \end{cases} \tag{13}$$

Ezek után a z_i , x_i és y_i tengelyek meghatározása következik:

$$\bar{z}_i = \bar{k}_i \tag{14}$$

$$\bar{x}_i = \begin{cases} \dfrac{\bar{a}_i}{\left|\bar{a}_i\right|}, \quad ha \quad \bar{a}_i \neq \bar{0} \\ \bar{z}_i \times \bar{z}_{i+1}\,, \quad ha \quad \bar{k}_i \cap \bar{k}_{i+1} \end{cases} \tag{15}$$

$$\bar{y}_i = \bar{z}_i \times \bar{x}_i \tag{16}$$

A {0} és az {n} rendszerre a következő megkötések érvényesek:

$$\begin{aligned} &\{0\} \rightarrow O_0 = O_1\left(\in \{1\}\right); \quad \bar{z}_0 = \bar{k}_1 \\ &\{n\} \rightarrow O_n = \bar{x}_{n-1} \cap \bar{k}_n \end{aligned} \tag{17}$$

Ha meghatározzuk a rendszereket a DH paraméterekkel, akkor ezek értékei is kiszámíthatók:

$a_{i-1};\alpha_{i-1}$ – a távolság illetve a szög a $\bar{z}_{i-1} \rightarrow \bar{z}_i$ tengelyek között az $\bar{x}_{i-1}$ mentén

$d_i;\theta_i$ – a távolság illetve a szög az $\bar{x}_{i-1} \rightarrow \bar{x}_i$ tengelyek között a $\bar{z}_{i-1}$ tengely mentén

Meghatározva a paraméterek értékeit, mindegyiket, egyenként egy homogén transzformációs mátrixba lehet foglalni:

$$\begin{aligned} &T_R(\bar{x},\alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_T(\bar{x},a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &T_R(\bar{z},\theta_i) = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_T(\bar{z},d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{18}$$

Az eredő transzformációs mátrix nem lesz más, mint ezek szorzata:

$$ {}^{i-1}_{\;\;i}[T] = T_R(\bar{x};\alpha_{i-1})\cdot T_T(\bar{x};a_{i-1})\cdot T_R(\bar{z};\theta_i)\cdot T_T(\bar{z};d_i) \tag{19} $$

Az (19) egyenlet kidolgozása után sikeresen átalakítható a robot bármely mozgó rendszeréhez tartozó pont helyzete a világkoordináta rendszerbe.

3. A mérés menete

3.1 Az Armadillo könyvtár függvényeinek megismerése

Az Armadillo C++ alatt megírt függvény csomag, amely segítségével C++ környezetben nagyon könnyen lehet mátrixműveleteket végezni. A könyvtár ingyenesen letölthető és könnyen illeszthető C++ alatt fejlesztett tervekhez. A könyvtár bevezet mátrix és vektor típusokat, ezekekkel operátorok használatával lehet műveleteket elvégezni.
Néhány mátrix típusok:

- Általános mátrix: *Mat<t> m(n, m)* – általános mátrix deklarálása, *n* sorok, *m* oszlopok száma, t a típus lehet int, float double, stb.
- *Row<t> r(n)* sorvektor, *n* az elemek száma
- *Col<t> v(n)* oszlopvektor, *n* az elemek száma
- *Eye<t>(n,m)* egységmátrix, nem felülírható típus

Néhány fontosabb művelet:

- értékátadás: *m=m1; m=I*
- mátrix elemeinek megadása: *m(i,j)=12,3; r=m(2,3)*
- mátrixszorzás: *m=m1*m2; v=m*v1;*
- elemek számának kelkérdezése: *int n=size(A);*

Szorzásnál, értékátadásnál kell vigyázni, hogy a mátrixok sorainak és oszlopainak száma a műveletnek megfelelően egyezzenek meg.

3.2 A direkt geometriát megvalósító osztály implementálása

Az osztály a *RobotGeometryProj* tervben található *RobotGeometry.cpp* állományban kell megvalósítani a Newmat osztály segítségével. Az osztály már hozzá van csatolva a tervhez.
A megvalósítandó függvények:

int RobotGeometry::TranslationX(double d, Mat<double> &T)

Eltolás x mentén. A visszatérítendő *T* mátrix egy egységmátrix, amelynek módosítjuk az 1,4 elemét *d* értékkel

int RobotGeometry::TranslationY(double d, Mat<double> &T)

Eltolás y mentén. A visszatérítendő *T* mátrix egy egységmátrix, amelynek módosítjuk a 2,4 elemét *d* értékkel *int RobotGeometry::TranslationZ(double d, Mat<double> &T)*

Eltolás *z* mentén. A visszatérítendő *T* mátrix egy egységmátrix, amelynek módosítjuk a 3,4 elemét *d* értékkel

int RobotGeometry::RotationX(double a, Mat<double> &T)

Elforgatás x körül. A visszatérítendő *T* mátrix egységmátrix, amelynek a bal felső 3X3 blokkját az alábbi mátrixxal módosítjuk:

$$Rot(z,a)=\begin{pmatrix}\cos a & -\sin a & 0\\ \sin a & \cos a & 0\\ 0 & 0 & 1\end{pmatrix}$$

int RobotGeometry::RotationY(double a, Mat<double> &T)

Elforgatás *y* körül. A visszatérítendő *T* mátrix egységmátrix, amelynek a bal felső 3X3 blokkját az alábbi mátrixxal módosítjuk:

$$Rot(z,a)=\begin{pmatrix}\cos a & 0 & \sin a\\ 0 & 1 & 0\\ -\sin a & 0 & \cos a\end{pmatrix}$$

int RobotGeometry::RotationZ(double a, Mat<double> &T)
Elforgatás z körül. A visszatérítendő *T* mátrix egységmátrix, amelynek a bal felső 3X3 blokkját az alábbi mátrixxal módosítjuk:

$$Rot(z,a)=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos a & -\sin a\\ 0 & \sin a & \cos a\end{pmatrix}$$

int RobotGeometry::DHMatrix(double theta, double d, double a, double alpha, Mat<double> &T)

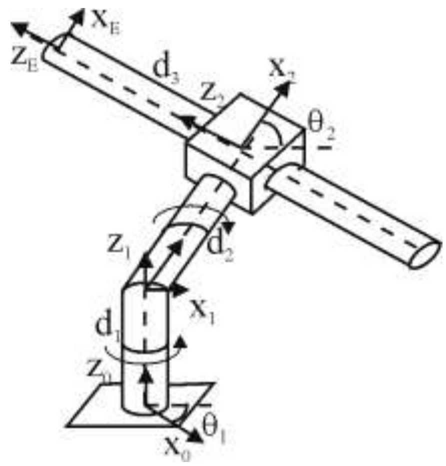
Denavtit Hartenberg mátrix egy csuklóra: A függvényben meghívjuk a *RotationZ*-t a *theta*-ra, a *TranslationZ*-t a *d*-re, a *TranslationX*-et az *a*-ra, a *RotationX*-et az *alpha*-ra, amjd a kapott 4 mátrixot összeszorozzuk és visszatérítjük a *T*-ben.

int RobotGeometry::DirectGeometry(int DOF, Col<double> Theta, Col<double> D, Col<double> A, Col<double> Alpha, Mat<double> &T)

A direkt geometriai feladat megoldása. A *DOF (Degree Of Freedom)* a robot szabadságfoka. A további függvény argumentumok oszlopvektorok, amelyek a teljes robotra tartalmazzák a *theta, d, a, alpha* paramétereket. A dimenziójuk meg kell egyezzen a robot szabadságfokával. A függvényben *DOF*-szor meghívjuk a *DHMatrix* függvényt az oszlopvektorok megfelelő elemeire, majd a kapott mátrixokat összeszorozzuk és visszatérítjük a *T* mátrixban.

3.3 A Stanford kar direkt geometriája

Legyen a 4 Ábrán látható a Stanford kar és a csuklókhoz hozzárendelt paraméterek. A kar Denavit Hartenberg paraméterei az 1. Táblázatban találhatóak.



4. Ábra: Stanford kar

1. Táblázat: A Stanford kar Denavit Hartenberg paraméterei

i	θ	*d*	*a*	α
1	$\theta_1=0$	*d1=1*	*0*	$\pi/2$
2	$\theta_2=\pi/2$	*d2=1*	*0*	$-\pi/2$
3	$\theta_3=\pi/2$	*d3=1*	*0*	*0*

A *RobotGeometryProj.cpp* állomány *main* függvényében hozzunk létre egy *RobotGeometry* obiektumot, majd töltsünk fel 4 sorvektort az 1 Táblázat szerint. A feltöltött vektorokkal hívjuk meg a *DirectGeometry* függvényt. Írassuk ki az eredményül kapott mátrixot, a 4. Ábra alapján ellenőrizzük az eredmény helyességét.

4. Kérdések és feladatok

1. Határozzuk meg analitikusan a Stanford kar végberendezésének *x, y, z* koordinátáját, majd teszteljük az eredményeket a programban kapott értékekkel.
2. Oldjuk meg analitikusan a Stanford kar direkt geometriai feladatát a Denavit Hartenberg táblázat alapján, majd teszteljük a megoldást a program kapott értékekkel.
3. Módosítsuk úgy a programot, hogy a mátrixok szorzását saját függvénnyel oldjuk meg.