Robotok irányítása

főiskolai jegyzet

– javított változat –

írta:

Tukora Balázs

Pécs, 2004

1. Bevezetés

Jelen jegyzet a Pécsi Tudományegyetem Pollack Mihály Műszaki Főiskolai Karán folyó Műszaki Informatika képzés Robotirányítási rendszerek I-II. tantárgyaihoz íródott. Célja a robotok irányításához és programozásához szükséges átfogó elméleti, matematikai alapokat megismertetni a hallgatókkal. A megértést, ahol csak lehet, kapcsolódó példák bemutatása hívatott elősegíteni. Az ábrák legnagyobb része és a programozási feladatok a FESTO cég COSIMIR nevű – ipari robotok programozására kifejlesztett és széles körben használt – keretrendszerének oktatási verziójával készültek.

2. Alapfogalmak

A széles körben elfogadott definíció szerint *ipari robot*nak a szabadon programozható, mozgást végző automatákat nevezzük. Mechanikai felépítésük – néhány ritka kivételtől eltekintve – rögzített alappontú, elágazás nélküli kinematikai lánccal felírható, vagyis az emberi karhoz hasonló eszközökről, robotkarokról van szó. Léteznek ún. manipulátorok illetve teleoperátorok, amelyek közvetlen mechanikus módon vagy közvetve, távirányítással követik a kezelő ember mozdulatait (például egy joystickkal irányítható, kamerával és manipulátorkarral felszerelt mobil bombakereső jármű), ezeket általában nem sorolják a robotok közé. A következő fejezetekben a robotok vezérlési kérdései kerülnek tárgyalásra. Kétféle szemszögből vizsgálhatjuk a témakört:

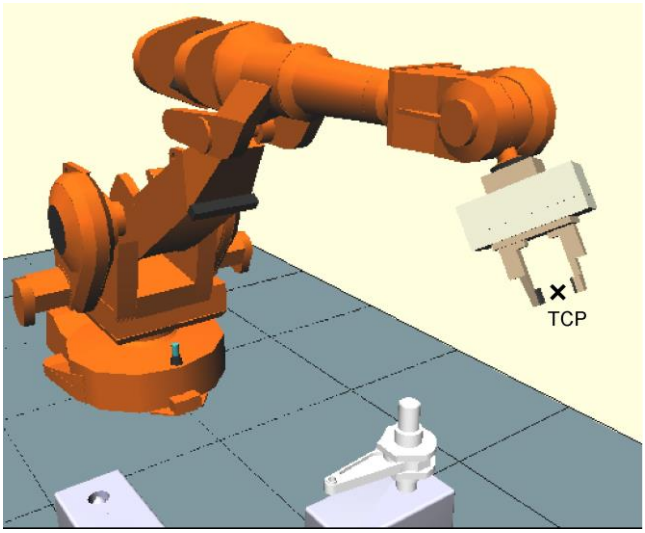
- A robotot üzemeltető szakember szempontjából, akinek az a célja, hogy az elvégzendő munkafolyamatot egyszerű, ám hatékony módon definiálja a robot számára.
- A robot (illetve tervezője) szempontjából, ahol a feladatot az jelenti, hogy a felhasználó által megadott utasításokat milyen matematikai szabályszerűségek és programozási technikák révén alakítsuk át tényleges, fizikai és egyéb kényszerektől befolyásolt mozgássá.

Lássuk csak, mit is csinál tulajdonképpen egy robotkar? Előre definiált útvonalon, pályán visz végig egy szerszámot, megfogót, szórópisztolyt, stb., és a pálya bizonyos pontjain ezekkel különféle műveleteket végez. A pálya bejárásának minősége különböző lehet:

- Bizonyos feladatoknál nem szükséges az útvonal kitüntetett pontjai közötti mozgást előre meghatározni. Ha például egy robotkarral át akarunk vitetni egy munkadarabot a robot egyik oldaláról a másikra, felesleges pontosan definiálnunk azt, hogy ezt milyen útvonalon tegye, elég csak a kezdő- és végpontot megadni. Ezt a vezérlést **PTP (Point to Point) irányítás**nak nevezik.

- Vannak esetek, ahol a szerszámnak vagy megfogott tárgynak szigorúan egyenes vonalban kell haladni: például egy tengely furatba illesztésénél. Más kötött pályaalakok is elképzelhetők. Ekkor **CP (Continuous Path), folytonos pályairányítás**ról beszélünk.
- Ívhegesztésnél vagy precíz festési, ragasztási munkálatoknál a szerszám elmozdulási illetve elfordulási sebessége is állandó kell legyen. Ez **sebességvezérlés**sel oldható meg.

Minden egyes szerszámra vagy megfogóra egy úgynevezett szerszám-középpontot (**TCP, Tool Center Point**) definiálunk (1. ábra). Ez lehet a festékszóró-pisztoly szórófeje, vagy az a pont, ahol az ujjak megfogják a munkadarabot. Pályabejárás alatt mindig a TCP adott pályájú mozgását értjük. Ezalatt a TCP pozíciója és orientációja (elfordulása) is a megkívánt módon változik.



1. ábra

A pályabejárás megtervezésénél még egyéb tényezőkre is ügyelnünk kell. Figyelembe kell vennünk a robot munkaterét:

- Milyen messze tud kinyúlni a kar, mit tudunk vele elérni a szerelőasztalon illetve egyéb magasságokban.

- Hol vannak azok a „holt terek” az elvileg ideális munkatéren belül, amelyeket az okoz, hogy a robot izületei csak bizonyos szögtartományban képesek elfordulni, elcsavarodni.
- Milyen akadályok nehezítik a robot szabad mozgását.

Egy robot sosem dolgozik „egyedül”. Emberek, más robotok, szerszámgépek, eszközök szolgálják ki őt, illetve szolgálja ki azokat. Érzékelőkkel van körülvéve, kamera segíti a munkáját. Ha egy robot feladata egy sajtológépbe helyezni az alapanyagot, majd kivenni a készre préselt munkadarabot, akkor pontosan tudnia kell, hogy a prés mikor van nyitva, mikor dolgozik, illetve tudatnia kell a gép felé, hogy mikor indíthatja el a sajtolást. Tulajdonképpen még a robotkar megfogója is a „külvilág” része. Függetlenül vezérelhető egység, szenzorokkal felszerelve. Éppen ezért a vezérlőegységnek több be- és kimenettel is rendelkeznie kell, amelyek a külvilággal való kapcsolatát biztosítják. A szenzorok jeleinek fogadásához esetenként elegendő lehet egy egyszerű bináris kapcsolat, intelligens eszközök bonyolultabb csatolókat igényelnek. A robot vezérlésénél figyelni illetve felügyelni kell ezeket a kapcsolatokat; állapotuktól függően várakozást illetve elágazásokat iktathatunk a munkafolyamatba.

Az eddigiek alapján a következő egységek szükségesek egy robot vezérlési feladatainak elvégzéséhez:

- CPU és aritmetikai processzor modul
- Memória modul a program illetve pályaadatok tárolásához
- Külső adattároló
- Kezelő egység (terminál)
- Kézi vezérlő egység
- I/O modul
    - Bináris I/O
    - Analóg egység
    - Digitális soros/párhuzamos egység
    - Egyéb interface-ek

- Szervo modul a motorszabályozáshoz
- Egyéb közvetlen szerszámvezérlő egységek

3. A robotok programozásának módjai

A sokfajta elvégzendő munkafolyamat, illetve az ezekhez tervezett különféle robotok más és más programozási technikákat igényelnek. Az ún. *online programozás* megköveteli a robot jelenlétét: a robotot vagy annak modelljét mozgatva tanítjuk be a bejárandó útvonalat. Az *offline programozás*i mód alkalmazásakor nincs szükségünk a robotra, egy számítógép mellett ülve, 3 dimenziós objektum-szimuláció segítségével vagy egyszerű szöveges bevitellel írjuk meg a programot.

3.1 Online programozás

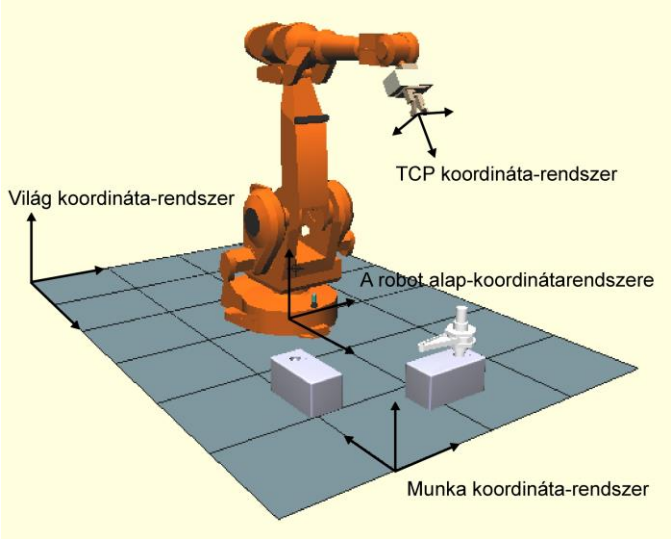
Az online programozás fogalma különböző technikákat takar:

- *Direkt betanítás (Direct Teach-In)*: Olyan helyeken alkalmazzák, ahol a robot folytonos pályairányítással vagy sebességvezérléssel mozgatja a szerszámot, viszonylag bonyolult pályát bejárva (például autókarosszériák festésénél). A technikus végigvezeti a robot karját a kívánt útvonalon, miközben a vezérlőegység folyamatosan feljegyzi a robotkar helyzetét, így később önállóan visszajátszhatja azt. Nagyméretű robotoknál a robotkar könnyített, hajtások nélküli modelljét mozgatja a betanító. Ezt a módszert angolul *Master-Slave Teach-in (mester-szolga betanítás)* néven említik.
- *Indirekt betanítás (Indirect Teach-In*, vagy egyszerűen *Teach-In):* Egy kézi vezérlőberendezés segítségével a robotot a pálya lényeges pontjaiba mozgatjuk, és ezek helyzetét memorizáljuk. A robot feladata lesz a pontok közötti pálya megtervezése és kiszámítása.

3.2 Offline programozás

Offline programozásnál a robot működésének definiálása általában valamilyen magas szintű nyelven történik. A program megírása előtt minden esetben be kell táplálni a bejárandó mozgáspálya kitüntetett pontjait. Ez történhet online betanítással, vagy a robottól teljesen függetlenül, szöveges vagy grafikus adatbevitellel.

Bár a TCP helyzetének és orientációjának leírásához elég a robot izületeinek aktuális állását rögzíteni, a programozó számára ez nem elég szemléletes, és nehezen kiszámítható. Éppen ezért a robot munkakörnyezetében különböző koordináta-rendszereket definiálunk, és a pontokat ezekre vonatkoztatjuk. A 2. ábra néhány általánosan használt koordináta-rendszert mutat be.



2. ábra

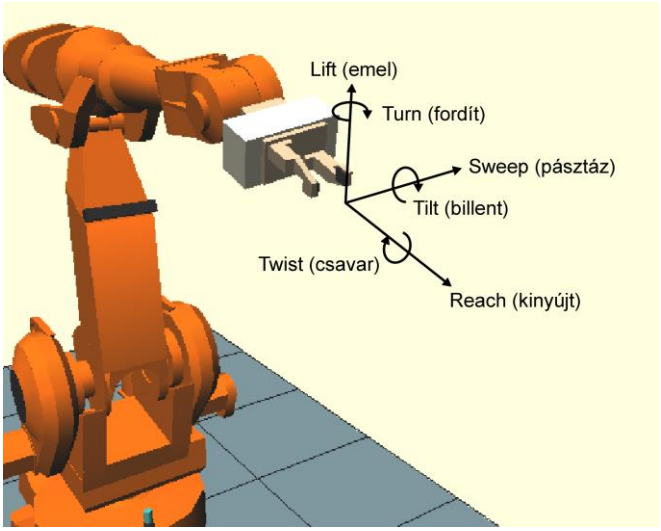
- A világ (vagy bázis) koordináta-rendszer a teljes munkaterület alap-koordinátarendszere. A TCP mozgását az egyszerűbb esetekben erre vonatkoztatjuk.
- A robot alap-koordinátarendszerét legtöbbször – a könnyebb számolás kedvéért – a világ koordináta-rendszerrel azonosnak tekintjük. Ha több

robot dolgozik együtt a munkatérben, akkor ez persze nem megvalósítható.

- A munka vagy más néven *aktuális* koordináta-rendszer az éppen végzett munkafolyamat alapját jelöli. Ehhez a koordináta-rendszerhez rendelhetjük a munkadarabok helyzetét. A munkadarabokhoz további koordináta-rendszereket is rendelhetünk.
- TCP (szerszám) koordináta-rendszer: pozíciója és orientációja a világ koordináta-rendszerhez képest egyértelműen definiálja a szerszám helyzetét. A TCP koordináta-rendszerben kiszámíthatjuk a megfogandó munkadarab távolságát és megközelítési irányát.

Az off-line programozás illetve a szabadon definiált koordináta-rendszerek használata több olyan lehetőséget nyújt, amelyet az egyszerűbb programozási módok nem biztosítottak:

- A pálya algoritmikus módszerekkel számítható, így könnyedén beprogramozhatunk olyan ciklikusan változó paraméterű folyamatokat, mint például a rácsszerűen elhelyezkedő furatok egymás utáni elkészítése.
- A térpontok helyzetét mindig csak abban a koordináta-rendszerben kell megadnunk, amelyikben az a legszemléletesebb, a legkönnyebben számítható. Egy, a munkadarabon levő lyuk helyzetét a leglogikusabb magához a munkadarabhoz – illetve a hozzá rendelt koordináta-rendszerhez – viszonyítani. Ha a munkadarab valamilyen mozgást végez, annak mozgásának alapján – a későbbiekben tárgyalt módszerekkel – a lyuk mozgása is viszonylag könnyedén meghatározható bármely vonatkoztatási rendszerben.
- A mozgások definiálása is szemléletesebbé válik, ha a megfelelő koordináta-rendszerre vonatkoztatjuk őket. A 3. ábra a szerszám szemszögéből elvégezhető mozdulatokat mutatja be: ezek a TCP koordináta-rendszerben történő egyszerű eltolások és elforgatások eredményei.



3. ábra

- A koordináta-rendszereknek nem kell feltétlenül derékszögűnek lenniük. Ha a robot geometriája mást kíván meg, alkalmazhatunk henger- vagy gömbi koordináta-rendszereket is (RTT illetve RRT karoknál).

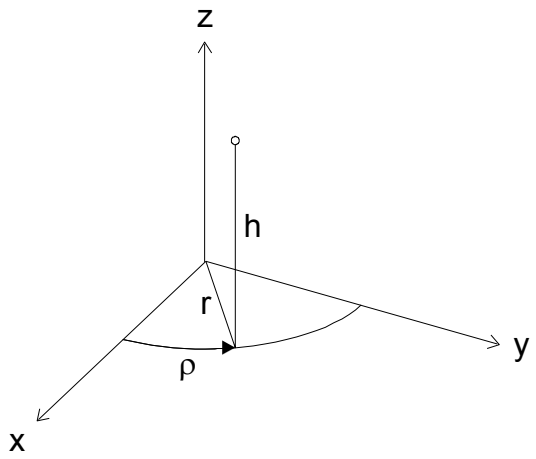
4. Pozíció és orientáció meghatározása

Ha egy szerszámnak vagy munkadarabnak meg akarjuk adni a pontos helyzetét egy bizonyos koordináta-rendszerben, akkor egyrészt a térbeli pozícióját, másrészt az orientációját (irányítottságát) kell meghatároznunk. Ennek érdekében a tárgyhoz kell rendelnünk egy saját koordináta-rendszert, és ennek a koordináta-rendszernek az alap-koordinátarendszerhez viszonyított pozícióját illetve elfordulását kell számszerűleg megadnunk. A következőkben a robotikában általánosan használt helyzet-meghatározási módokat tárgyaljuk.

4.1 Pozíció meghatározása hengerkoordinátákkal

Egy pontnak (vagy egy koordináta-rendszer origójának) helyzetét megadó három hengerkoordináta jelentése a következő (4. ábra):

- r : a z tengelytől való távolság
- ρ : a z tengelyű, r sugarú körön történő elfordulás mértéke
- h : az xy alapsíktól való távolság

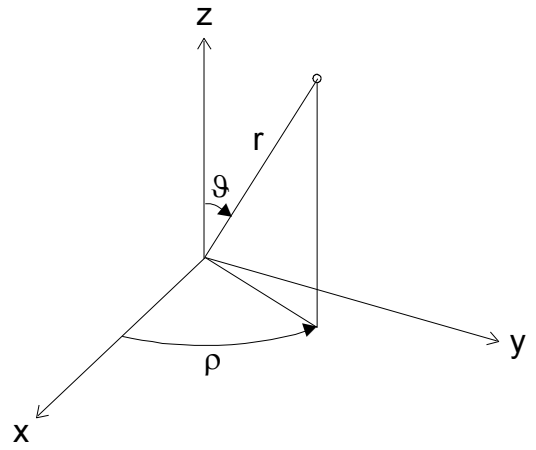


4. ábra

4.2 Pozíció megadása gömbkoordinátákkal

A gömb-koordinátarendszerben a következők szerint alakulnak az egyes koordináták (5. ábra):

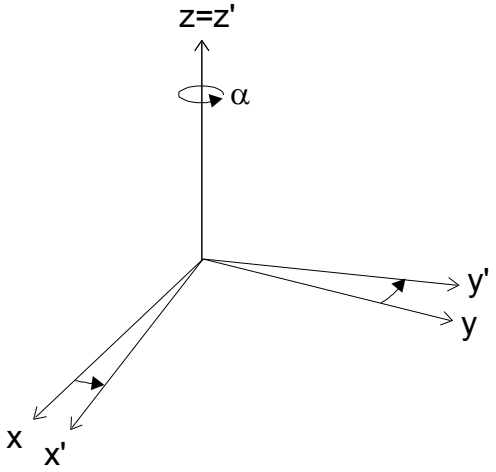
- r : a pontnak az origótól mért távolsága
- ρ : a pontba mutató vektor xy síkbeli vetületének az elfordulási szöge
- ϑ : a vektor z tengelytől mért elhajlásának a szöge



5. ábra

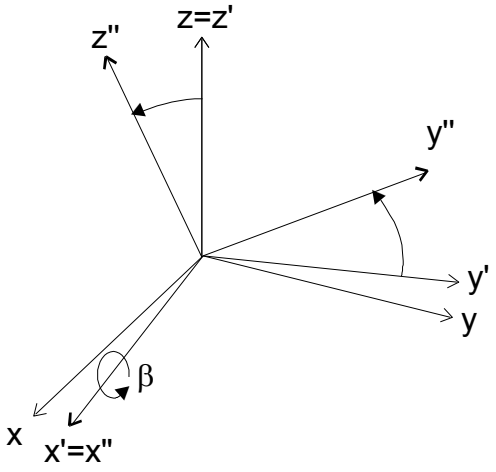
4.3 Orientáció megadása Euler-szögekkel

Forgassunk el egy koordináta-rendszert a z tengelye körül α szöggel (6. ábra)!



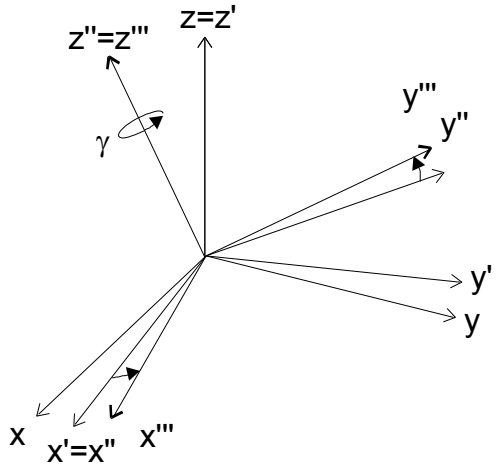
6. ábra

Most az elforgatással kapott x' tengely körül forgassuk tovább β szöggel (7. ábra)!



7. ábra

Ezután z'' körül forgassunk γ szöggel (8. ábra)!



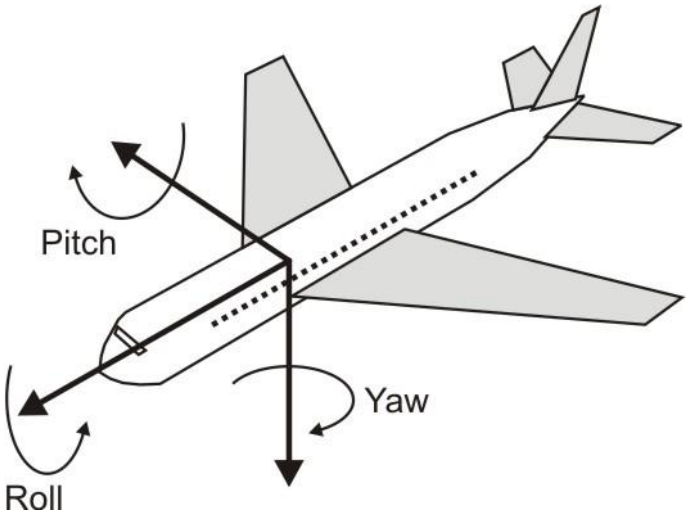
8. ábra

Az x-y-z és az x’’’-y’’’-z’’’ koordináta-rendszerek (így tetszőleges két koordináta-rendszer) orientációjának viszonyát egyértelműen megadja az α, β, γ számhármas. Ezeket a szögeket Euler-szögeknek nevezzük.

4.4 Orientáció megadása roll-pitch-yaw (billenés-bólintás-elfordulás) transzformációkkal

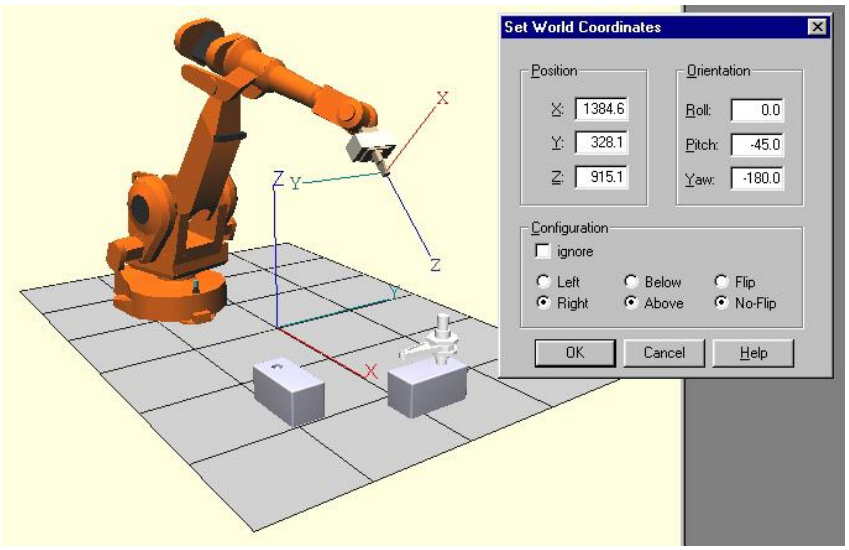
Egy pilóta a repülőgépének haladási irányát a következőképp változtathatja meg (9. ábra):

- Az ún. csűrőlapokkal megbillenti a gépet annak hossztengelye körül (roll);
- a magassági kormánnyal bólintásra készteti, a szárnyak által kifeszített oldalirányú tengely körül (pitch);
- az oldalkormánnyal a függőleges tengely körül fordítja el a repülőt (yaw).



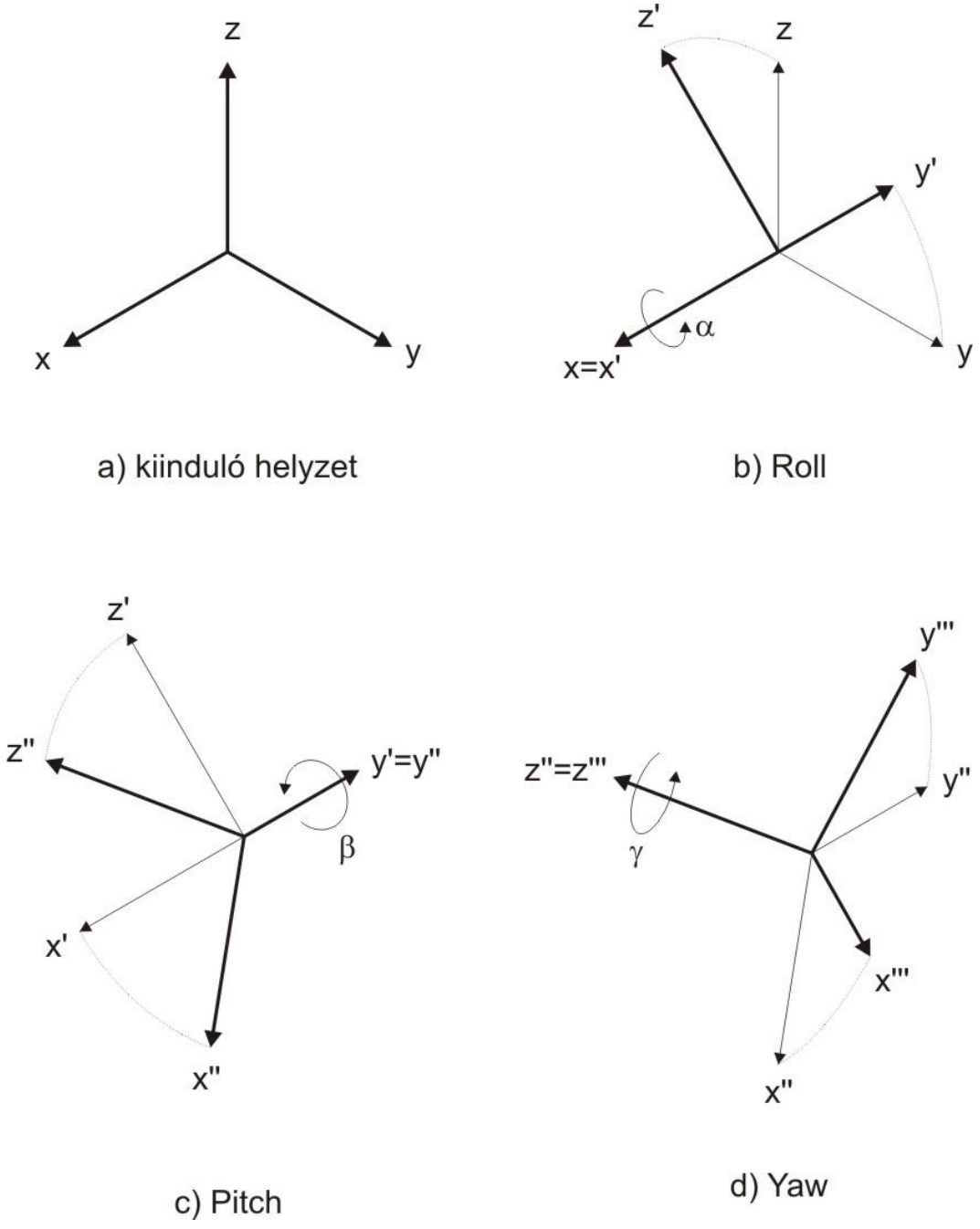
9. ábra

Ez egy eléggé szemléletes módszer, így a robottechnikában is elterjedt: például a megfogó vagy szerszám irányultságának megadásához a világ koordináta-rendszerhez képest (10. ábra).



10. ábra

A roll-pitch-yaw forgatások rendre az x, y, z tengely körül történnek α, β, majd γ szögekkel (11. ábra).

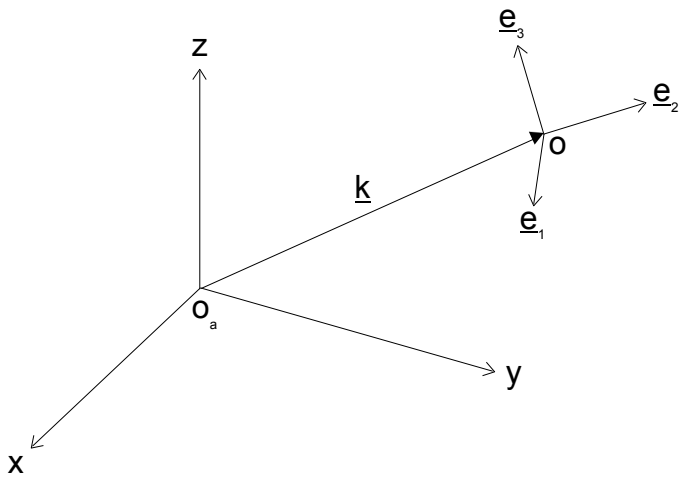

z
x
y
a) kiinduló helyzet
z'
z
y'
α
x=x'
y
b) Roll
z'
z"
y'=y"
β
x'
x"
c) Pitch
y'''
z"=z'''
y"
γ
x'''
x"
d) Yaw

11. ábra

5. Homogén transzformációk

A fenti módokon képesek vagyunk egy tárgy pozícióját és orientációját külön-külön megadni. A homogén transzformációk használatával ezt egyszerre, egyetlen matematikai formulában is megtehetjük. Látni fogjuk, hogy a homogén transzformációk módszerével a robottechnikában felmerülő összes geometriai feladatot kezelni tudjuk.

Vegyünk két koordináta-rendszert! Az O origójú koordináta-rendszer helyzetét akarjuk megadni az O_a alaprendszerhez képest (12. ábra).



12. ábra

A pozíciót a $\underline{k}$ vektor definiálja. Felírhatjuk mátrixos alakban $\underline{k}$ alaprendszerbeli koordinátáit:

$$\underline{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}. \qquad (1)$$

Az orientáció egy háromszor hármas, ún. forgatómátrixszal jellemezhető. A forgatómátrix az alaprendszer x,y,z irányaira vett vetületeinek értékeit tartalmazza. Más szóval a mátrix három oszlopa sorra az $\underline{e}_1$, $\underline{e}_2$, $\underline{e}_3$ vektorokat írja le úgy, mintha az alap koordináta-rendszer origójából indulnának ki.

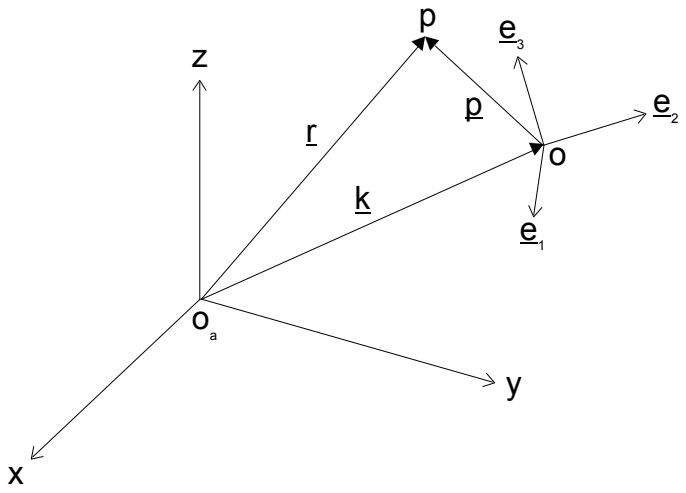
$$\underline{\underline{R}} = \begin{bmatrix} \underline{e}_1 & \underline{e}_2 & \underline{e}_3 \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix}. \tag{2}$$

Amennyiben a két mátrixot egyetlen, homogén transzformációs mátrixba rendezzük az alábbi módon:

$$\underline{\underline{H}} = \left[\begin{array}{ccc:c} & \underline{\underline{R}} & & \underline{k} \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} \underline{e}_1 & \underline{e}_2 & \underline{e}_3 & \underline{k} \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{3}$$

akkor bármely O koordináta-rendszerbeli pont helyzete kiszámítható az alaprendszerben (13. ábra), a következő formulával:

$$\underline{r} = \underline{\underline{H}} \cdot \underline{p}, \text{ vagyis } \begin{bmatrix} r_x \\ r_y \\ r_z \\ \hdashline 1 \end{bmatrix} = \left[\begin{array}{ccc:c} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \hdashline 1 \end{bmatrix}. \tag{4}$$

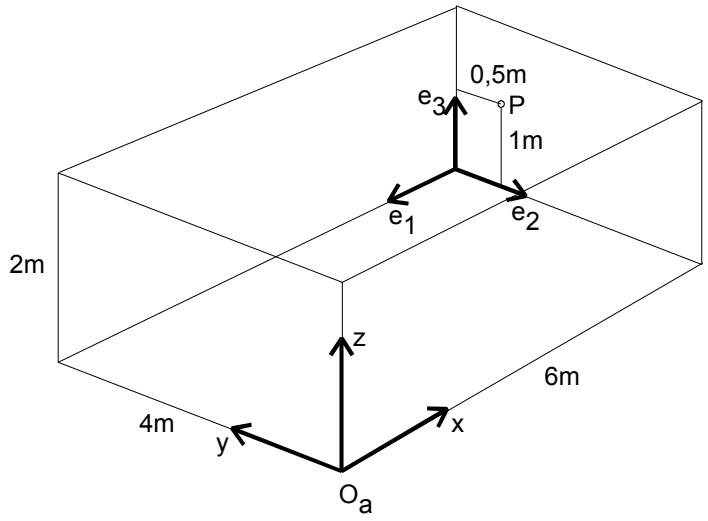


13. ábra

5.1 Példa homogén transzformációk alkalmazására

Helyezzünk el a 14. ábra szerint egy terem két sarkában egy-egy koordináta-rendszert; az alaprendszer legyen az innenső sarkon. A túlsó

koordináta-rendszerhez képest definiáljunk egy P pontot. Határozzuk meg a P pont alap-koordinátarendszerbeli pozícióját!



14. ábra

Megoldás:

A megoldáshoz a (4) egyenletet kell felhasználnunk:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = \left[\begin{array}{ccc:c} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ 1 \end{bmatrix}. \tag{5}$$

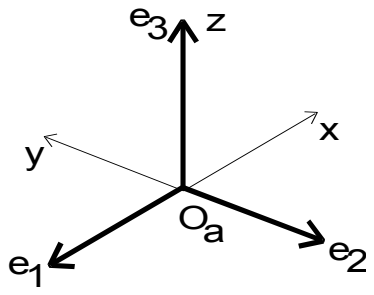
Az egyenlet kitöltéséhez először határozzuk meg a P pont koordinátáit a távoli koordináta-rendszerben. Ez ugye nem más, mint az origótól mért távolsága sorra az e_1, e_2, e_3 irányban:

$$\underline{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0{,}5 \\ 1 \end{bmatrix}. \tag{6}$$

Másodikként határozzuk meg a transzformációs mátrixot. A távolabbi koordináta-rendszer az alaprendszer egységvektorainak irányában a következő mértékben van eltolva:

$$\underline{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 0 \end{bmatrix}. \tag{7}$$

Az orientációs mátrix felírásában segít a 15. ábra. Rajzoljuk fel a két koordináta-rendszert közös origóval, és olvassuk le az $\underline{e}_1$, majd az $\underline{e}_2$ és $\underline{e}_3$ vektorok koordinátáit az alaprendszerben.



15. ábra

$$\underline{\underline{R}} = \begin{bmatrix} \underline{e}_1 & \underline{e}_2 & \underline{e}_3 \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

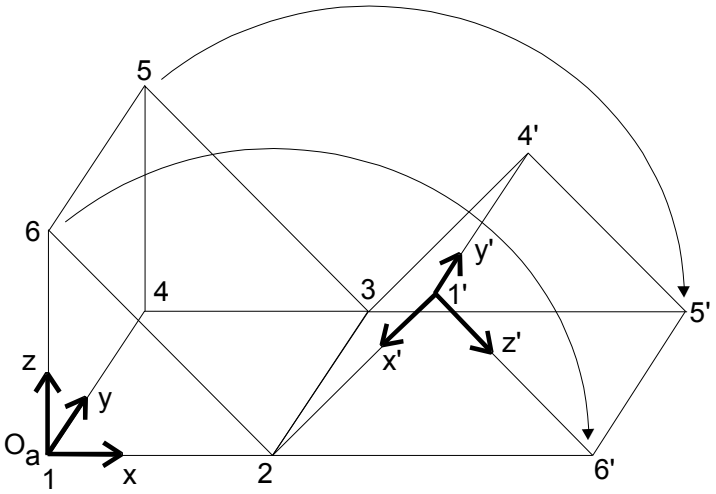
Így már meg tudjuk oldani a mátrix-egyenletet:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = \left[\begin{array}{ccc:c} -1 & 0 & 0 & 6 \\ 0 & -1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 0{,}5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \cdot 0 + 0 \cdot 0{,}5 + 0 \cdot 1 + 6 \cdot 1 \\ 0 \cdot 0 - 1 \cdot 0{,}5 + 0 \cdot 1 + 4 \cdot 1 \\ 0 \cdot 0 + 0 \cdot 0{,}5 + 1 \cdot 1 + 0 \cdot 1 \\ \hdashline 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 3{,}5 \\ 1 \\ 1 \end{bmatrix}. \tag{9}$$

Vagyis a koordináták: 6; 3,5 illetve 1, ami az ábráról is könnyen ellenőrizhető.

5.2 Példa: szilárd test mozgatása

Forgassunk el egy hasábot a 16. ábra szerinti módon! Számítsuk ki a hasáb csúcsainak elmozdítás utáni koordinátáit az O_a alaprendszerben!



16. ábra

A csúcsok eredeti koordinátái:

$$Cs_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix};\ Cs_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};\ Cs_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix};\ Cs_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix};\ Cs_5 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix};\ Cs_6 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Megoldás:

Írjuk fel az elmozdítás homogén transzformációs mátrixát!

A testhez rögzített koordináta-rendszer origója a következő helyre került ((4) jelöléseivel) :

$$\underline{k} = \begin{bmatrix} 1+\cos 45^\circ \\ 0 \\ \sin 45^\circ \end{bmatrix} = \begin{bmatrix} 1+\frac{\sqrt{2}}{2} \\ 0 \\ \frac{\sqrt{2}}{2} \end{bmatrix}. \qquad (10)$$

A forgatómátrix (az x-z síkban -135º-os forgatás történt):

$$\underline{\underline{R}} = \begin{bmatrix} \underline{x}' & \underline{y}' & \underline{z}' \end{bmatrix} = \begin{bmatrix} \cos -135^\circ & 0 & \cos -45^\circ \\ 0 & 1 & 0 \\ \sin -135^\circ & 0 & \sin -45^\circ \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \end{bmatrix}, \text{ így} \qquad (11)$$

$$\underline{\underline{H}} = \left[\begin{array}{ccc|c} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 1+\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{12}$$

A csúcsok a koordináta-rendszerrel együtt mozdultak el, ezért a koordinátáik ehhez képest nem változtak. A (4) egyenletet használva az összes elmozdított csúcs új koordinátái megkaphatók az alap-rendszerben:

$$\left[\begin{array}{c} \underline{Cs'}_i \\ \hline 1 \end{array}\right] = \underline{\underline{H}} \cdot \left[\begin{array}{c} \underline{Cs}_i \\ \hline 1 \end{array}\right], \text{ i=1..6.} \tag{13}$$

Az egyszerűség kedvéért helyezzük az összes csúcsot egyetlen mátrixba:

$$\underline{\underline{Cs}} = \left[\begin{array}{cccccc} \underline{Cs}_1 & \underline{Cs}_2 & \underline{Cs}_3 & \underline{Cs}_4 & \underline{Cs}_5 & \underline{Cs}_6 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] = \left[\begin{array}{cccccc} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right], \text{ ezzel} \tag{14}$$

$$\underline{\underline{Cs}}' = \underline{\underline{H}} \cdot \underline{\underline{Cs}} = \left[\begin{array}{ccc|c} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 1+\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \hline 1 & 1 & 1 & 1 \end{array}\right] \cdot \left[\begin{array}{cccccc} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] =$$

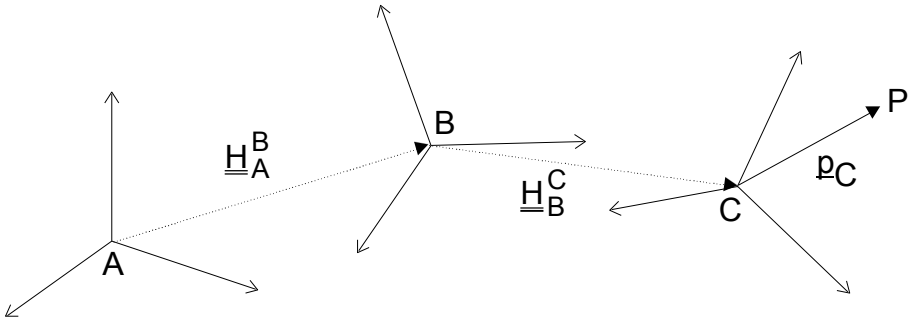
$$= \left[\begin{array}{cccccc} 1+\frac{\sqrt{2}}{2} & 1 & 1 & 1+\frac{\sqrt{2}}{2} & 1+\sqrt{2} & 1+\sqrt{2} \\ 0 & 0 & 1 & 1 & 1 & 0 \\ \frac{\sqrt{2}}{2} & 0 & 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] \tag{15}$$

A mátrix oszlopai sorra az egyes csúcsok új koordinátáit adják. Az ábra alapján ez is ellenőrizhető.

6. Relatív transzformációk

Mint arról már korábban szó volt, az alap- (világ) koordináta-rendszeren kívül más koordináta-rendszereket is használunk a robotmanipulációs feladatoknál. Ezeknek a koordináta-rendszereknek csak némelyike van az alaprendszerhez képest definiálva. Egy tárgyon elhelyezkedő furatok helyzetét a tárgyhoz rögzített koordináta-rendszerben célszerű megadni, a tárgy koordináta-rendszere a munka koordináta-rendszerhez viszonyított, és csak ez a rendszer van (esetleg) közvetlen kapcsolatban a világ koordináta-rendszerrel. A *relatív transzformáció*k segítségével bármely két közvetetten kapcsolódó rendszer egymáshoz viszonyított közvetlen helyzete is kiszámítható.

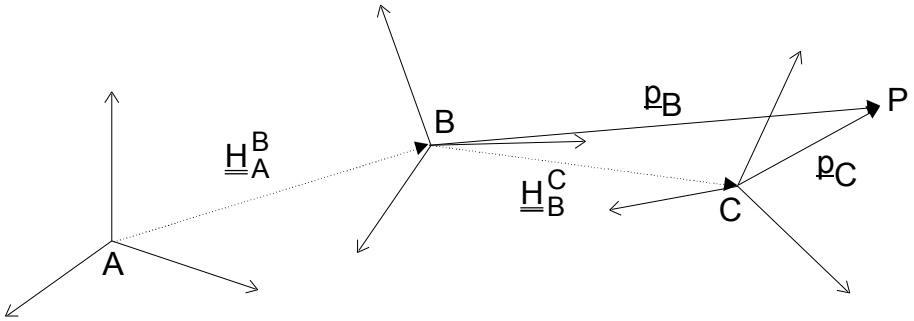
Vegyünk három koordináta-rendszert: A-t, B-t és C-t! B helyzetét A-hoz képest ismerjük, C helyzetét B-hez képest (17. ábra). Az alsó indexek a viszonyítási rendszert jelölik. Hogyan tudjuk egy C-hez képest definiált pont helyzetét megadni A-ban?



17. ábra

A (4) egyenlettel meghatározhatjuk P helyzetét B-ben (18. ábra):

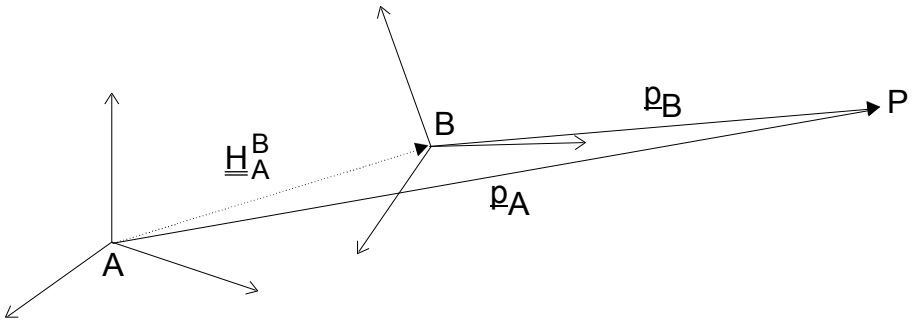
$$\underline{p}_B = \underline{\underline{H}}_B^C \cdot \underline{p}_C, \tag{16}$$



18. ábra

majd a kapott eredményt felhasználva P helyzetét A-ban (19. ábra):

$$\underline{p}_A = \underline{\underline{H}}_A^B \cdot \underline{p}_B . \tag{17}$$



19. ábra

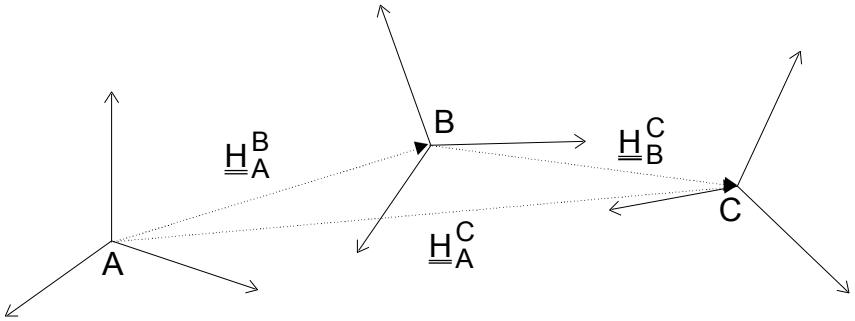
A két szorzást egyetlen egyenletbe is írhatjuk:

$$\underline{p}_A = \underline{\underline{H}}_A^B \cdot \underline{\underline{H}}_B^C \cdot \underline{p}_C . \tag{18}$$

$\underline{\underline{H}}_{AC}^B$ és $\underline{\underline{H}}_B^C$ összeszorzásával megkapjuk az A és C koordináta-rendszer viszonyát leíró transzformációs mátrixot (20. ábra):

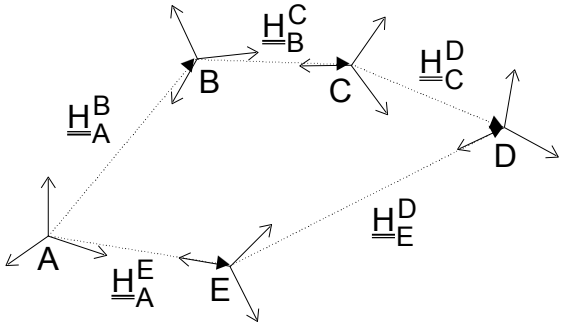
$$\underline{\underline{H}}_A^C = \underline{\underline{H}}_A^B \cdot \underline{\underline{H}}_B^C, \text{ így} \tag{19}$$

$$\underline{p}_A = \cdot \underline{\underline{H}}_A^C \cdot \underline{p}_C . \tag{20}$$



20. ábra

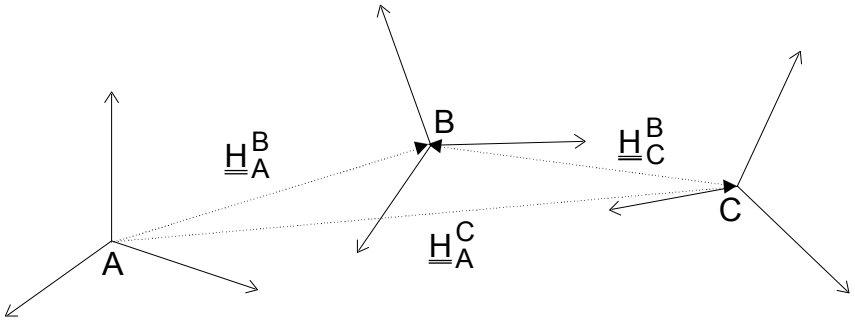
A (19) összefüggés úgy is felírható, ha az ábrán a kezdőpontból (A) a végpontig (C) követjük a nyilakat, sorra összeszorozzuk az egymás után következő mátrixokat, és a párhuzamos nyílfolyamok szorzatait egyenlővé tesszük egymással. A 21. ábra mutatja ennek általánosítását:



21. ábra

$$\underline{\underline{H}}_A^D = \underline{\underline{H}}_A^B \cdot \underline{\underline{H}}_B^C \cdot \underline{\underline{H}}_C^D = \underline{\underline{H}}_A^E \cdot \underline{\underline{H}}_E^D . \qquad (21)$$

Vajon hogyan oldható meg a feladat, ha nem C-t definiáltuk B-hez képest, hanem fordítva: B-t C-hez képest (22. ábra)?



22. ábra

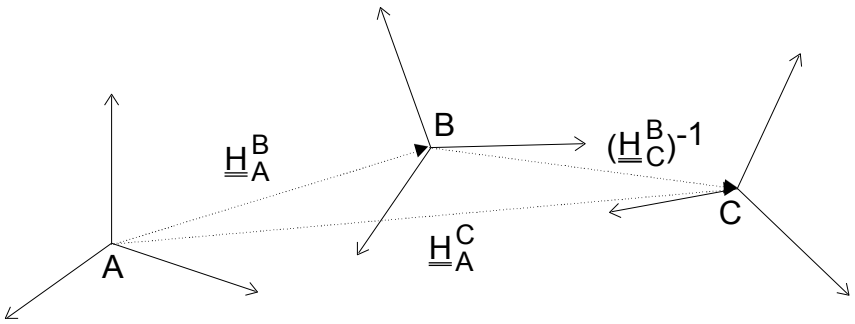
Az ábra szerint:

$$\underline{\underline{H}}_A^C \cdot \underline{\underline{H}}_C^B = \underline{\underline{H}}_A^B, \tag{22}$$

átrendezve:

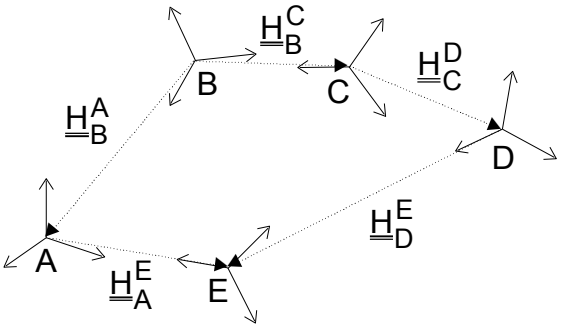
$$\underline{\underline{H}}_A^C = \underline{\underline{H}}_A^B \cdot \left(\underline{\underline{H}}_C^B\right)^{-1}, \tag{23}$$

vagyis a nyilak egy inverz képzéssel megfordíthatók, és így a lánc egyirányúvá tehető (23. ábra).



23. ábra

A 24. ábrán az A és D rendszerek közti viszony felírásához két utat választhatunk, egy-egy invertálással (24. egyenlet):



24. ábra

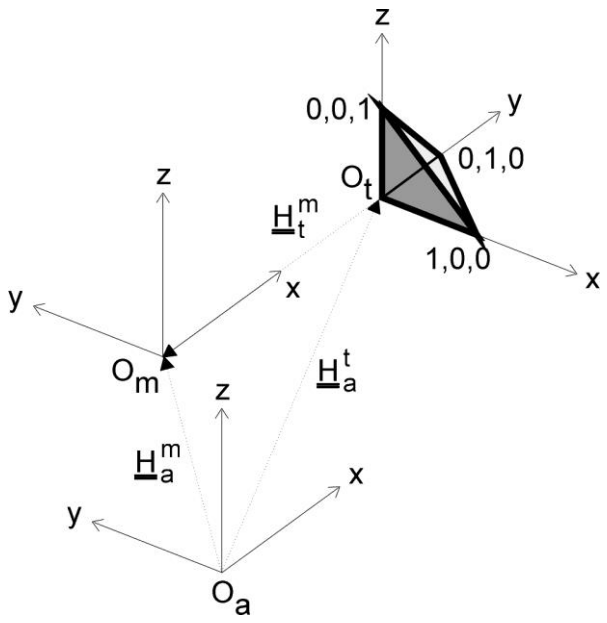
$$\underline{\underline{H}}_A^D = \left(\underline{\underline{H}}_B^A\right)^{-1} \cdot \underline{\underline{H}}_B^C \cdot \underline{\underline{H}}_C^D = \underline{\underline{H}}_A^E \cdot \left(\underline{\underline{H}}_D^E\right)^{-1}. \tag{24}$$

Könnyen belátható, hogy e módszerrel bármely két koordináta-rendszer viszonyát meghatározhatjuk.

6.1 Példa: relatív transzformációk számítása

Mik lesznek a 25. ábrán látható test csúcsainak koordinátái az alaprendszerben, ha a koordináta-rendszerek kapcsolatát leíró homogén transzformációs mátrixok az alábbiak:

$$\underline{\underline{H}}_a^m = \left[\begin{array}{ccc:c} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \quad \underline{\underline{H}}_t^m = \left[\begin{array}{ccc:c} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -10 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right].$$



25. ábra

Megoldás:

Mindenekelőtt ki kell számítanunk a tárgy (t) koordináta-rendszer alapkoordináta-rendszerbeli helyzetét megadó transzformációs mátrixot. Ehhez inverz képzésre lesz szükség, amit a homogén transzformációs mátrixokra érvényes speciális egyenlőséggel lehet egyszerűen meghatározni:

$$\underline{\underline{H}}^{-1} = \left[\begin{array}{ccc:c} \underline{e}_1 & \underline{e}_2 & \underline{e}_3 & \underline{k} \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]^{-1} = \left[\begin{array}{ccc:c} \underline{e}_1^{\,T} & & & -\underline{e}_1^{\,T} \cdot \underline{k} \\ \underline{e}_2^{\,T} & & & -\underline{e}_2^{\,T} \cdot \underline{k} \\ \underline{e}_3^{\,T} & & & -\underline{e}_3^{\,T} \cdot \underline{k} \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \tag{25}$$

Ezt felhasználva a keresett transzformációs mátrix:

$$\underline{\underline{H}}_a^t = \underline{\underline{H}}_a^m \cdot \underline{\underline{H}}_m^t = \underline{\underline{H}}_a^m \cdot (\underline{\underline{H}}_t^m)^{-1} =$$

$$= \left[\begin{array}{ccc:c} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \cdot \left[\begin{array}{ccc:c} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -10 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]^{-1} = \left[\begin{array}{ccc:c} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \cdot \left[\begin{array}{ccc:c} 0 & 1 & 0 & 10 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] =$$

$$= \left[\begin{array}{ccc:c} 0 & 1 & 0 & 18 \\ -1 & 0 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \qquad (26)$$

A csúcsok eredeti koordinátái:

$$\underline{\underline{Cs}}_t = \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hdashline 1 & 1 & 1 & 1 \end{array}\right],$$

így a csúcsok alap-koordinátarendszerbeli koordinátái:

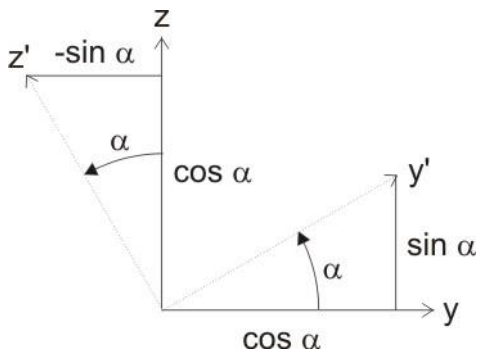
$$\underline{\underline{Cs_a}} = \underline{\underline{H}}_a^t \cdot \underline{\underline{Cs_t}} = \left[\begin{array}{ccc:c} 0 & 1 & 0 & 18 \\ -1 & 0 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \cdot \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hdashline 1 & 1 & 1 & 1 \end{array}\right] = \left[\begin{array}{cccc} 18 & 18 & 19 & 18 \\ 6 & 5 & 6 & 6 \\ 0 & 0 & 0 & 1 \\ \hdashline 1 & 1 & 1 & 1 \end{array}\right]. \qquad (27)$$

6.2 Példa: a transzformációs mátrix és az orientációs szögek viszonya

Robottechnikai feladatoknál sokszor előfordul (jelen jegyzetben is többször szükségünk lesz rá), hogy a roll-pitch-yaw vagy Euler-féle orientációs szögekből meg kell határoznunk az elforgatásokat leíró eredő transzformációs mátrixot, illetve egy forgatómátrixból az orientációs szögeket. A következő két példa ezeknek a menetét mutatja be.

6.2.1 A roll-pitch-yaw transzformáció mátrixos alakban történő megadása

A „roll” billenés az x körüli α szögű elforgatás eredménye (28. ábra).

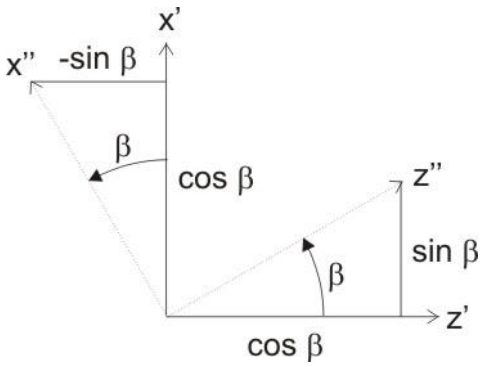


26. ábra

Az ezt leíró forgatómátrix:

$$\underline{\underline{R}}(\alpha)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha\\ 0 & \sin\alpha & \cos\alpha\end{bmatrix}. \qquad (28)$$

A „pitch” bólintás az elforgatott koordináta-rendszer y’ tengelye körüli forgatást jelent β szöggel (27. ábra).

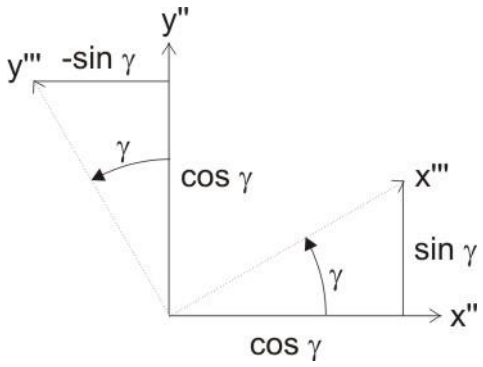


27. ábra

A forgatómátrix:

$$\underline{\underline{R}}(\beta)=\begin{bmatrix}\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{bmatrix}. \tag{29}$$

A „yaw” elfordulás a z tengely körül történik γ szöggel. A z tengely változatlan marad, x’ és y’ tengelyek koordinátái pedig a 26. ábra szerint alakulnak (z a lapból kifelé mutat).



28. ábra

Így a forgatómátrix:

$$\underline{\underline{R}}(\gamma)=\begin{bmatrix}\cos\gamma & -\sin\gamma & 0\\ \sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{bmatrix}. \tag{30}$$

Az eredő forgatómátrix az egymás után következő relatív transzformációk mátrixának szorzata, de vigyázzunk: a szorzatban megfordul a transzformációk sorrendje. Először ugyanis a „roll” forgatást végezzük el, és ennek eredményét szorozzuk be a relatív transzformációknak megfelelően balról a „pitch”, majd a „yaw” transzformációk mátrixával.

$$\underline{\underline{R}}(\gamma,\beta,\alpha)=\underline{\underline{R}}(\gamma)\cdot\underline{\underline{R}}(\beta)\cdot\underline{\underline{R}}(\alpha)=$$
$$=\begin{bmatrix}\cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha-\sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha+\sin\gamma\sin\alpha\\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha+\cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha-\cos\gamma\sin\alpha\\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha\end{bmatrix}. \tag{31}$$

A roll-pitch-yaw műveletet leíró homogén transzformációs mátrix tehát:

$$\underline{\underline{H}}(\gamma,\beta,\alpha) = \left[\begin{array}{ccc:c} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha-\sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha+\sin\gamma\sin\alpha & 0 \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha+\cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha-\cos\gamma\sin\alpha & 0 \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \tag{32}$$

6.2.2 A roll-pitch-yaw szögek kiszámítása az eredő homogén transzformációs mátrixból

Ebben az esetben a roll-pitch-yaw elforgatások eredménye ismert az alábbi mátrix alakjában (a 11. ábra jelöléseivel):

$$\underline{\underline{R}} = \left[\underline{x'''} \quad \underline{y'''} \quad \underline{z'''}\right] = \begin{bmatrix} x'''_x & y'''_x & z'''_x \\ x'''_y & y'''_y & z'''_y \\ x'''_z & y'''_z & z'''_z \end{bmatrix}. \tag{33}$$

A szögeket az alábbi mátrix-egyenlet megoldása adja:

$$\underline{\underline{R}}(\gamma,\beta,\alpha) = \begin{bmatrix} x'''_x & y'''_x & z'''_x \\ x'''_y & y'''_y & z'''_y \\ x'''_z & y'''_z & z'''_z \end{bmatrix} = \begin{bmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha-\sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha+\sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha+\cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha-\cos\gamma\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}. \tag{34}$$

Abban az esetben, ha $\cos\beta \neq 0$, a másik két szög könnyen kiszámítható a mátrixok azonos helyen álló tagjainak egyenlőségbe írásával:

$$\cos\gamma\cos\beta = x'''_x, \tag{35}$$

$$\sin\gamma\cos\beta = x'''_y, \tag{36}$$

amelyekből:

$$\gamma = \arctan\left(\frac{x'''_y}{x'''_x}\right), \tag{37}$$

és hasonlóan:

$$\cos\beta\sin\alpha = y_z''', \tag{38}$$

$$\cos\beta\cos\alpha = z_z''', \tag{39}$$

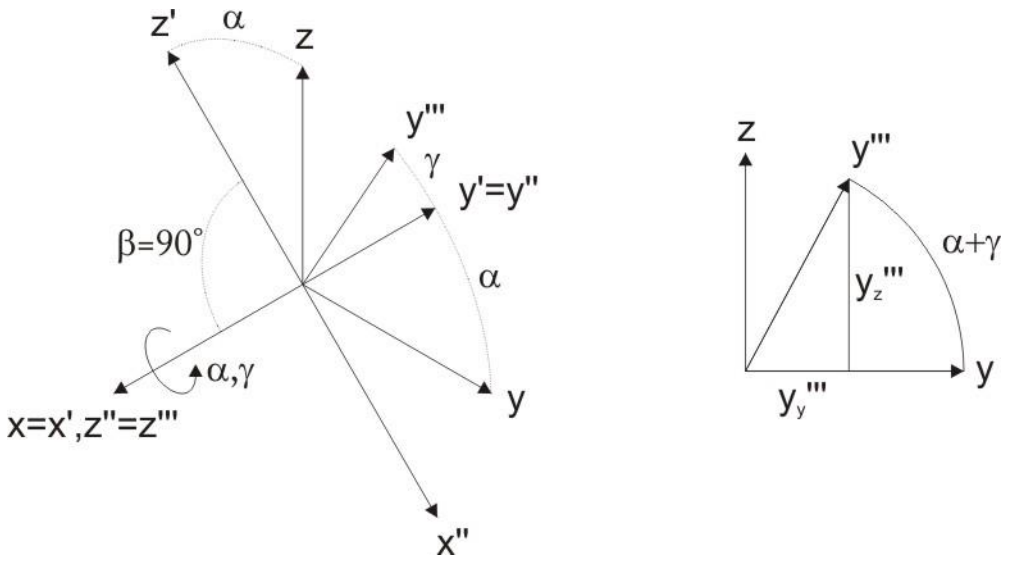
$$\alpha = \arctan\left(\frac{y_z'''}{z_z'''}\right). \tag{40}$$

β értéke pedig:

$$\beta = \arcsin(-x_z'''). \tag{41}$$

Amennyiben $\cos\beta = 0$, a koordináta-rendszert $\pm 90^\circ$-kal forgattuk el, α és γ egy síkba kerül (vö. a 11. ábrával), így összegük vagy különbségük határozza meg az elfordulást. Ha $\beta = 90^\circ$, akkor y''' $\alpha+\gamma$ szögben fordul el (29. ábra). Az ábra alapján:

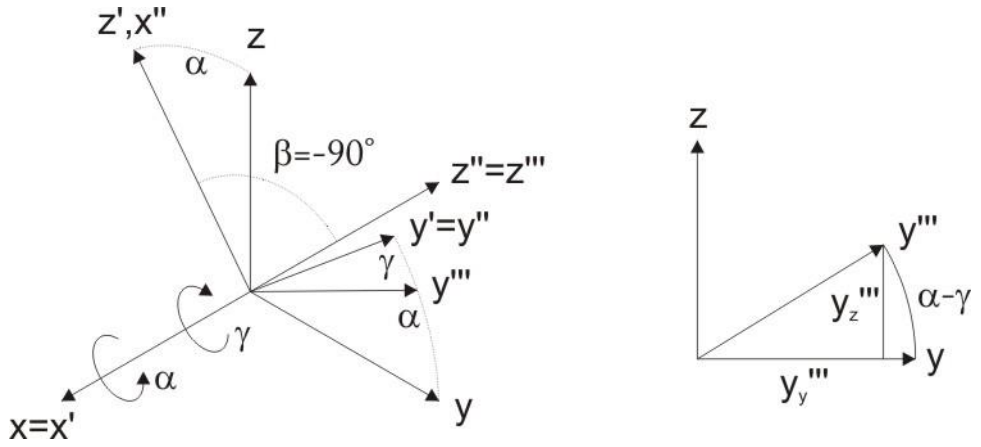
$$\alpha+\gamma = \arctan\left(\frac{y_z'''}{y_y'''}\right). \tag{42}$$



29. ábra

Ha β elfordulása -90°, a Yaw (γ) elforgatás a Roll (α) forgatással ellenkező irányú (30. ábra), tehát:

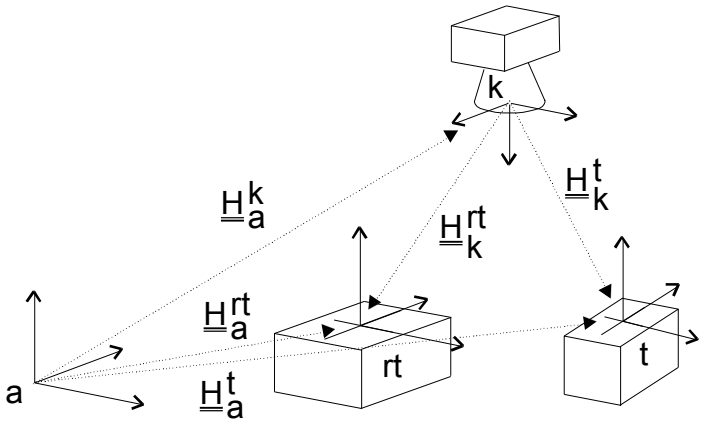
$$\alpha - \gamma = \arctan\left(\frac{y_z'''}{y_y'''}\right). \tag{43}$$



30. ábra

6.3 Vizuális információ feldolgozása

A robot „látóterébe” kerülő munkadarabok pontos helyzete nem mindig adható meg előzetesen (például a robotnak egy futószalagra össze-vissza lehelyezett alkatrészeket kell a megfelelő helyen és módon megragadnia). Ez esetben egy ipari kamera (és bizonyos képfeldolgozó software-ek) segítségével meghatározható a tárgyak pozíciója és orientációja. A 31. ábra egy olyan elrendezést mutat, ahol egy ismert pozíciójú referenciatárgy kamerabeli képe alapján először magának a kamerának a helyzetét számítjuk ki (erre persze rögzített kamera esetén nincs szükség), majd a vizsgált tárgyat is tartalmazó képet feldolgozva határozzuk meg a tárgy koordinátáit.



31. ábra

$$\underline{\underline{H}}_a^{rt} = \underline{\underline{H}}_a^{k} \cdot \underline{\underline{H}}_k^{rt}, \text{ így} \tag{44}$$

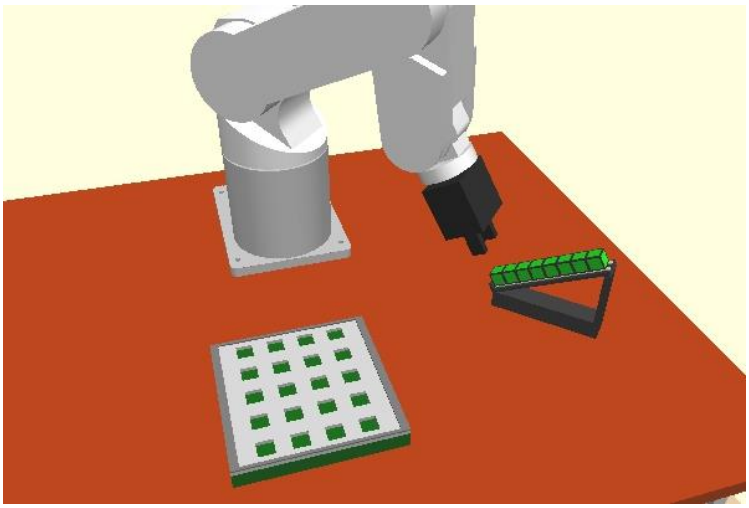
$$\underline{\underline{H}}_a^{k} = \underline{\underline{H}}_a^{rt} \cdot \left(\underline{\underline{H}}_k^{rt}\right)^{-1}. \tag{45}$$

Ezzel megvan a kamera helyzete. A tárgy transzformációs mátrixa pedig:

$$\underline{\underline{H}}_a^{t} = \underline{\underline{H}}_a^{k} \cdot \underline{\underline{H}}_k^{t} = \underline{\underline{H}}_a^{rt} \cdot \left(\underline{\underline{H}}_k^{rt}\right)^{-1} \cdot \underline{\underline{H}}_k^{t}. \tag{46}$$

6.4 Pályapontok számítása relatív transzformációkkal

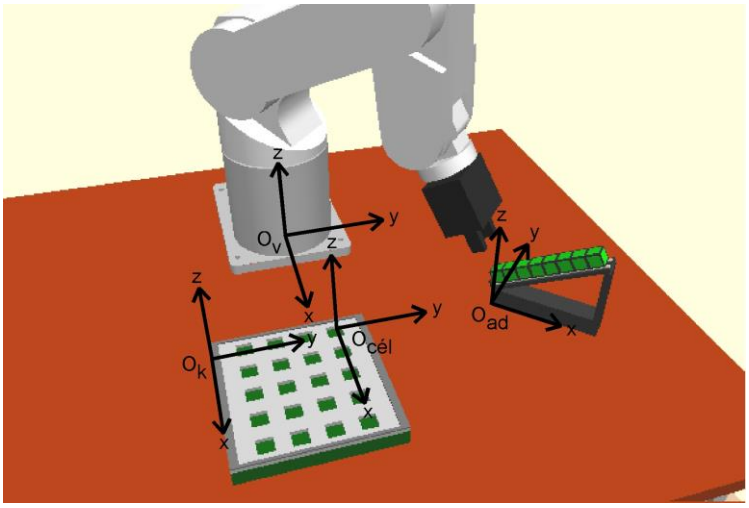
Mint arról már említést tettünk, PTP (Point-To-Point) programozásnál a TCP által bejárt pályának csak bizonyos pontjait kell megadnunk, a köztük bejárt utat a robotirányító számítógép határozza meg. E térpontokat a legegyszerűbb kézi vezérlőegységgel, on-line felvenni, ám sokszor ez a módszer nem elég pontos – ekkor a munkatérben levő különböző objektumok helyzetének ismeretében ki kell számítanunk a kívánt pozíciókat. A pályának ezen sarokpontjait sokszor a mozgatni, megmunkálni kívánt munkadarabhoz képest kell definiálni, például a munkadarab megközelítésénél vagy megfogásánál. Mivel az alkatrészek helyzetét is relatív transzformációkkal írjuk fel, célszerű a pályapontok számításánál is ezt a módszert követnünk. Lássunk erre egy példát!

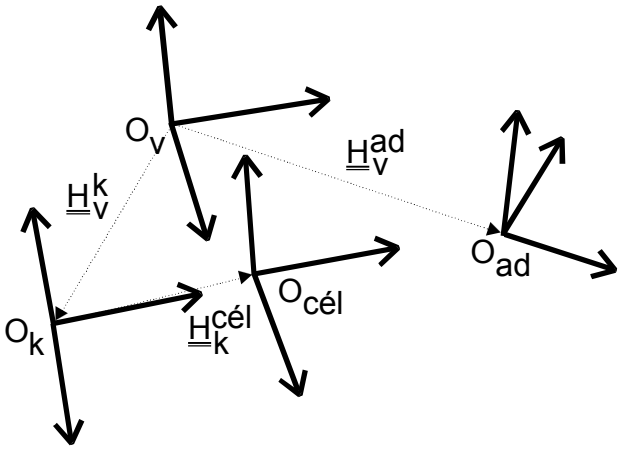


32. ábra

A 32. ábrán látható elrendezésben egy csúszdás adagolóról kell a kockákat rácsszerűen elhelyezkedő nyílásokba (ún. palettába) átrakosgatni. Ismerjük az adagoló helyzetét, a legalsó kocka elhelyezkedését az adagolón (kiemelésekor a következő kocka csúszik ennek a helyére), a rácsot magában foglaló keret helyzetét, és számítással meghatározhatók a rácspontok koordinátái. Vegyük sorra részletesen az ismert adatokat (számszerű értékek nélkül):

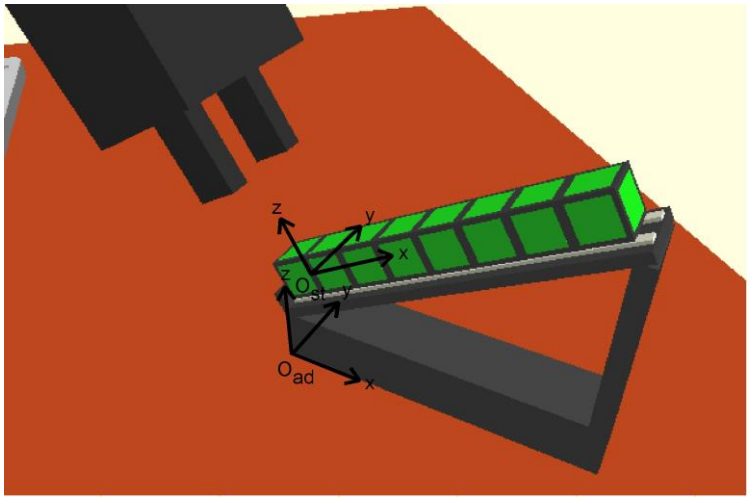
A világ koordináta-rendszer az egyszerűség kedvéért megegyezik a robot alap-koordinátarendszerével (O_v). Az adagoló koordináta-rendszerének és a világ-koordináta-rendszernek a viszonya: $\underline{\underline{H}}_v^{ad}$ (33. ábra).

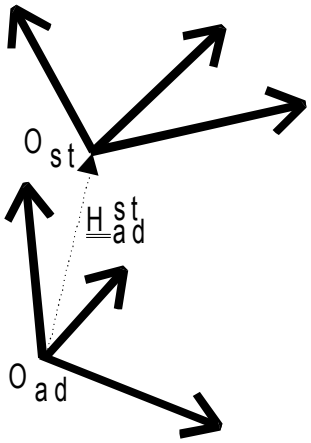




33**. ábra**

A legalsó kocka *megfogó pontj*ának (Grip Point – ahol a TCP-nak kell lennie, ha stabilan meg akarjuk ragadni) kiindulási állapotban az adagolóhoz viszonyított helyzete: $\underline{\underline{H}}_{ad}^{st}$ (34. ábra).





34. ábra

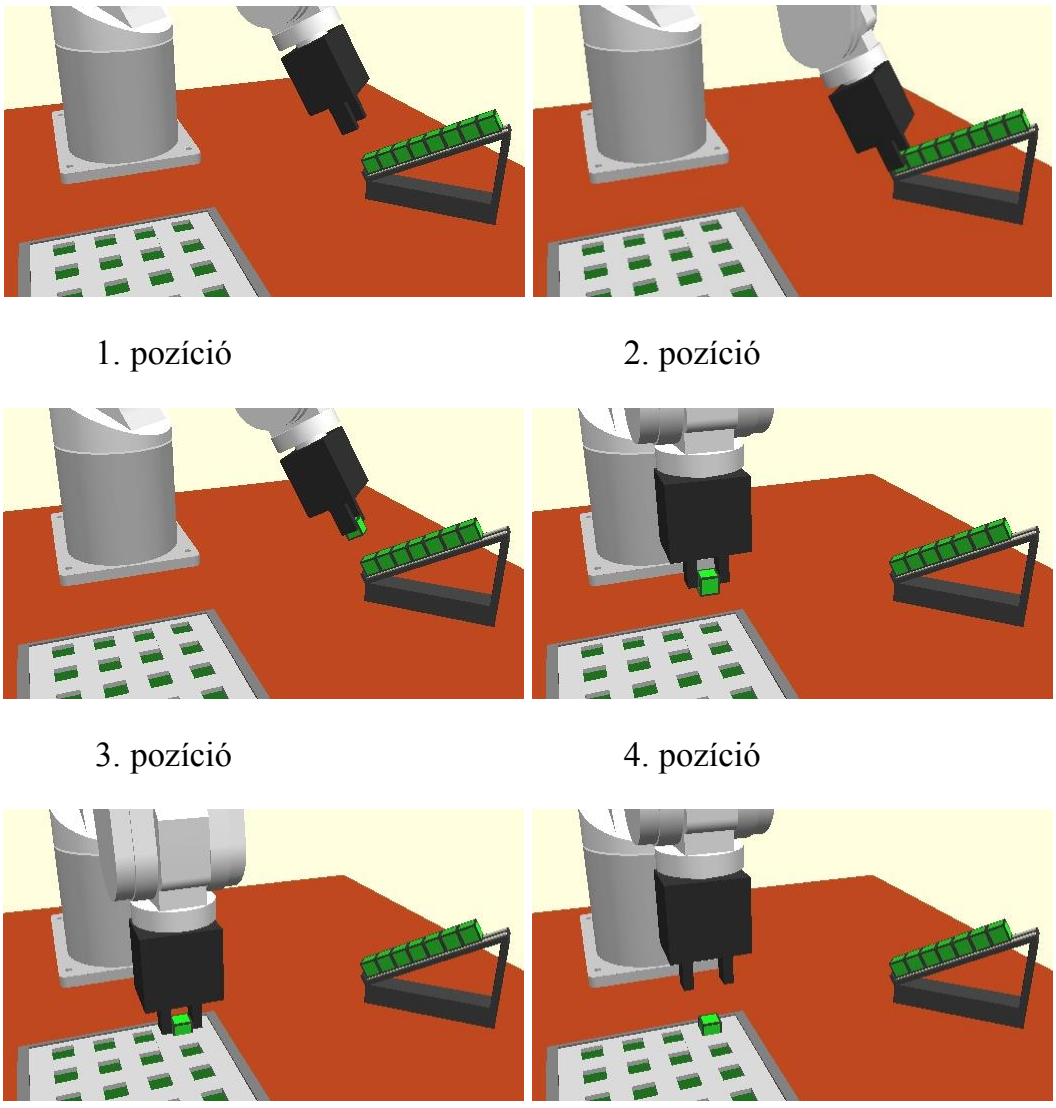
A keret koordináta-rendszerének a világ koordináta-rendszerhez viszonyított helyzete: $\underline{\underline{H}}_{v}^{k}$.

A legelső nyílásba helyezett kocka megfogó pontja a keret rendszerében (itt azért van szükség koordináta-rendszerre, és nem csak egy vektorra, mert a kocka irányultságát is meg akarjuk adni a cél állapotban), a 33. ábra szerint: $\underline{\underline{H}}_{k}^{cél}$. A többi nyílással most ne foglalkozzunk!

Ennyit ismerünk tehát. Hogyan jutunk el ezen adatokat felhasználva a robot beprogramozásáig?

1. lépés: A robot által végzett mozgássorozat kidolgozása

Az elvégzendő feladatot végiggondolva a pálya jellegzetes pontjai a következők lesznek (35. ábra):



1. pozíció 2. pozíció

3. pozíció 4. pozíció

5. pozíció 6. pozíció

35. ábra

A mozgássorozat az alábbiak szerint alakul:

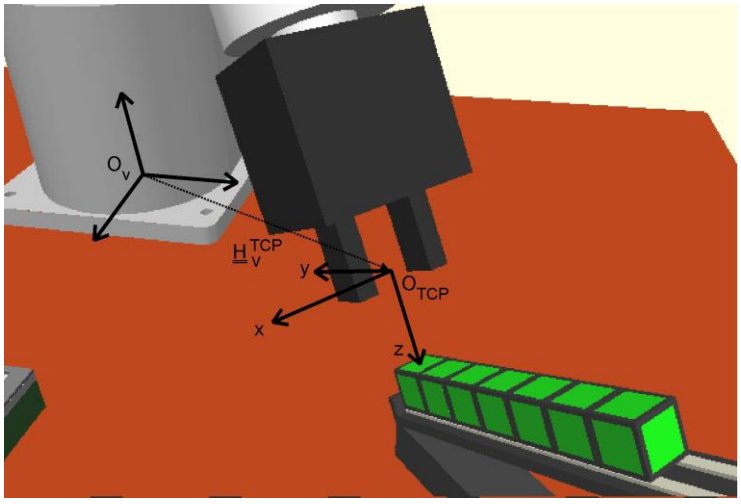
1. fázis: a munkadarab megközelítése tetszőleges görbéjű pályán (az 1. pozícióba jutás).

2. fázis: elmozdulás a megfogó-pozícióba. Az 1. pozícióból egyenes vonalú mozgással kell a 2. pozícióba érkeznünk.
3. fázis: a munkadarab kiemelése (szintén egyenes vonalban a 3. pozícióba).
4. fázis: elfordulás a paletta fölé (tetszőleges görbén a 4. pozícióba).
5. fázis: a munkadarab lyukba helyezése (lineáris pályán az 5. pozícióba).
6. fázis: eltávolodás a palettától (egyenes vonalban a 6. pozícióba).

2. lépés: A TCP helyzetének kiszámítása a pálya sarokpontjain

Az összes meghatározó pozícióban ki kell számítanunk a TCP helyzetét a robotkar alap-koordinátarendszerében (amely esetünkben megegyezik a világ koordináta-rendszerrel). A TCP mindenkori helyzetét legpontosabban az ismert adatokból relatív transzformációkkal határozhatjuk meg. Ehhez az szükséges, hogy a TCP-hez is rögzítsünk egy koordináta-rendszert a következő szabályokkal (36. ábra):

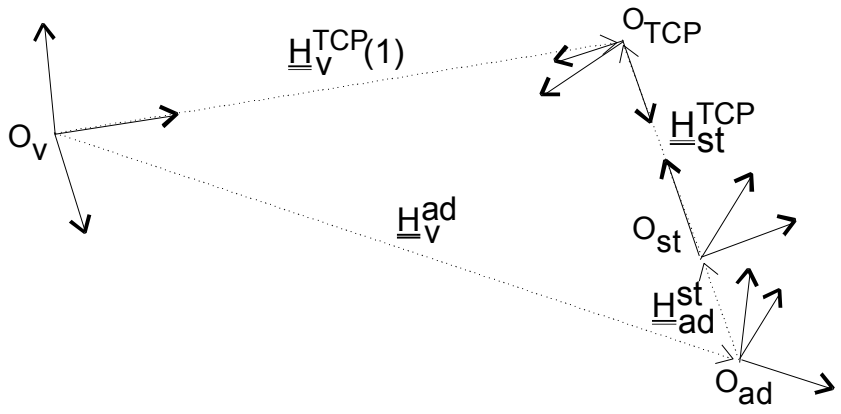
- A Z tengely a megközelítés irányába mutat.
- Az X tengely az egyik ujjtól a másik felé (mindegy, hogy melyiktől melyikig, de a számítások során végig azonos módon).
- Az Y tengely a jobbsodrású rendszer geometriája szerint az X és Z tengelyek helyzetéből adódik.



36. ábra

A TCP helyzetét az alap és a TCP koordináta-rendszer viszonyát leíró $\underline{\underline{H}}_{v}^{TCP}$ transzformációs mátrix alakjában keressük, minden egyes meghatározó pályapontban. Lássuk hát:

- 1. pozíció (a munkadarab megközelítése):



37. ábra

Mint tudjuk, $\underline{\underline{H}}_{v}^{ad}$ az adagoló helyzetét adja meg, $\underline{\underline{H}}_{ad}^{st}$ pedig a legalsó kockáét az adagolóhoz képest. $\underline{\underline{H}}_{st}^{TCP}$-vel azt definiáljuk, hogyan közelítse meg a megfogó a kockát az első pozícióban. Ha például 50 mm-re akarjuk megközelíteni, akkor az ezt leíró mátrix a következőképp néz ki (az index a pozíciót jelöli):

$$\underline{\underline{H}}_{st}^{TCP}(1) = \left[\begin{array}{ccc:c} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 50 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \tag{48}$$

Ezek alapján:

$$\underline{\underline{H}}_{v}^{TCP}(1) = \underline{\underline{H}}_{v}^{ad} \cdot \underline{\underline{H}}_{ad}^{st} \cdot \underline{\underline{H}}_{st}^{TCP}(1). \tag{47}$$

- 2. pozíció (a munkadarab megfogása):

Itt a TCP és a kocka megfogó pontja egybeesik, így $\underline{\underline{H}}_{st}^{TCP}(2)$ az előző ábra szerint:

$$\underline{\underline{H}}_{st}^{TCP}(2) = \left[\begin{array}{ccc:c} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{49}$$

a képlet pedig változatlan marad (47. egyenlet):

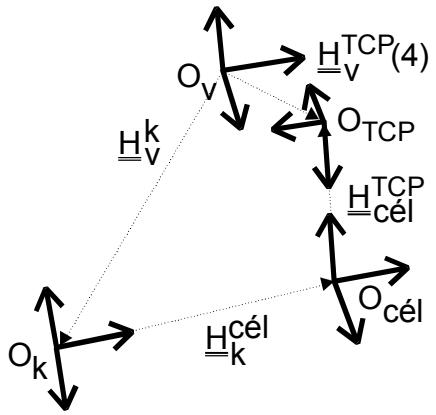
$$\underline{\underline{H}}_{v}^{TCP}(2) = \underline{\underline{H}}_{v}^{ad} \cdot \underline{\underline{H}}_{ad}^{st} \cdot \underline{\underline{H}}_{st}^{TCP}(2). \tag{50}$$

- 3. pozíció (a munkadarab elemelése az adagolótól):

Ez teljesen megegyezik az 1. pozícióval:

$$\underline{\underline{H}}_{v}^{TCP}(3) = \underline{\underline{H}}_{v}^{TCP}(1). \tag{51}$$

- 4. pozíció (a robot elfordult a paletta fölé)



38. ábra

$$\underline{\underline{H}}_{v}^{TCP}(4) = \underline{\underline{H}}_{v}^{k} \cdot \underline{\underline{H}}_{k}^{cél} \cdot \underline{\underline{H}}_{cél}^{TCP}(4). \tag{52}$$

Legyen a megközelítés mértéke itt is 50 mm, így

$$\underline{\underline{H}}_{cél}^{TCP}(4)=\left[\begin{array}{ccc:c} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 50 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \tag{53}$$

- 5. pozíció (a kocka célhelyzetbe rakása)

A cél és TCP koordináta-rendszer origója egybeesik, ezért:

$$\underline{\underline{H}}_{cél}^{TCP}(5)=\left[\begin{array}{ccc:c} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{54}$$

a képlet pedig a 4. pozíció szerinti:

$$\underline{\underline{H}}_{v}^{TCP}(5)=\underline{\underline{H}}_{v}^{k}\cdot\underline{\underline{H}}_{k}^{cél}\cdot\underline{\underline{H}}_{cél}^{TCP}(5). \tag{55}$$

- 6. pozíció (elemelkedés a palettától)

Ez a helyzet a 4. pozícióval teljesen megegyezik.

$$\underline{\underline{H}}_{v}^{TCP}(6)=\underline{\underline{H}}_{v}^{TCP}(4). \tag{56}$$

3. lépés: A robot programozása

Legelőször is a kiszámított pozíciókat kell betáplálnunk a vezérlőegységbe. A fejlettebb vezérlő számítógépek több száz térpont bevitelét is megengedik. A TCP helyzeteit általában nem transzformációs mátrix, hanem x-y-z (pozíció) és roll-pitch-yaw (orientáció) számhármasok alakjában kell betáplálnunk. A konverzióra már láthattunk példát az 6.2.2. fejezetben.

Most nézzünk először egy egyszerűbb programot az előbbi mozgássorozat elvégzésére, majd egy bonyolultabbat, amely nem csak egyetlen kockát mozgat, hanem a paletta rácspontjainak kiszámításával az összes kockát megfelelő sorrendben átpakolja. Mindkettő egy széles körűen használt magas szintű programnyelven, MELFA-BASIC-ben íródott.

- Az egyszerűbb program

A programban olyan utasítások szerepelnek, amelyek a robot elemi mozgásait biztosítják. Nem használunk ciklusokat, illetve magas szintű programnyelvekben biztosított egyéb funkciókat. Ez a legegyszerűbb robot-programozási mód. A fent kiszámított pozíciók közül P1-től P4-ig van szükségünk, ezeket előre betápláltuk a számítógépbe. Az egyes utasítások jelentése a következő:

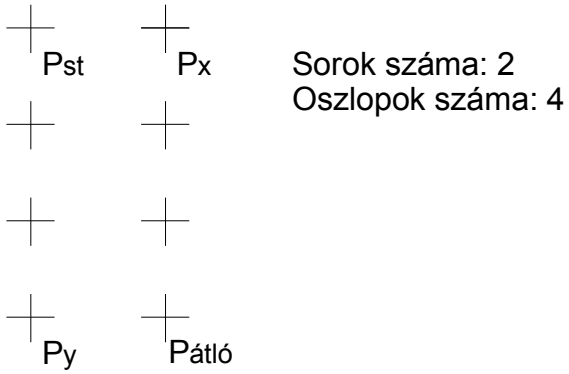
DEF: DEFinition, változó definiálása
HOPEN: Hand OPEN, kéz kinyitása
HCLOSE: Hand CLOSE, kéz összezárása
MOV: MOVe, mozgás csukló interpolációval
MVS: MoVe Straight, egyenes vonalú mozgás
END: END, befejezés

```
5     DEF P1,P2,P3,P4   ' ** a betáplált pontok definiálása
10    HOPEN 1           ' ** a(z 1.számú) kéz kinyitása
20    MOV P1            ' ** TCP mozgatása a P1 pozícióba
30    MVS P2            ' ** TCP egyenes vonalú mozgatása P2-be
40    HCLOSE 1          ' ** a kéz összeszorítása (megfogás)
50    MVS P1            ' ** a kocka kiemelése
60    MOV P3            ' ** át a paletta fölé
70    MVS P4            ' ** a kocka lyukba helyezése
80    HOPEN 1           ' ** a kocka elengedése
90    MVS P3            ' ** eltávolodás a palettától
100   END
```

- A bonyolultabb program

Mivel a robotmanipulációs feladatok során gyakran találkozunk palettákkal (rácsszerűen elhelyezendő alkatrészekkel), a MELFA-BASIC programnyelvben külön utasítások segítik ezek használatát. Például nemcsak egyszerű változókat, térpontokat, hanem palettákat is definiálhatunk a következő módon (a 39. ábra jelöléseivel, ahol a P-vel jelölt térpontokat a pályapontokhoz hasonlóan előre definiáljuk egy x,y,z számhármassal és tetszőleges orientációval):

```
DEF PAL     <paletta száma>,Pst,Px,Py,Pátló,2,4,<sorrend>
```



39. ábra

A paletta 4. pontjára pedig egyszerűen utalhatunk:

```
point_of_palette=PLT <paletta száma>,4
```

A MELFA-BASIC-ben utasíthatjuk a robotot arra is, hogy bizonyos mértékben közelítsen meg egy pontot:

```
MVS P1,50          ' ** P1 megközelítése 50mm-re, egyenes vonalban
```

Ezen felül még találkozhatunk a magas szintű nyelvekben megszokott ciklusokkal, függvényekkel, stb. Mindezen tulajdonságok jelentősen megkönnyítik a robot programozójának a munkáját. Lássuk hát a teljes programot!

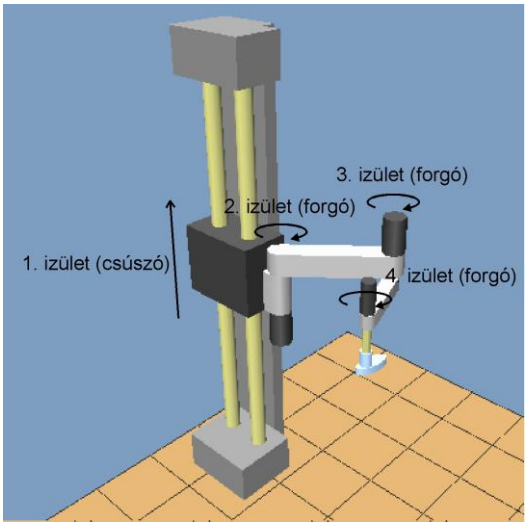
```
10    DEF PLT 1,PST,PX,PY,PATL,2,4,1 ' ** A paletta definiálása
20    DEF INTE i              ' ** Ciklusváltozó definiálása
30    DEF POS PPaletta        ' ** Paletta pozíció definiálása
40    HOPEN 1                 ' ** Kéz nyitás
50    FOR i=1 TO 8            ' ** 8 kockát fog a robot mozgatni
60    MOV P2,-50              ' ** Az adagolóban levő legalsó
                                  kocka fölé helyezkedés 50 mm-re
70    MVS P2                  ' ** Lineáris mozgás a megfogó
                                  pontba
80    HCLOSE 1                ' ** A kocka megfogása
90    MVS P2,-50              ' ** Felemelés
100   PPaletta = PLT 1,i      ' ** Az első rácspont számítása
110   MOV PPaletta,-50        ' ** Átfordulás a paletta fölé
120   MVS PPaletta            ' ** Rácspontra mozgatás
130   HOPEN 1                 ' ** Rácspontra helyezés
140   MVS PPaletta,-50        ' ** Az üres megfogó felemelkedik
150   NEXT i                  ' ** FOR ciklus vége
160   END
```

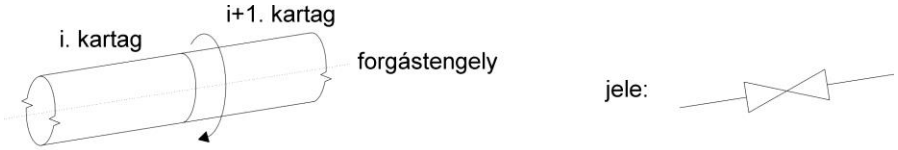
7. A robotkar geometriája

Az eddigiekben leírtuk, hogyan határozhatja meg a robotot beprogramozó szakember a kívánt feladatnak megfelelő pályapontokat és a TCP pontos mozgását. Ezek után a robot vezérlőegységén van a sor, hogy a fenti információkból „levezényelje” a robot egyes részeinek összehangolt működését. A vezérlési feladat megtervezéséhez mindenek előtt szükségünk van a robotkar geometriai felépítésének egzakt leírására.

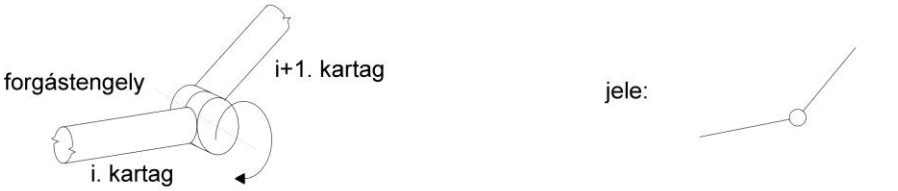
A robotkarok az emberi karhoz hasonlóan izületek és kartagok láncából állnak. Az izületek forgó vagy csúszó mozgásra képesek, és mindegyikük közvetlen kapcsolatban van a robothajtás egy mozgató elemével (pl. motorral). A forgó izületeknek két fajtája van: csavaró illetve billenő (40. ábra). A robot mozgatása vagy megfelelő helyzetbe való állítása gyakorlatilag az izületek pozíciójának (forgó izületeknél az elfordulás szögének, csúszónál az elmozdulás mértékének) a szabályozását jelenti.

Egy adott pozícióban levő tárgy tetszőleges irányból történő megfogásához 6 szabadságfokú robotkar szükséges (3 térkoordináta + 3 orientációs szög). Mindez legalább hat, egymástól független izületet igényel. Ha egy manipulációs feladat nem követeli meg a munkadarab tetszés szerinti forgatását, akkor elég 5 vagy négy izületből álló kar munkába állítása is (ld. 40. ábra).





csavaró izület



billenő izület

csúszó izület

40. ábra

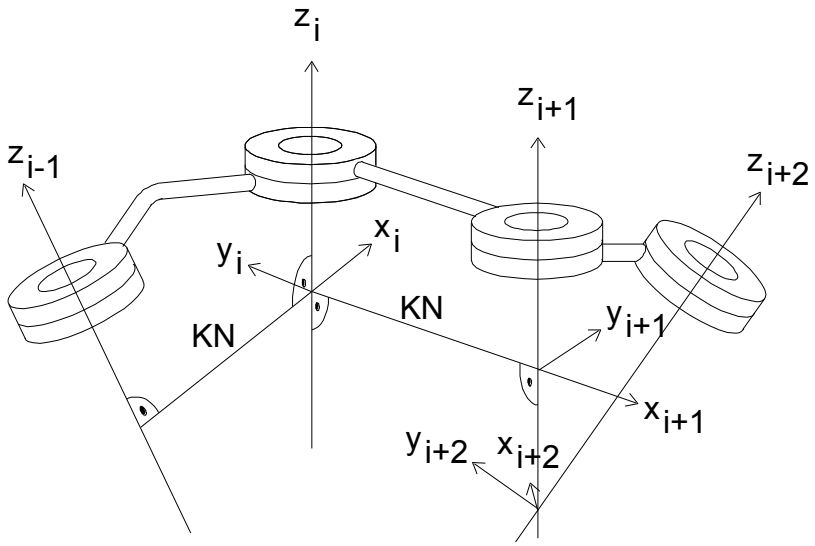
7.1 A Denavit-Hartenberg transzformáció

A számítások egységessé tétele miatt a robotkar geometriájának leírásánál is a homogén transzformációs módszer használata javasolt. Ha minden egyes izülethez egy-egy koordináta-rendszert rendelünk, és sorra leírjuk az egymást követő koordináta-rendszerek közötti transzformációs mátrixokat (vagyis hogyan áll az i. izület az i-1-hez képest), akkor ezzel az egész kar pozícióját megadhatjuk. Fontos, hogy a koordináta-rendszerek hozzárendelése mindig ugyanazon elv szerint történjen, és hogy az izületi szögek és – csúszó izületek esetén – az elmozdulások a lehető legegyszerűbb formában szerepeljenek a transzformációs mátrixokban. Célszerű továbbá olyan módszert keresnünk, amellyel a robot geometriája szinte „ránézésre" mátrixos formába írható. Mindezen elveket egyesíti a Denavit-Hartenberg féle leírási mód:

1. lépés: A koordináta-rendszerek izületekhez rendelése

A hozzárendelésnél a következő szabályokat kell követni (41. ábra):

- z_i tengely legyen az i. izület forgás-tengelye, csúszó izület esetén az elmozdulás tengelye.
- x_i tengely a z_{i-1} és a z_i tengely közös normálisának irányába mutasson. (A közös normális a mindkét térbeli egyenesre merőleges szakasz)
- Az i-ik koordináta-rendszer origóját az (i-1)-ik és i-ik izületi tengely közös normálisának és az i-ik izület tengelyének metszéspontjába helyezzük.
- Párhuzamos forgástengelyek esetén végtelen sok közös normális állítható. Ebben az esetben azt a közös normálist választjuk, amelyik a megelőző izülethez rendelt koordináta-rendszer origóján halad át.
- Egymást metsző tengelyeknél a koordináta-rendszer origója a tengelyek metszéspontja, az x_i tengely irányultsága pedig a $\underline{z}_{i-1} \times \underline{z}_i$ vektoriális szorzat által eredményezett vektorral párhuzamos.

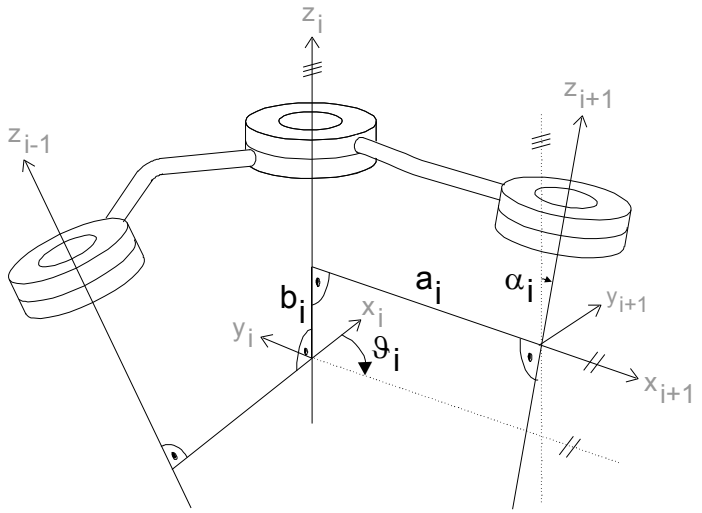


41. ábra

2. lépés: A koordináta-rendszerek viszonyának meghatározása

Az (i+1)-ik koordináta-rendszer viszonyát az i-ikhez képest négy értékkel definiáljuk (42. ábra):

- elfordulás z_i vagyis az i-ik izület tengelye körül ϑ_i szöggel,
- elmozdulás z_i mentén b_i távolsággal,
- elmozdulás x_{i+1} mentén a_i távolsággal,
- elfordulás x_{i+1} körül α_i szöggel.



42. ábra

A fenti négy transzformációt leíró mátrixok sorra a következők:

$$\underline{\underline{H}}(\vartheta_i) = \left[\begin{array}{ccc:c} \cos\vartheta_i & -\sin\vartheta_i & 0 & 0 \\ \sin\vartheta_i & \cos\vartheta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{57}$$

$$\underline{\underline{H}}(b_i) = \left[\begin{array}{ccc:c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b_i \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{58}$$

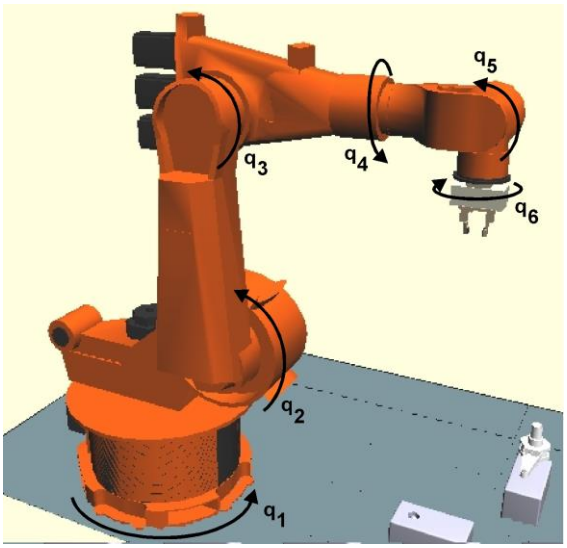
$$\underline{\underline{H}}(a_i) = \left[\begin{array}{ccc:c} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{59}$$

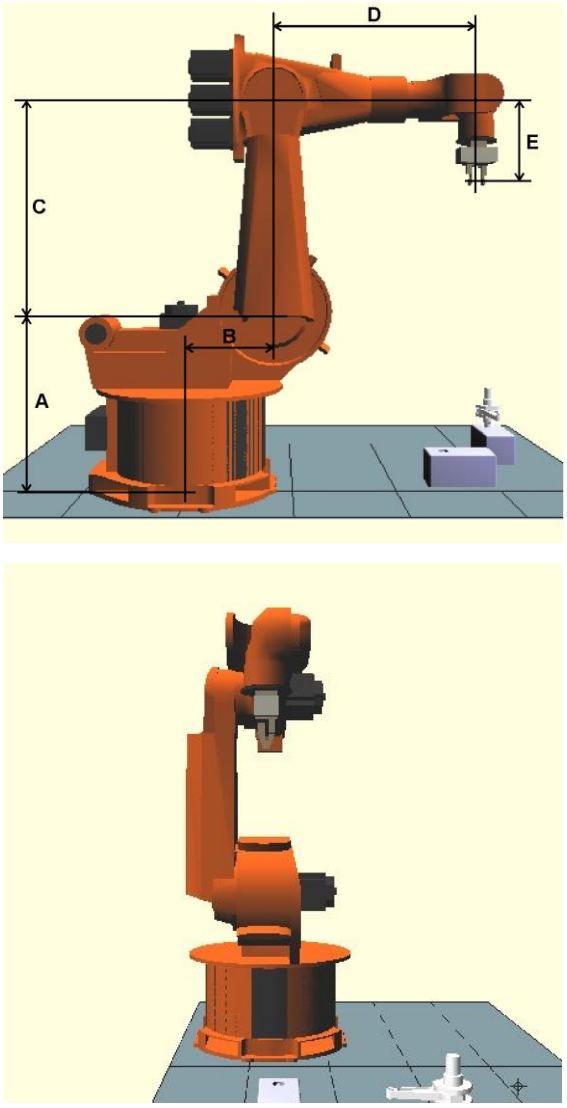
$$\underline{\underline{H}}(\alpha_i) = \left[\begin{array}{ccc:c} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \tag{60}$$

Ezek eredő transzformációja pedig:

$$\underline{\underline{H}}_i^{i+1} = \underline{\underline{H}}(\vartheta_i)\cdot\underline{\underline{H}}(b_i)\cdot\underline{\underline{H}}(a_i)\cdot\underline{\underline{H}}(\alpha_i) = \left[\begin{array}{ccc|c} \cos\vartheta_i & -\sin\vartheta_i\cos\alpha_i & \sin\vartheta_i\sin\alpha_i & a_i\cos\vartheta_i \\ \sin\vartheta_i & \cos\vartheta_i\cos\alpha_i & -\cos\vartheta_i\sin\alpha_i & a_i\sin\vartheta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & b_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \quad (61)$$

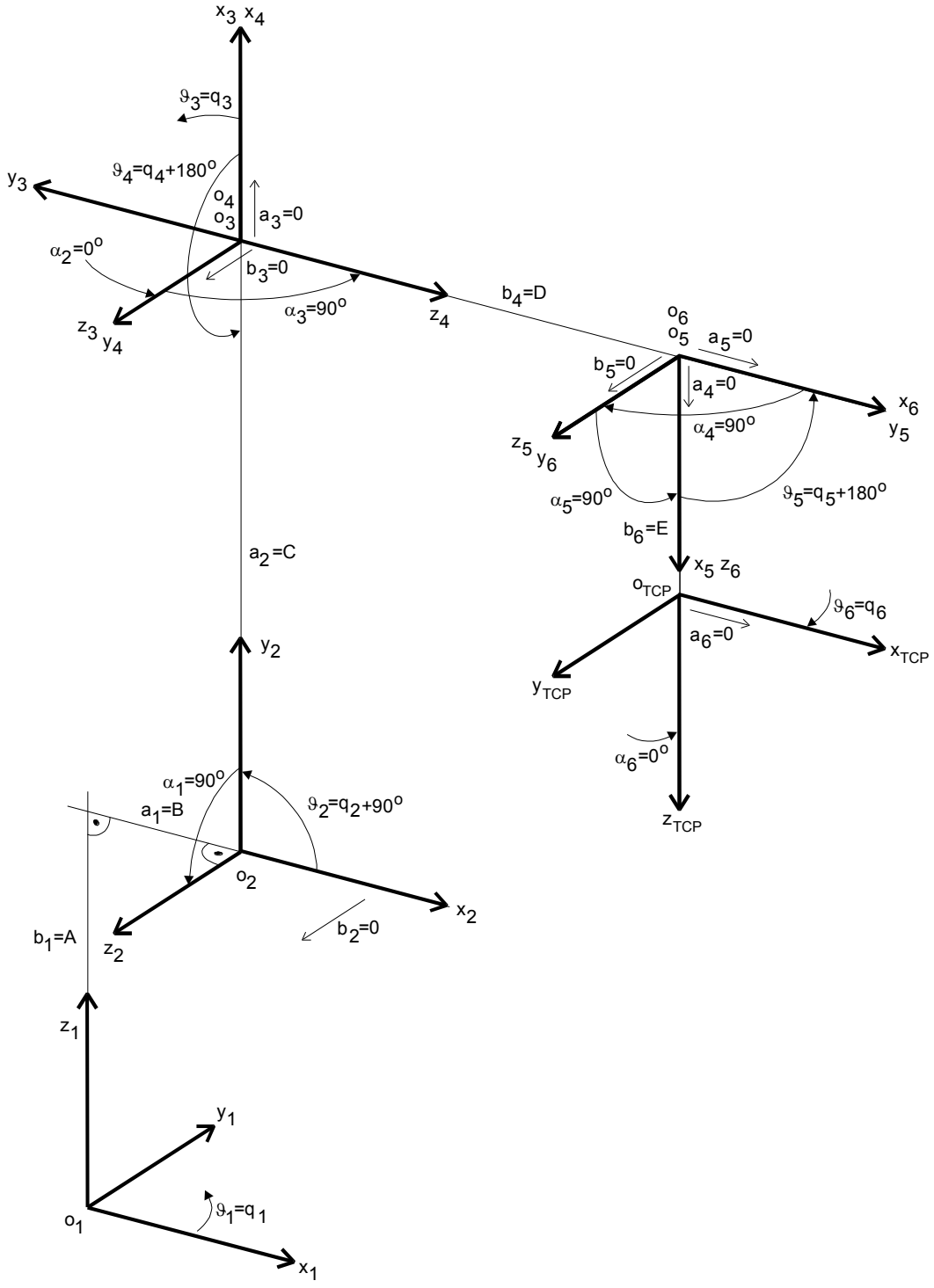
Világosítsuk meg az eddigieket egy példával! Rendeljünk a 43. ábrán látható KUKA KR125 típusú robothoz koordináta-rendszereket, és írjuk fel a koordináta-rendszerek viszonyát! q_1-q_6 izületi változók az egyes izületek elfordulási szögét jelölik a nyíl irányában. Értékük az ábrán bemutatott helyzetben rendre nulla, kivéve q_5-öt, amely –90°.





43. ábra

A koordináta-rendszerek és a transzformációs értékek – a fenti perspektivikus kép nézőpontjából – a 44. ábra szerint alakulnak.



44. ábra

Láthatjuk, hogy a ϑ_i szögek és a q_i izületi szögek – egy-egy konstans nagyságú eltéréstől eltekintve – megegyeznek. Az α_i, a_i, b_i változók egyszerű kapcsolatba hozhatók a robot geometriai felépítésének jellegzetes mérőszámaival. Mindez a látszólag bonyolult ábra ellenére viszonylag

egyszerű transzformációs mátrixokat eredményez. Az 1. táblázatban soroltuk fel a kapott összefüggéseket.

i	1	2	3	4	5	6
ϑ_i	q_1	q_2+90°	q_3	q_4+180°	q_5+180°	q_6
b_i	A	0	0	D	0	E
a_i	B	C	0	0	0	0
α_i	90°	0	90°	90°	90°	0

1. táblázat

A táblázat elemeit a (61) egyenletbe behelyettesítve felírhatjuk a mátrixokat. (62-65) azonosságokat felhasználva:

$$\sin(q+90°) = \cos q, \tag{62}$$

$$\cos(q+90°) = -\sin q, \tag{63}$$

$$\sin(q+180°) = -\sin q, \tag{64}$$

$$\cos(q+180°) = -\cos q, \tag{65}$$

$$\underline{\underline{H}}_1^2 = \left[\begin{array}{ccc:c} \cos\vartheta_1 & 0 & \sin\vartheta_1 & B\cos\vartheta_1 \\ \sin\vartheta_1 & 0 & -\cos\vartheta_1 & B\sin\vartheta_1 \\ 0 & 1 & 0 & A \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} \cos q_1 & 0 & \sin q_1 & B\cos q_1 \\ \sin q_1 & 0 & -\cos q_1 & B\sin q_1 \\ 0 & 1 & 0 & A \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{66}$$

$$\underline{\underline{H}}_2^3 = \left[\begin{array}{ccc:c} \cos\vartheta_2 & -\sin\vartheta_2 & 0 & C\cos\vartheta_2 \\ \sin\vartheta_2 & \cos\vartheta_2 & 0 & C\sin\vartheta_2 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} -\sin q_2 & -\cos q_2 & 0 & -C\sin q_2 \\ \cos q_2 & -\sin q_2 & 0 & C\cos q_2 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{67}$$

$$\underline{\underline{H}}_3^4 = \left[\begin{array}{ccc:c} \cos\vartheta_3 & 0 & \sin\vartheta_3 & 0 \\ \sin\vartheta_3 & 0 & -\cos\vartheta_3 & 0 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} \cos q_3 & 0 & \sin q_3 & 0 \\ \sin q_3 & 0 & -\cos q_3 & 0 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{68}$$

$$\underline{\underline{H}}_4^5 = \left[\begin{array}{ccc:c} \cos\vartheta_4 & 0 & \sin\vartheta_4 & 0 \\ \sin\vartheta_4 & 0 & -\cos\vartheta_4 & 0 \\ 0 & 1 & 0 & D \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} -\cos q_4 & 0 & -\sin q_4 & 0 \\ -\sin q_4 & 0 & \cos q_4 & 0 \\ 0 & 1 & 0 & D \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{69}$$

$$\underline{\underline{H}}_5^6 = \left[\begin{array}{ccc:c} \cos\vartheta_5 & 0 & \sin\vartheta_5 & 0 \\ \sin\vartheta_5 & 0 & -\cos\vartheta_5 & 0 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} -\cos q_5 & 0 & -\sin q_5 & 0 \\ -\sin q_5 & 0 & \cos q_5 & 0 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \tag{70}$$

$$\underline{\underline{H}}_6^{TCP} = \left[\begin{array}{ccc:c} \cos\vartheta_6 & -\sin\vartheta_6 & 0 & 0 \\ \sin\vartheta_6 & \cos\vartheta_6 & 0 & 0 \\ 0 & 0 & 1 & E \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} \cos q_6 & -\sin q_6 & 0 & 0 \\ \sin q_6 & \cos q_6 & 0 & 0 \\ 0 & 0 & 1 & E \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \tag{71}$$

7.2 A robot-geometria direkt feladata

A vezérlőegység legalapvetőbb számítási feladata az, hogy az izületi szögekből és elmozdulásokból meghatározza a TCP helyzetét (a robot alap-koordinátarendszerében). Ezt nevezzük *a robot-geometria direkt feladatá*nak. A gyakorlatban megkívánt számítás azonban ennek éppen a fordítottja, vagyis az, hogy a TCP helyzetéből meghatározzuk az izületi szögeket és elmozdulásokat, s ezáltal a robotot a megfelelő pozícióba állíthassuk. Ez *a robot-geometria inverz feladata*.

A direkt feladat megoldásához rendelkezésünkre állnak az egymást követő izületi koordináta-rendszerek viszonyát leíró transzformációs mátrixok. A relatív transzformációkat tárgyaló részben tanultak alapján könnyedén felírható a TCP-hoz tartozó koordináta-rendszer helyzete:

$$\underline{\underline{H}}_1^{TCP} = \underline{\underline{H}}_1^2 \cdot \underline{\underline{H}}_2^3 \cdot \underline{\underline{H}}_3^4 \cdot \underline{\underline{H}}_4^5 \cdot \underline{\underline{H}}_5^6 \cdot \underline{\underline{H}}_6^{TCP}, \tag{72}$$

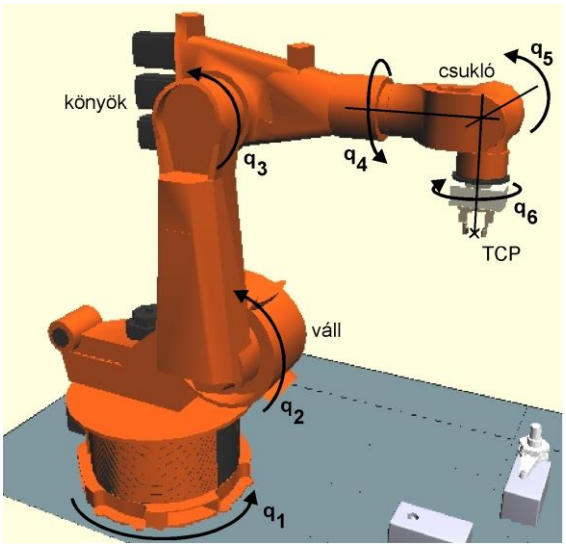
ahol feltételeztük, hogy az alap-koordinátarendszer egybeesik az első izülethez rendelt koordináta-rendszerrel, és a robotkar 6 szabadságfokú.

A fenti művelet 60 szorzást és 45 összeadást igényel. Ezt a számítási mennyiséget a műveletek megfelelő csoportosításával csökkenteni lehet.

7.3 A robot-geometria inverz feladata

Többféle módszer létezik az izületi szögek meghatározásához a TCP helyzetének ismerete alapján (inverz feladat). A számítások sokszor bonyolultak, és a kapott eredmények nem egyértelműek: van, hogy ugyanazt a TCP helyzetet az izületi szögek más-más kombinációi is előállíthatják, s ezek közül esetleg a munkatér adottságai alapján kell választanunk. Általánosan használható megoldási módok nemigen léteznek, inkább csak sémák, útmutatók. Minden egyes robotra meg kell találni a felépítéséből adódó legegyszerűbb számítást; éppen ezért a következőkben egy konkrét példát ismertetünk.

Az olyan felépítésű robotkarok, mint a már bemutatott KUKA KR125 különleges adottsága az, hogy az utolsó három izület forgástengelye egy pontban, a csuklóban metszi egymást (45. ábra). Ez megkönnyíti a dolgunkat: a hat ismeretlenes egyenletrendszert két darab három ismeretlenesre tudjuk bontani. A csukló koordinátáit ugyanis a TCP helyzete egyértelműen meghatározza; a csukló pozíciója pedig csak a q_1, q_2, q_3 izületi változóktól függ. A másik három változó számítása az orientációs szögek számításához hasonló: a csukló „tövéhez” képest úgy kell beállítanunk q_4,q_5,q_6-ot, hogy a megfogó a kívánt irányba nézzen.



45. ábra

A robotkar felépítése egy további könnyebbséggel szolgál: a csukló vízszintes síkú pozíciójából közvetlenül számítható q_1. A csukló függőleges pozícióját q_2 és q_3 összege határozza meg.

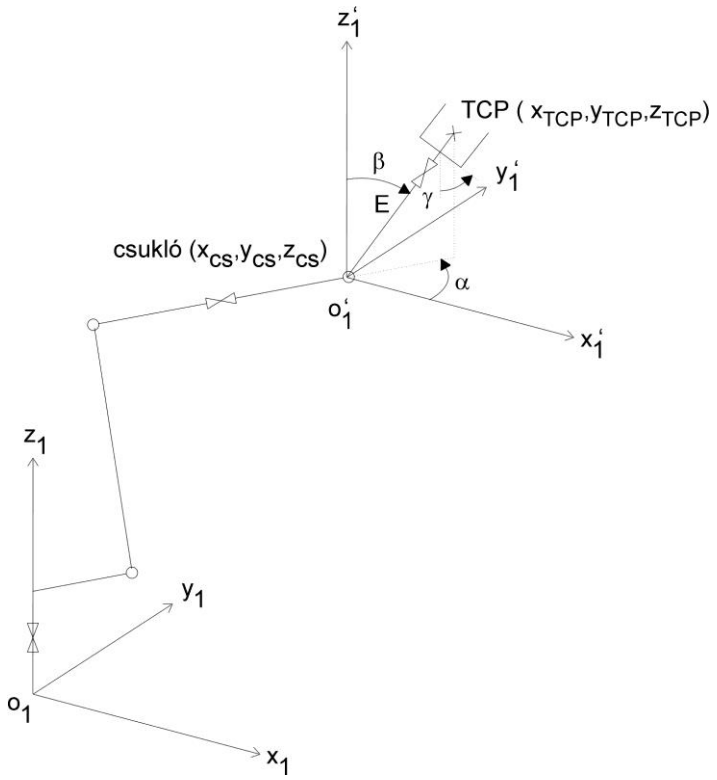
1. lépés: a csuklópozíció számítása

A 46. ábra mutatja a TCP és a csukló viszonyát az orientációs szögek függvényében. A rajz alapján x_{cs},y_{cs},z_{cs} csuklókoordináták a következőképp számíthatók:

$$x_{cs} = x_{TCP} - E\cos\alpha\sin\beta, \tag{73}$$

$$y_{cs} = y_{TCP} - E\sin\alpha\sin\beta, \tag{74}$$

$$z_{cs} = z_{TCP} - E\cos\beta. \tag{75}$$



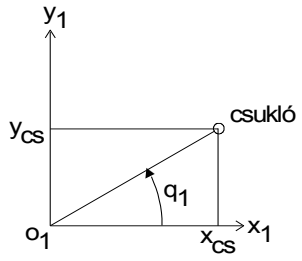
46. ábra

2. lépés: q1 számítása

A csuklónak az alap-koordinátarendszer z tengelye körüli – vízszintes síkban történő – elfordulása kizárólag q_1 függvénye. A 47. ábra szerint:

$$q_1 = \arctan \frac{y_{cs}}{x_{cs}}, \text{ ha a csukló az I. vagy a IV. térnegyedben van,} \tag{76}$$

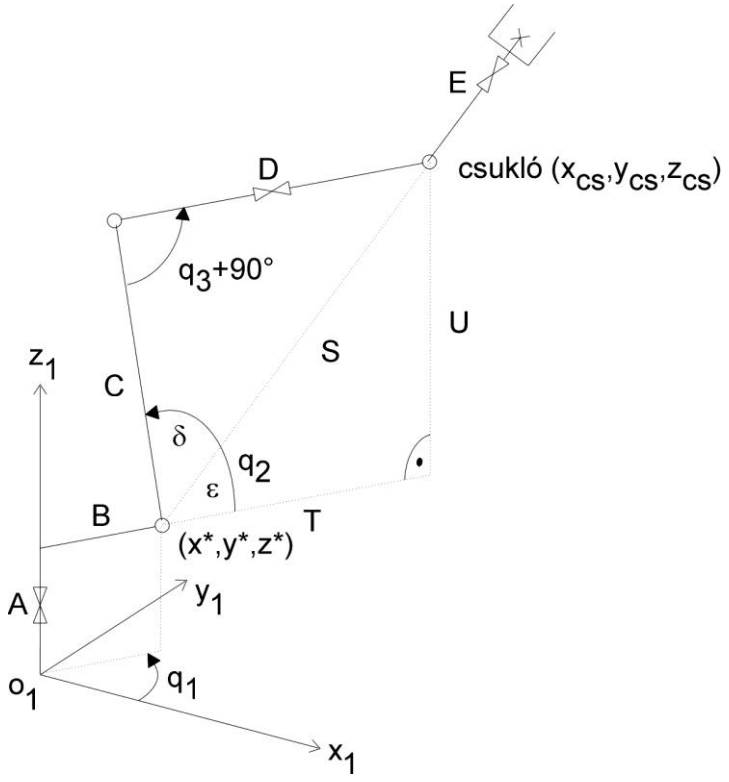
$$q_1 = 180^\circ + \arctan \frac{y_{cs}}{x_{cs}}, \text{ ha a csukló a II. illetve III. térnegyedben tartózkodik.} \tag{77}$$



47. ábra

3. lépés: q_2 és q_3 számítása

Végigböngészve a 48. ábrát, q_2 és q_3 geometriai összefüggések alapján kiszámíthatók.



48. ábra

q_3 a cosinus-tétel szerint nyerhető:

$$S^2 = C^2 + D^2 - 2CD\cos(q_3 + 90^\circ), \text{ ahol} \tag{78}$$

$$S^2 = (x_{cs} - x^*)^2 + (y_{cs} - y^*)^2 + (z_{cs} - z^*)^2, \text{ továbbá} \tag{79}$$

$$x^* = B\cos q_1, \tag{80}$$

$$y^* = B\sin q_1, \tag{81}$$

$$z^* = A. \tag{82}$$

q_2 két szögre bontható:

$$q_2 = \delta + \varepsilon. \tag{83}$$

δ kiszámítása q_3-hoz hasonlóan:

$$D^2 = C^2 + S^2 - 2CS\cos\delta, \tag{84}$$

ε értéke pedig:

$\varepsilon = \arctan\frac{U}{T}$, illetve (85)

$\varepsilon = 180^\circ + \arctan\frac{U}{T}$, ha a csukló a B kartag fölé került (T negatív). (86)

A még hiányzó értékek:

$U = z_{cs} - z^*$, (87)

$T = \sqrt{x_{cs}^2 + y_{cs}^2} - \sqrt{(x^*)^2 + (y^*)^2}$. (88)

4. lépés: q_4, q_5 és q_6 kiszámítása

A robot utolsó három izületi tengelye egy pontban metszi egymást, így a

$$\underline{\underline{H}}_4^{TCP} = \underline{\underline{H}}_4^5 \cdot \underline{\underline{H}}_5^6 \cdot \underline{\underline{H}}_6^{TCP} \tag{89}$$

mátrix formailag azonos lesz azokkal a transzformációs mátrixokkal, amelyeket a roll-pitch-yaw illetve Euler-szögekkel történő forgatás-sorozatok eredményeként kaptunk (a 69, 70, 71 egyenletekből, az eredményt vö. a 32. egyenlettel):

$$\underline{\underline{H}}_4^{TCP} = \underline{\underline{H}}_4^5 \cdot \underline{\underline{H}}_5^6 \cdot \underline{\underline{H}}_6^{TCP} =$$

$$= \left[\begin{array}{ccc|c} \cos q_4 \cos q_5 \cos q_6 - \sin q_4 \sin q_6 & -\cos q_4 \cos q_5 \sin q_6 - \sin q_4 \cos q_6 & \cos q_4 \sin q_5 & E \cos q_4 \sin q_5 \\ \sin q_4 \cos q_5 \cos q_6 + \cos q_4 \sin q_6 & -\sin q_4 \cos q_5 \sin q_6 - \cos q_4 \cos q_6 & \sin q_4 \sin q_5 & E \sin q_4 \sin q_5 \\ -\sin q_5 \cos q_6 & \sin q_5 \sin q_6 & \cos q_5 & E \cos q_5 + D \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{90}$$

Mivel q_1, q_2, illetve q_3, s ezáltal

$$\underline{\underline{H}}_1^4 = \underline{\underline{H}}_1^2 \cdot \underline{\underline{H}}_2^3 \cdot \underline{\underline{H}}_3^4 \tag{91}$$

ismert, $\underline{\underline{H}}_4^{TCP}$ a következők szerint kiszámítható (a 72. egyenletből kiindulva):

$$\underline{\underline{H}}_1^{TCP} = \underline{\underline{H}}_1^2 \cdot \underline{\underline{H}}_2^3 \cdot \underline{\underline{H}}_3^4 \cdot \underline{\underline{H}}_4^5 \cdot \underline{\underline{H}}_5^6 \cdot \underline{\underline{H}}_6^{TCP} = \underline{\underline{H}}_1^4 \cdot \underline{\underline{H}}_4^{TCP}, \tag{92}$$

$$\underline{\underline{H}}_4^{TCP} = \left(\underline{\underline{H}}_1^4\right)^{-1} \cdot \underline{\underline{H}}_1^{TCP}. \tag{93}$$

Tehát nincs más dolgunk, mint $\underline{\underline{H}}_4^{TCP}$ egyes elemeit (93)-ból meghatározni, és a kapott mátrixból az 6.2.2. példában eljártak szerint q_4-et, q_5-öt és q_6-ot kiszámítani.

8. Pályavezérlés

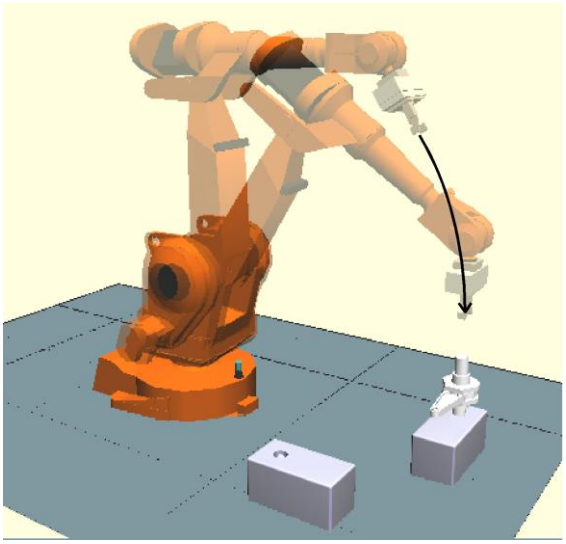
A következőkben arról lesz szó, hogy a robot vezérlőegysége miképp vezeti végig a TCP-t a megkívánt pályán. A számítások kiindulási adatai a pálya sarokpontjainak koordinátái illetve orientációs szögei, az eredmények pedig az egyes izületek szögpozíció-, szögsebesség- illetve szöggyorsulás-görbéi a pálya mentén. Ezek alapján határozhatók meg a robot mozgatásához szükséges pillanatnyi hajtónyomatékok; a robotok hajtásszabályozásának a feladata ezeknek a megfelelő szinten tartása és a hajtásokat közvetlenül működtető áram-, feszültségszintek, vezérlőimpulzusok, stb. előállítása.

8.1 PTP irányítás

PTP (Point-To-Point) irányítás során csak a pálya tartópontjait definiáljuk, a közöttük bejárt pályagörbét nem. A vezérlőegység számára az egyedüli szempont az, hogy a robot a lehető legrövidebb idő alatt jusson el a kezdőhelyzetből a végállapotba. Ez a legegyszerűbben úgy oldható meg, ha az összes izületet a maximális sebességgel az új pozícióba mozgatjuk. Mivel az izületeknek különböző nagyságú elfordulásokat vagy utakat kell megtenniük, eltérő sebességgel és gyorsulással – hiszen az első motornak a robot egészét kell mozgatnia, az utolsónak csak a szerszámot, így méretezésük során más-más szempontok érvényesülnek –, az izületek nem

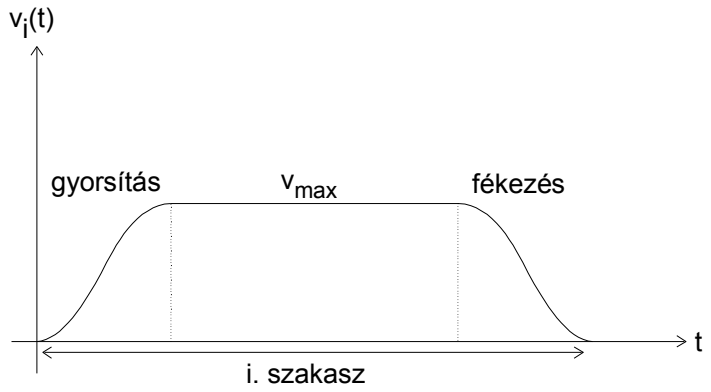
egyszerre fejezik be a munkájukat. A pályagörbe az izületek egymás utáni leállásakor töréseket szenved, ezért előre nem tervezhető.

A PTP vezérlésnek fejlettebb változata az, amelynél az gyorsabb izületek sebességét úgy állítják be, hogy a leglassabb izülettel együtt végezzenek. Ezt a módszert *szinkronizált izület-interpoláció*nak (*synchronous PTP interpolation*) nevezzük. A robot mozgása így mentesül a felesleges rezgésektől. (49. ábra)



49. ábra

PTP irányításnál a vezérlőegység számításigénye nem túl nagy, mivel nincs szükség az inverz geometriai feladat megoldására. Az egyes izületek sebesség-görbéje az 50. ábra szerint alakul. A gyorsító és fékező szakasz meredeksége a hajtásra jellemző gyorsulások függvénye.



50. ábra

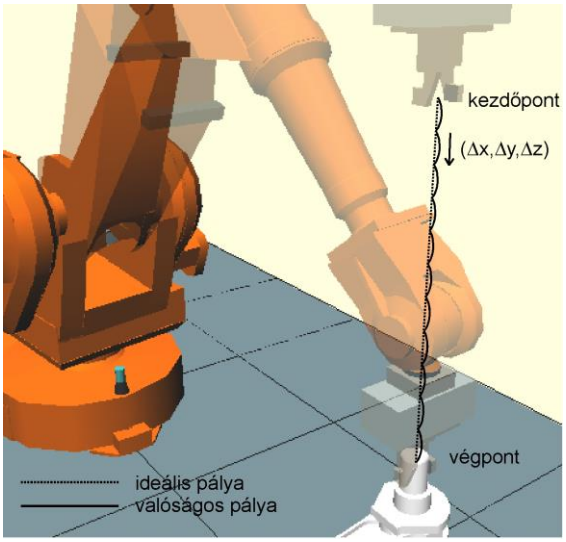
8.2 Lineáris pályairányítás

Lineáris pályairányítást alkalmazva a TCP egyenes vonalban mozog a kezdőpontból a végpontba, és eközben az orientációja is egyenletesen változik. A számítási módszer az ún. *megfogó-állapot vagy Descartes-típusú interpoláció (*angolul *Cartesian interpolation*):

- A szakasz megtételéhez szükséges időt apró Δt intervallumokra bontjuk (*Cartesian cycle-time, ciklusidő*),
- kiszámítjuk a TCP Δt idő alatti $(\Delta x, \Delta y, \Delta z)$ elmozdulását,
- a TCP pillanatnyi helyzetéből és a $(\Delta x, \Delta y, \Delta z)$ inkremensből az inverz geometriai feladat megoldásával meghatározzuk a $\Delta \underline{q} = [\Delta q_1 \quad \Delta q_2 \quad ...]^T$ izületi szögváltozásokat illetve elmozdulásokat,
- az izületeket a számolt pozícióba mozgatjuk (PTP interpolációval),
- a fenti lépéseket addig folytatjuk, míg a célhoz nem érünk.

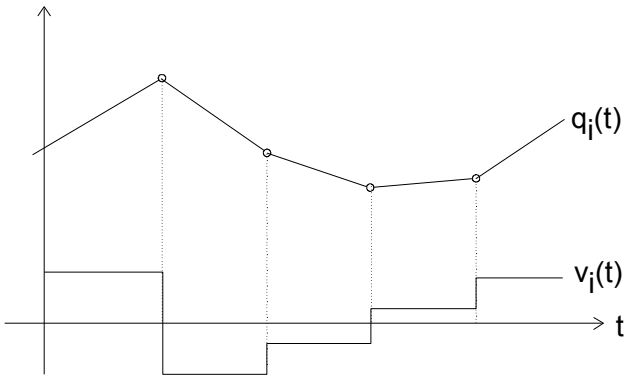
A megfogó-állapot interpolációval bejárt szakasz csak elvben lehet teljesen egyenes. A Δt intervallumonként kiszámolt trajektória-pontok között a TCP izületi interpolációval halad, s ez „döcögőssé" teszi az utat (az 51. ábra mindezt kissé elnagyolva mutatja). A pályabejárás minőségén kis ciklusidő választásával segíthetünk. A ciklusidő csökkentését a vezérlő számítógép

kapacitása korlátozhatja, hiszen minden egyes ciklusban az inverz geometriai feladatot is meg kell oldania, s ez hatalmas számítási igényt támaszt.



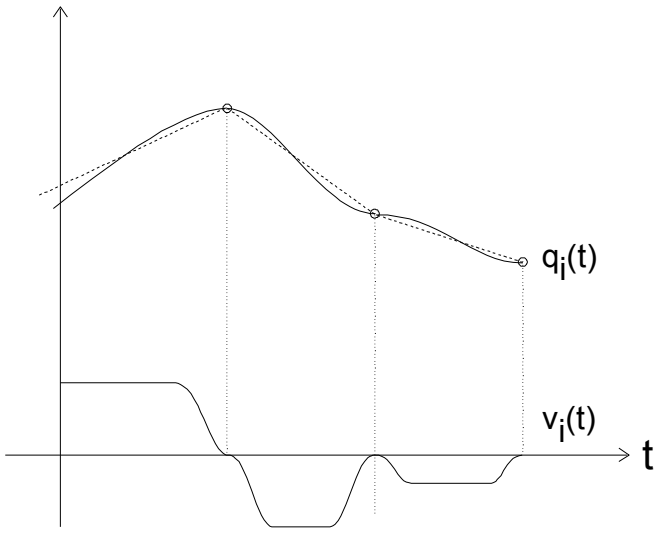
51. ábra

Az 52. ábra egy izület megfogó-állapot interpolációval számított szögváltozását mutatja. Látható, hogy az egyes trajektória-pontok között más és más a görbe meredeksége, vagyis az izület szögsebessége. A trajektória-pontokon a szögsebességek átmenet nélkül változnak meg: a sebesség-görbe ezeken a helyeken töréssel rendelkezik, ez pedig végtelen mértékű szöggyorsulást feltételez. A valóságban ez nem lehetséges, a hajtások által leadott nyomaték és az ezzel mozgatott tömeg minden egyes izületre meghatározzák a legnagyobb gyorsulást.



52. ábra

Az ellentmondás leküzdésére két megoldást alkalmazhatunk. Az egyszerűbb az, hogyha minden egyes trajektória-ponton megállítjuk a robotot, és az izületeket az új sebességre gyorsítjuk (53. ábra). Ez gazdaságossági szempontból nem javasolt, mert lassú lesz a TCP mozgása. Néhány olcsóbb, főként oktatórobot dolgozik így.



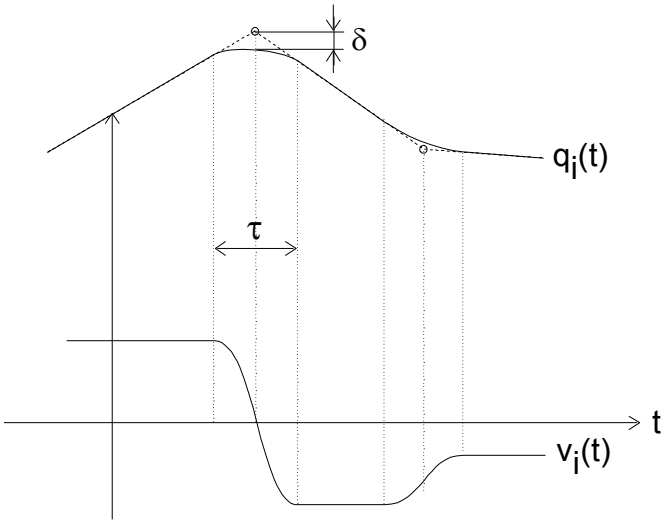
53. ábra

A másik megoldás a következő (54. ábra):

- A maximális izületi gyorsulások figyelembevételével a trajektória-pontok környezetében meghatározzuk azokat a τ átmeneti időket,

amelyek alatt az egyes izületek mozgása az egyik sebességről a másikra változik,

- a kiszámolt idők közül a legnagyobbat vesszük (a leglassabb izületét), s a többi izület gyorsulását is erre az időre számítjuk ki,
- az izületek sebességét a trajektória-pontok előtti $\tau/2$ időpontban a számolt gyorsulásokkal változtatni kezdjük, hogy τ idő múlva már az új sebességgel mozogjanak.

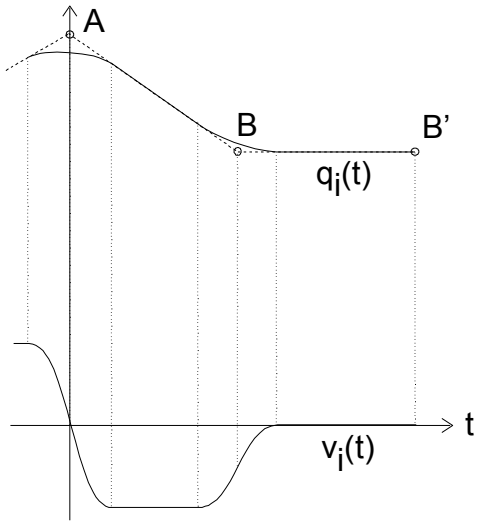


54. ábra

Az ábrán látható, hogy az izületi változók értékei nem érik el az ideális esetben számoltakat a trajektória-pontokon: δ eltéréssel csak megközelítik azokat. A robot egy bonyolult pályán fog az egyenes körül mozogni, egy bizonyos határon belül. A hajtások nyomatékai, s így a legnagyobb gyorsulásaik adottak, ezért ha nagyobbra választjuk a robot mozgási sebességét, a τ átmeneti idők megnőnek, s ezáltal a δ hibák is nagyobbak lesznek. A robot sebessége tehát fordított arányban áll a pályabejárás pontosságával.

A mozgás befejezésekor nem engedhetjük meg, hogy pontatlan pozícióban álljon meg a robot. Ez úgy oldható meg, hogy a legutolsó trajektória-pontot (a célállapotot) megismételjük. Ekkor az izületek sebessége a megengedhető lassulásokkal zérusra csökken, és a robot a kívánt

pozícióban áll meg (55. ábra). Akkor is ezt a módszert követjük, ha a robotot egy bizonyos trajektória-ponton hiba nélkül akarjuk átvezetni.



55. ábra

Bonyolultabb formájú pályákon történő irányítás

Előfordulhat, hogy a robotnak egy ívet, vagy íveket tartalmazó görbét (például autók karosszériaelemeinek a vonalát) kell követnie. Az ehhez szükséges számítások a lineáris pályairányításnál alkalmazottakhoz vezethetők vissza, az elméletek lényeges vonásokban nem különböznek.

8.3 Sebességvezérlés

Bizonyos robottechnikai feladatoknál a szerszám sebességét is állandó értéken kell tartanunk (pl. festésénél). A TCP adott sebességgel és irányban mozog, és orientációja is egyenletesen változik. Az izületek megfelelő mozgása az ún. Jakobi-mátrix segítségével határozható meg:

Ismert a TCP pozíciója és orientációja egy adott pillanatban:

$\underline{z} = [x \quad y \quad z \quad \alpha \quad \beta \quad \gamma]^T$, s ezáltal (94)

$\underline{q} = [q_1 \quad q_2 \quad ...]^T$ izületi változók értékei is, (95)

továbbá a TCP x,y,z tengelyirányú sebességei és a forgástengelyek körüli szögsebességei:

$$\frac{d\underline{z}}{dt} = [v_x \quad v_y \quad v_z \quad \varpi_x \quad \varpi_y \quad \varpi_z]^T . \tag{96}$$

Keressük azokat a

$\frac{d\underline{q}}{dt} = \left[\frac{dq_1}{dt} \quad \frac{dq_2}{dt} \quad ...\right]^T$ izületi sebességeket, amelyek ezt a mozgást megvalósítják. (97)

$\frac{d\underline{z}}{dt}$ és $\frac{d\underline{q}}{dt}$ között a (robot geometriája alapján felírható) $\underline{\underline{J}}(\underline{q})$ Jakobi-mátrix teremt kapcsolatot:

$$\frac{d\underline{z}}{dt} = \underline{\underline{J}}(\underline{q})\frac{d\underline{q}}{dt} . \tag{98}$$

A keresett értékek a Jakobi-mátrix meghatározása után annak invertálásával kaphatók meg:

$$\frac{d\underline{q}}{dt} = \underline{\underline{J}}(\underline{q})^{-1}\frac{d\underline{z}}{dt} . \tag{99}$$

Lássunk erre egy egyszerű példát!

Az 56. ábrán egy kétdimenziós robotkar látható. Ismerjük az izületi szögek pillanatnyi értékét:

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} 45^\circ \\ 45^\circ \end{bmatrix} . \tag{100}$$

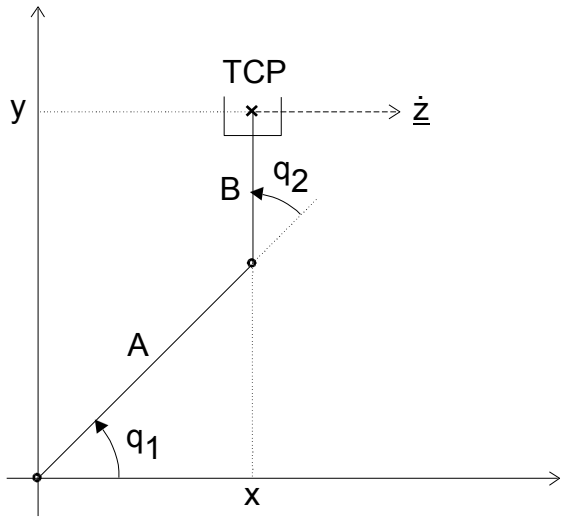
A TCP vízszintes illetve függőleges irányú sebessége a következő:

$$\dot{\underline{z}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \tag{101}$$

tehát a TCP vízszintes irányban egységnyi sebességgel halad.

Mekkora izületi szögsebességek biztosítják ezt?

$$\frac{d\underline{q}}{dt} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = ? \tag{102}$$



56. ábra

1. lépés: a Jakobi mátrix felírása:

Az izületi változók és a TCP koordinátái közötti összefüggések:

$$x = A\cos q_1 + B\cos(q_1 + q_2), \tag{103}$$

$$y = A\sin q_1 + B\sin(q_1 + q_2) \tag{104}$$

Deriváljuk az idő szerint mindkét egyenletet:

$$\dot{x} = -A\sin q_1\dot{q}_1 - B\sin(q_1 + q_2)(\dot{q}_1 + \dot{q}_2), \tag{105}$$

$$\dot{y} = A\cos q_1\dot{q}_1 + B\cos(q_1 + q_2)(\dot{q}_1 + \dot{q}_2), \tag{106}$$

tehát

$$\dot{x} = [-A\sin q_1 - B\sin(q_1 + q_2)]\dot{q}_1 - B\sin(q_1 + q_2)\dot{q}_2, \tag{107}$$

$$\dot{y} = [A\cos q_1 + B\cos(q_1 + q_2)]\dot{q}_1 + B\cos(q_1 + q_2)\dot{q}_2. \tag{108}$$

A Jakobi mátrix ezekből felírható:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix}=\begin{bmatrix}-A\sin q_1-B\sin(q_1+q_2) & -B\sin(q_1+q_2)\\ A\cos q_1+B\cos(q_1+q_2) & B\cos(q_1+q_2)\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}, \tag{109}$$

$$\underline{\underline{J}}(\underline{q})=\begin{bmatrix}-A\sin q_1-B\sin(q_1+q_2) & -B\sin(q_1+q_2)\\ A\cos q_1+B\cos(q_1+q_2) & B\cos(q_1+q_2)\end{bmatrix}. \tag{110}$$

2. lépés: a Jakobi-mátrix kiszámítása az adott q_1, q_2 értékekkel, s a kapott mátrix invertálása:

$$\underline{\underline{J}}(\underline{q})=\begin{bmatrix}-2\sin 45°-\sin 90° & -\sin 90°\\ 2\cos 45°+\cos 90° & \cos 90°\end{bmatrix}=\begin{bmatrix}-\sqrt{2}-1 & -1\\ \sqrt{2} & 0\end{bmatrix}, \tag{111}$$

$$\underline{\underline{J}}(\underline{q})^{-1}=\begin{bmatrix}0 & \dfrac{1}{\sqrt{2}}\\ -1 & -\dfrac{\sqrt{2}+1}{\sqrt{2}}\end{bmatrix}. \tag{112}$$

4. lépés: a szögsebességek meghatározása:

$$\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}=\underline{\underline{J}}(\underline{q})^{-1}\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix}=\begin{bmatrix}0 & \dfrac{1}{\sqrt{2}}\\ -1 & -\dfrac{\sqrt{2}+1}{\sqrt{2}}\end{bmatrix}\cdot\begin{bmatrix}1\\ 0\end{bmatrix}=\begin{bmatrix}0\\ -1\end{bmatrix}, \tag{113}$$

vagyis a 2. izület egységnyi sebességgel forog visszafelé (ami az ábrára tekintve is ellenőrizhető).

Az összefüggésből kiderül, hogy $\underline{\underline{J}}(\underline{q})$ az izületi változók függvénye, ezért a robot különböző pozícióiban más és más lesz az értéke. Éppen ezért, ha egy szakaszon egyenletes sebességgel végig akarjuk vezetni a TCP-t, akkor időről időre újra kell számolnunk a szükséges szögsebességeket. A módszer a lineáris interpolációhoz hasonló:

- Mindenekelőtt meghatározzuk a Jakobi-mátrixot,
- az út megtételéhez szükséges időt apró Δt időintervallumokra osztjuk (*ciklusidő*),

- a kezdőpontban kiszámítjuk a Jakobi-mátrix aktuális értékét, invertáljuk, és meghatározzuk a szükséges szögsebességeket,
- a robotot a kellő sebességre gyorsítjuk, miközben meghatározzuk a robot pozícióját Δt idő elteltével, s erre is meghatározzuk a szögsebességeket az előbbi pont szerint,
- az izületek sebességét a lineáris interpolációnál tárgyaltakhoz hasonlóan az újonnan kiszámított értékekre korrigáljuk,
- majd a számításokat további Δt idők hozzáadásával újra meg újra elvégezzük, és a robotot ennek megfelelően mozgatjuk, míg a célhoz nem értünk.

9. A robotok dinamikai rendszere

Az eddigiekben megterveztük a robot mozgását és meghatároztuk annak paramétereit: a pálya egyes pontjain fennálló izületi sebességeket és gyorsulásokat. Ezekből az értékekből kell pillanatról pillanatra kiszámolnunk a kívánt mozgás fenntartását biztosító nyomatékokat. A legutolsó lépés a kapott értékeket a hajtásokat közvetlenül működtető beavatkozó jellé alakítani: feszültséggé, árammá vagy vezérlőimpulzussá. Az izületi sebességek, a robotra ható külső és belső erők, valamint a hajtások által leadott nyomatékok közötti viszonyt dinamikai egyenletek írják le, melyeknek megoldása a robot mozgásegyenleteit eredményezi. Az egyenletekben szereplő erőhatások a következők:

- a robotra ható gravitációs erő, továbbá a szerszám, megfogó, illetve a fölemelt munkadarab súlya,
- az izületi hajtások által kifejtett erők,
- a szomszédos izületek között fennálló erőhatások,
- a tömegek mozgatásából származó erők:
    - tehetetlenségi (gyorsulással arányos) erők,
    - centrifugális (a sebesség négyzetével arányos) erők

- az izületek mechanikai csatolásából származó (az izületek sebességeinek szorzatával arányos) erők.

Valamennyi dinamikai hatás figyelembevétele rendkívül bonyolult, gyakorlati számításokra alkalmatlan eredményeket adna, ezért elhanyagolásokat kell tennünk: például a csillapító vagy veszteségi elemek elhagyásával, a karok rugalmasságának, csavarodásának mellőzésével. Minden egyes robotra meg kell szerkeszteni a lényeges elemeket magába foglaló dinamikai modellt, amely segítségével megkaphatjuk a robot térbeli tömegeloszlását leíró tehetetlenségi tenzort. Ez alapján írjuk fel, majd oldjuk meg a mozgásegyenleteket.

A témakör megismeréséhez már rendelkezésünkre áll szakirodalom: kimerítő részletességgel tárgyalja a [3]-al jelölt forrásmű, jegyzetünkben csak megismételni tudnánk eredményeit.

10. Irodalomjegyzék, források

[1] Dr. Arz Gusztáv, Dr. Lipóth András, Dr. Merksz István: Robotmanipulátorok, LSI, 1988, Bp.

[2] Festo Cosimir Educational robotvezérlő program, Súgó

[3] Dr. Kulcsár Béla: Robottechnika, LSI, Bp.

[4] Dr Siegler András: Robot irányítási modellek, LSI, 1987, Bp.

[5] Dr Siegler András: Egy 6 szabadságfokú antropomorf manipulátor kinematikája és számítógépes vezérlése, SZTAKI tanulmányok, 116/1980

Az ábrák nagy része a Festo Cosimir Educational program látványának felhasználásával készült.

11. Tartalomjegyzék