

Tudományos Diákköri Dolgozat

Posztprocesszor vizsgálata KUKA robotra RobotExpert szoftverrel

Analyzing of postprocessing on KUKA robot with RobotExpert software



Készítette:

Cservenák Ákos

G-1MMx

Konzulensek:

Dr. Szabó Tamás

Dr. Hegedűs György

2014

Tartalomjegyzék

1. Bevezetés	2
2. A robot kinematikai leírása.....	3
2.1 Koordinátarendszerek felvétele.....	3
2.2 Denavit-Hartenberg-féle paraméterek felvétele	6
2.3 Direkt kinematikai feladat leírása.....	6
2.4 Direkt kinematikai feladat megoldása SciLab szoftverrel.....	9
3. Hálózat beállítás	11
4. RobotExpertben készített program feltöltése a robotvezérlőbe	15
5. A program futtatása során felmerült problémák és ezek elhárítása.....	18
6. Összefoglalás.....	22
7. Irodalomjegyzék.....	23
Függelék	24
F1. Helyzetmátrixok.....	24
F2. Homogén koordináták és további helyzetmátrixok.....	25
F3. A SciLab szoftverben megírt program forráskódja	26
F4: Az orientációhoz szükség forráskód-részlet	29

1. Bevezetés

Jelen TDK dolgozat témáját a 2013-ban készített [5] szakdolgozat vetette fel. A szakdolgozatban a RobotExpert szoftver megismerése, és programozói szintű használata valósult meg. Két feladat került leírásra, egy egyszerű logisztikai, és egy turbinalapát sorjázását bemutató feladat. A szakdolgozat beadásakor a Műhelycsarnokban megtalálható KUKA KRC 15/2 roboton technikai okok miatt elmaradt a gyakorlati tesztelés, azaz a RobotExpert programozási lehetőségeinek a valóságban történő kipróbálása. A jelenlegi tanév elejére elhárultak a technikai akadályok, így lehetőség nyílt a KUKA robot megismerésére, és RobotExperttel való programozást megvalósítására.

Jelen dolgozat továbbá célul tűzi ki az adott robot elméleti kinematikai leírását is. Az [1] irodalom a Denavit-Hartenberg-féle paraméterek, helyzetmátrixok és homogén koordináták, valamint a direkt és inverz kinematikai feladat leírására szolgál. A [2] irodalom a korábbi, 2012-ben készített TDK dolgozat, amely a jelen TDK dolgozat kinematikai részéhez szolgál alapul. Ott egy Fanuc LRMate 200iC típusú robotra készült el a kinematikai leírás, amely szerkezetét tekintve elég jó hasonlóságot mutat a KUKA robottal. A [3] irodalom tartalmazza a KUKA robot geometriai adatait. A [4] a hálózati felépítést magába foglaló irodalom. A [6] irodalom felsorolja a robotvezérlőben megjelenő figyelmeztető- és hibaüzeneteket.

A 2. fejezet a KUKA robot kinematikai leírását ismerteti. A 2.1 fejezetben a [2] irodalomban is látható módon a csukló koordináta-rendszerek kerülnek meghatározásra a [3] irodalomban található méretek alapján. A 2.2 fejezetben a 2.1 fejezet segítségével az [1] irodalomban használt jelölések alapján kerül felírásra a Denavit-Hartenberg-féle paraméterek. Ezek a paraméterek szükségesek az ún. direkt, illetve inverz kinematikai feladatokhoz. A 2.3 fejezetben a direkt kinematikai feladat került ismertetésre, részletezve a helyzetmátrixokat és a homogén koordinátákat. A 2.4 fejezet ismerteti az elkészített SciLab szimulációs programot.

A 3. fejezet egy laptopon feltelepített RobotExpert szimulációs programja és a robot-vezérlő közötti hálózati beállítást tartalmazza.

A 4. fejezet a RobotExpertben készült fájl robot-vezérlőbe történő feltöltésén vezet végig.

Az 5. fejezet a program futtatása során felmerült problémákat tárja fel, majd ezek megoldására is kitér a [6] irodalom alapján.

A 6. fejezet összefoglalja az elért eredményeket, és kitér a továbbfejlesztési lehetőségekre.

2. A robot kinematikai leírása

A direkt kinematikai feladat célja, hogy a robot ismert csuklóí alapján meghatározható legyen az effektor helyzete. Ehhez a 2.2 alfejezetben ismertetésre kerülő Denavit-Hartenberg-féle paramétereket használjuk fel, hasonlóan a korábban készített [2] TDK dolgozatban ismertetett módon. Egy direkt kinematikai feladatot SciLab-ban is programoztunk, melynek részleteit és eredményét szintén ez a fejezet mutatja be.

Az inverz kinematikai feladat célja, hogy az effektor helyzetének ismeretében meghatározhatóak legyenek az egyes csuklószögek. Tehát adott lesz a $H_{0,k}$ mátrix, és ismeretlenek a θ_k szögek. Az eddigiek alapján jelenleg $k=1,2,3,4,5$ és 6 értékek lesznek. A θ_1 , θ_2 és θ_3 szögeket geometriai megfontolás alapján vehető fel. Ezután a θ_4 , θ_5 és θ_6 meghatározása az effektor térbeli orientációját tartalmazó transzformációs mátrix alapján lehetséges. Egy inverz kinematikai feladat megoldása jelen TDK dolgozat keretében nem kerül sor, ezt későbbi publikációkban szeretnénk ismertetni.

2.1 Koordinátarendszerek felvétele

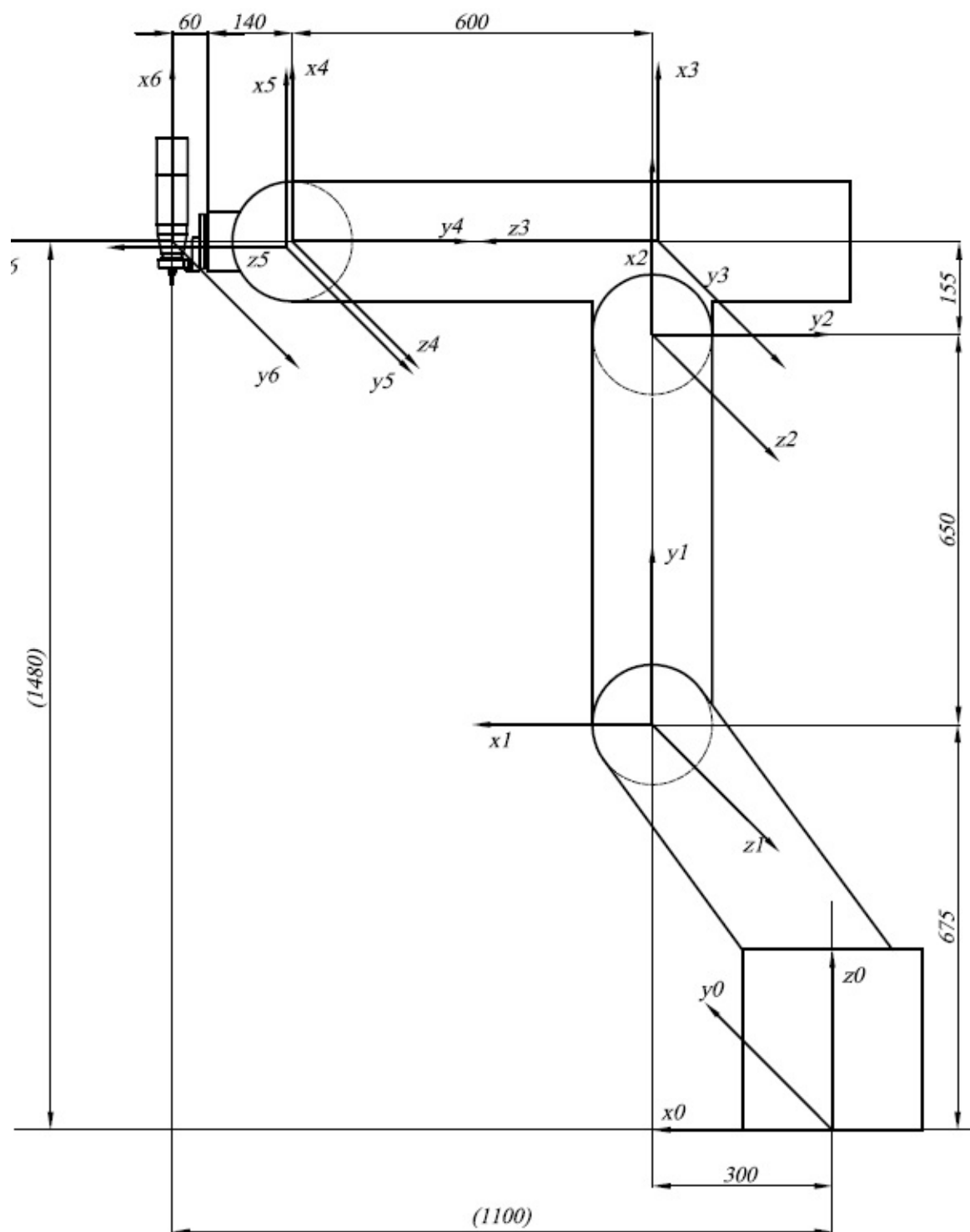
A koordinátarendszerek meghatározásához először az 1. ábrán látható geometriai méreteket kellett megkeresni a [3] irodalomból.

A megmunkáló egység egy felfogó segítségével helyezkedik el a roboton. Az aggregát középvonala 60 mm-re helyezkedik el a robot 6. csuklójának ismert pontjától.

Az [3] irodalomban található geometriai méretek és a 6. csukló - aggregát távolság alapján meg lehetett határozni a [2] irodalomban felépített módon az egyes csuklók koordinátarendszereit az 1. ábrán látható módon.

A Műhelycsarnokban megtalálható KUKA robot az alaphelyzetébe állt, majd erről a helyzetről megközelítőleg oldalnézetből készült egy fénykép, amely a 2. ábra illusztrál. Az ábrán fel lett tüntetve a 1. ábrán is látható csukló-koordinátarendszerek.

Megjegyzés: a koordinátarendszer tengelyei és feliratai egyes helyeken fehérrel, egyes helyeken pedig pirossal lett jelölve a jobb láthatóság érdekében.



1. ábra: Csuklók koordináta-rendszerei



2. ábra: KUKA robot a csuklóban elhelyezett koordinátarendszerekkel

2.2 Denavit-Hartenberg-féle paraméterek felvétele

A távolságok és koordináta-rendszerek alapján meg lehetett határozni az s_k , θ_k , α_k és a_k Denavit-Hartenberg-féle paramétereket, ahol $k=1,2,3,4,5$ és 6 a robot 6 darab csuklója alapján. A kapott értékeket az 1. táblázat sorolja fel.

Az 1. táblázatban használt jelölések a [1] alapján:

- s_k : a z_{k-1} és x_k tengelyek metszéspontjának z_{k-1} koordinátája (KR_{k-1} -ben),
- θ_k : a z_{k-1} tengely körüli forgás előjeles szöge, amely az x_{k-1} tengelyt az x_k tengelybe viszi,
- α_k : az x_k tengely körüli forgás előjeles szöge, amely a z_{k-1} tengelyt a z_k tengelybe viszi,
- a_k : a z_{k-1} és z_k tengelyek távolsága (a normáltranszverzális hossza, amely nem lehet negatív).

1. táblázat: KUKA robot alaphelyzetben felvett paraméterei

Paraméterek Csuklók	s_k	θ_k	α_k	a_k
1	675	0°	+90°	300
2	0	+90°	0°	650
3	0	0°	+90°	155
4	600	0°	-90°	0
5	0	0°	+90°	0
6	140+60	0°	0°	0

2.3 Direkt kinematikai feladat leírása

Ebben az alfejezetben a cél a megfogó helyzetének a meghatározása lesz a 2.2 alfejezetben meghatározott Denavit-Hartenberg-féle paraméterek alapján.

A megfogó helyzetét az alaphoz képest egy helyzetmátrix segítségével írható fel. Ennek lépései a [1] alapján:

1. Transzlációs és rotációs mátrixok felírása az egyes csuklók között,
2. A csuklók közötti helyzetmátrixok felírása a vonatkozó transzlációs és rotációs mátrixok megfelelő sorrendű összeszorzásával,

3. A helyzetmátrixok összeszorzásával megkapható az alap és a megfogó közötti helyzetmátrix.

A translációs mátrixok felírását a (1) és (2) összefüggések írják le egy általános „k” csuklóra a [7] alapján (a „c” a koszinusz, az „s” a szinusz trigonometrikus függvényeket jelenti):

$$\underline{\underline{H}}_{s_k} = \begin{bmatrix} 0 & 0 & 0 & | & 0 \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & 1 & | & s_k \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (1)$$

$$\underline{\underline{H}}_{a_k} = \begin{bmatrix} 1 & 0 & 0 & | & a_k \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & 1 & | & 0 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (2)$$

A rotációs mátrixok felírását mutatják a (3) és (4) összefüggések a [7] alapján egy általános „k” csuklóra (itt is a „c” a koszinusz, az „s” a szinusz trigonometrikus függvényeket jelenti):

$$\underline{\underline{H}}_{\alpha_k} = \begin{bmatrix} 1 & 0 & 0 & | & 0 \\ 0 & c\alpha_k & -s\alpha_k & | & 0 \\ 0 & s\alpha_k & c\alpha_k & | & 0 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (3)$$

$$\underline{\underline{H}}_{\theta_k} = \begin{bmatrix} c\theta_k & -s\theta_k & 0 & | & 0 \\ s\theta_k & c\theta_k & 0 & | & 0 \\ 0 & 0 & 1 & | & 0 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (4)$$

Az (1), (2), (3) és (4) összefüggésekben szereplő mátrixok összeszorzása alapján fel lehet írni az (5) összefüggésben szereplő két csukló közötti helyzetmátrixot:

$$\underline{\underline{H}}_{k-1,k} = \underline{\underline{H}}_{s_k} \underline{\underline{H}}_{\theta_k} \underline{\underline{H}}_{\alpha_k} \underline{\underline{H}}_{a_k} \quad (5)$$

Kifejtve „k-1” és „k” csuklók közötti helyzetmátrix a (6) összefüggésnek megfelelően:

$$\underline{\underline{H}}_{k-1,k} = \begin{bmatrix} \underline{\underline{h}} & \underline{\underline{r}} \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix}_{k-1,k} = \begin{bmatrix} c\theta_k & -s\theta_k c\alpha_k & s\theta_k s\alpha_k & | & a_k c\theta_k \\ s\theta_k & c\theta_k c\alpha_k & -c\theta_k s\alpha_k & | & a_k s\theta_k \\ 0 & s\alpha_k & c\alpha_k & | & s_k \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (6)$$

Itt a „h” almátrix az orientációért, az „r” vektor az elmozdulásért felelős.

Végül az alap és a „k” csukló közötti helyzetmátrixok felírhatóak az általános „k” csuklóra a (7) összefüggésben:

$$\underline{\underline{H}}_{0k} = \underline{\underline{H}}_{01} \underline{\underline{H}}_{12} \dots \underline{\underline{H}}_{k-1,k} = \prod_{j=1}^k \underline{\underline{H}}_{j-1,j} = \begin{bmatrix} \underline{\underline{h}} & \underline{\underline{r}} \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix}_{0k} \quad (7)$$

A KUKA robot esetében 6 csukló van, így k értéke 1,2,3,4,5 és 6 értékek lehetnek.

A robotra való mátrixok felírása és kiszámítása a PTC MathCAD Express szoftverrel történt, az eredmények ebből a szoftverből lettek beillesztve.

Az 1-2 csuklók közötti „ H_{12} ”, a 2-3 csuklók közötti „ H_{23} ”, a 3-4 csuklók közötti „ H_{34} ”, a 4-5 csuklók közötti „ H_{45} ” és az 5-6 csuklók közötti „ H_{56} ” mátrixokhoz szükséges translációs és rotációs mátrixok számítása a Függelék F1. fejezetében tekinthető át.

A (6) összefüggés alapján a 0-1 csukló közötti helyzetmátrix:

$$H_{01} := H_{s1} \cdot H_{\theta1} \cdot H_{a1} \cdot H_{a1} = \begin{bmatrix} 1 & 0 & 0 & 300 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 675 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Az 1-2 csukló közötti helyzetmátrix:

$$H_{12} := H_{s2} \cdot H_{\theta2} \cdot H_{a2} \cdot H_{a2} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 650 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A 2-3 csukló közötti helyzetmátrix:

$$H_{23} := H_{s3} \cdot H_{\theta3} \cdot H_{a3} \cdot H_{a3} = \begin{bmatrix} 1 & 0 & 0 & 155 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A 3-4 csukló közötti helyzetmátrix:

$$H_{34} := H_{s4} \cdot H_{\theta4} \cdot H_{a4} \cdot H_{a4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 600 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A 4-5 csukló közötti helyzetmátrix:

$$H_{45} := H_{s5} \cdot H_{\theta5} \cdot H_{a5} \cdot H_{a5} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Végül az 5-6 csukló közötti helyzetmátrix:

$$H_{56} := H_{s6} \cdot H_{\theta6} \cdot H_{a6} \cdot H_{a6} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 200 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A helyzetmátrixok utolsó oszlopának első 3 eleme, a (6) összefüggésben szereplő „r” vektornak megfelelően mutatja az előző csuklóhoz képesti elmozdulást, látható, hogy összhangban van az 1. ábrán feltüntetett méretekkel.

Az egyes helyzetmátrixok összeszorzásával kapjuk meg az aggregát helyzetmátrixát az alaphoz képest.

$$H_{06} := H_{01} \cdot H_{12} \cdot H_{23} \cdot H_{34} \cdot H_{45} \cdot H_{56} = \begin{bmatrix} 0 & 0 & 1 & 1100 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1480 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A homogén koordinátákat általános alakban a következőképpen írható fel:

$$R_k = H_{0,k} \cdot R_0$$

Itt az „R₀” értéke az alap koordinátája lesz, vagyis X=Y=Z=0 mm:

$$R_0 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Így lehet felírni k=1,2,3,4,5 és 6 értékek esetén a homogén koordinátákat. Az „R₀₁”, „R₀₂”, „R₀₃”, „R₀₄” és „R₀₅” vektorok és a hozzájuk szükséges „H₀₂”, „H₀₃”, „H₀₄” és „H₀₅” helyzetmátrixok kiszámítását és eredményét a Függelék F2. fejezete ismerteti. A „R₀₆” homogén koordináta az alap és az aggregát közötti elmozdulást jelenti. Ennek kiszámítása és eredménye a következőképpen néz ki:

$$R_{06} := H_{06} \cdot R_0 = \begin{bmatrix} 1100 \\ 0 \\ 1480 \\ 1 \end{bmatrix}$$

Látható, hogy X irányban 1100 mm-re, Y irányban 0 mm-re, míg Z irányban 1480 mm-re helyezkedik el az aggregát középvonala, mely megfelel a 1. ábrán látható méreteknek.

2.4 Direkt kinematikai feladat megoldása SciLab szoftverrel

A feladat a θ_k szögek változtatása a többi érték változatlan hagyása mellett, a cél pedig egy háromdimenziós ablakban láthatóvá válik a robotkar mozgása.

Ehhez először deklarálni kellett s_k , θ_k , α_k és a_k változókat, majd a „H” általános mátrixot. A „H” helyzetmátrixot fel kellett tölteni a (6) összefüggésben szereplő értékek alapján.

Deklarálni kellett egy $\theta_{max} = 0,5$ értéket. Létre kellett hozni egy „for” ciklust, amelyben „t” fut 0-tól 2.1π -ig, 0,1 lépéssel.

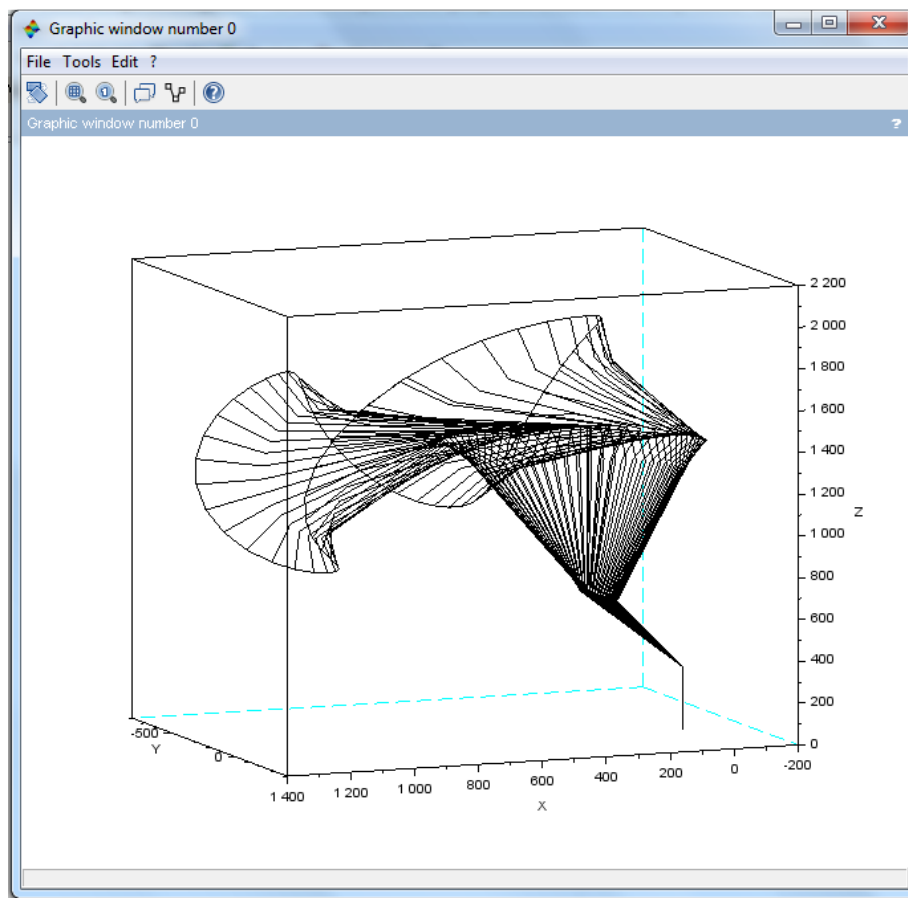
A szögek változása szinusz függvény alapján változik a következő módon:

$$\theta_1 = \theta_{max} \cdot \sin(t) \qquad \theta_2 = \theta_{max} \cdot \sin(t)$$

$$\theta_3 = \theta_{max} \cdot \sin(2 \cdot t) \qquad \theta_4 = \theta_{max} \cdot \sin(2 \cdot t)$$

$$\theta_5 = 2 \cdot \theta_{max} \cdot \sin(2 \cdot t) \qquad \theta_6 = \theta_{max} \cdot \sin(2 \cdot t)$$

Majd ezek alapján meg kellett határozni az összes helyzetmátrixot és homogénkoordinátákat, ami a 2.3 alfejezetben ismertetésre került. Ezt követően a kirajzoláshoz szükséges X, Y és Z koordinátákat kellett deklarálni a homogén koordináták alapján. Végül a „param3d” parancs segítségével ki lehetett rajzoltatni a robotkar mozgását a 3. ábrának megfelelően.



3. ábra: SciLab-ban programozott robotkar mozgás kirajzoltatása

Ez a görbe egy bonyolult geometriájú megmunkálását illusztrálja, ilyen példának lehetne említeni a festést.

Az elkészült program forráskódját a Függelék F3. fejezete ismerteti.

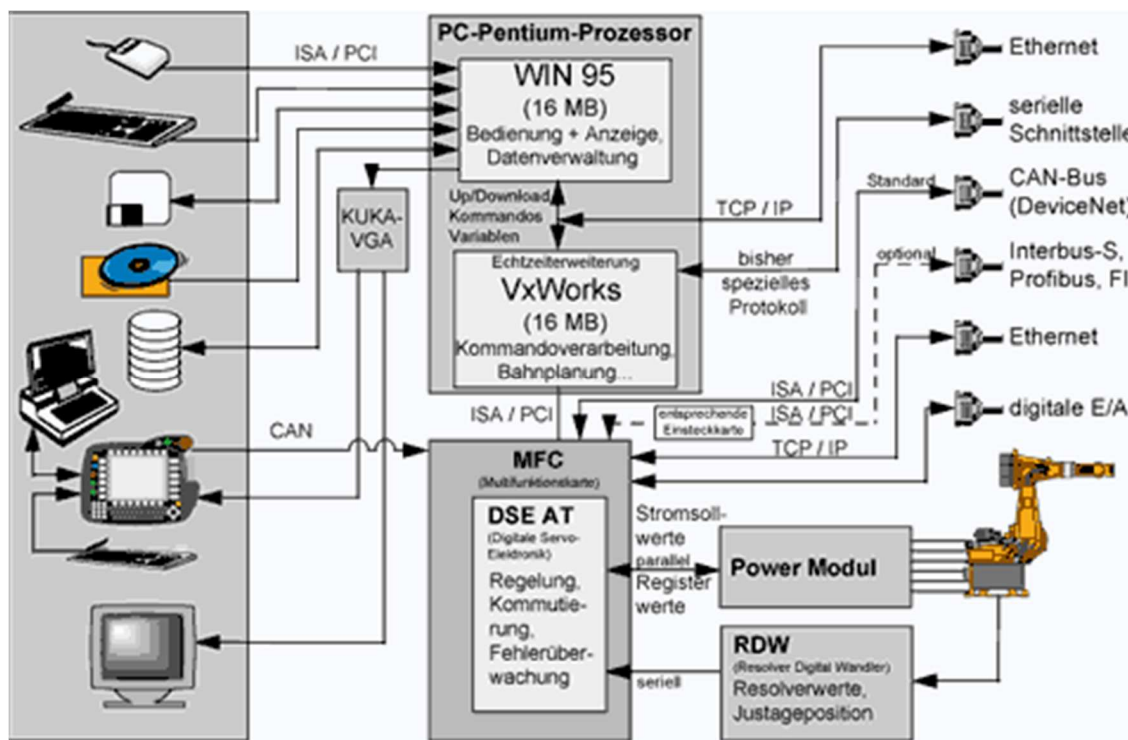
3. Hálózat beállítás

A RobotExpertben elkészített végrehajtható fájlt a tényleges végrehajtáshoz a robot vezérlőjébe kell tölteni. A robot vezérlője Windows 95, a RobotExpertet futtató laptop Windows 7 operációs rendszert futtat, így az operációs rendszer eltérősége miatt nehezebb volt megvalósítani az összeköttetést.

A robot vezérlője és a laptop közötti kapcsolatot a következő módokon lehet létrehozni:

- CD-olvasó és író segítségével,
- floppy-lemez,
- USB pendrive-val,
- hálózaton keresztül.

Ezek közül legegyszerűbben hálózati úton lehet megvalósítani, ugyanis itt nincs szükség valamilyen eszköz folyamatos illesztésére és leválasztására, hanem mindig folyamatos kapcsolat jön létre a két gép között. Ehhez mindkét gépen beállításokat kellett módosítani. A robot vezérlőjében két hálózati kártya is található a [4] „PC-Based Robot Control KR C2” alfejezeténél található módon, melyet az 4. ábra szemléltet.



4. ábra: A robot környezet felépítése

A VxWorks hálózati kártya valósítja meg az operációs rendszer és a servo-vezérlési elektronika közötti kommunikációt, és ez a kártya felelős a parancsvégrehajtásért, az útvonaltervezésért.

Az SMC EtherEz hálózati kártya a számítógép és a külső hálózat közötti kapcsolatot hozza létre. A külső hálózatra a kártya Ethernet csatlakozóján keresztül válik lehetővé. A gépen a hálózati kártya driverei már telepítve voltak, így az IP-cím és az alhálózati maszk számát kellett beírni. Az IP-cím mindkét hálózati kártya esetén fix.

A VxWorks esetén a [3] alapján a következőket kellett beírni:

- IP-cím: 192.0.1.2
- Alhálózati maszk: 255.255.255.0

A SMC EtherEz esetén a Műhelycsarnokban található hálózat alapján kellett beállítani az adatokat:

- IP-cím: 193.6.12.100
- Alhálózati maszk: 255.255.255.0
- Itt még be kellett állítani a router alapértelmezett IP-címét: 193.6.12.254.

Az SMC EtherEz esetén automatikus IP-címmel szemben egy fix IP-címet volt érdemes beállítani, mert a laptop távoli meghajtóként mindig ugyanarra a címre tud csatlakozni.

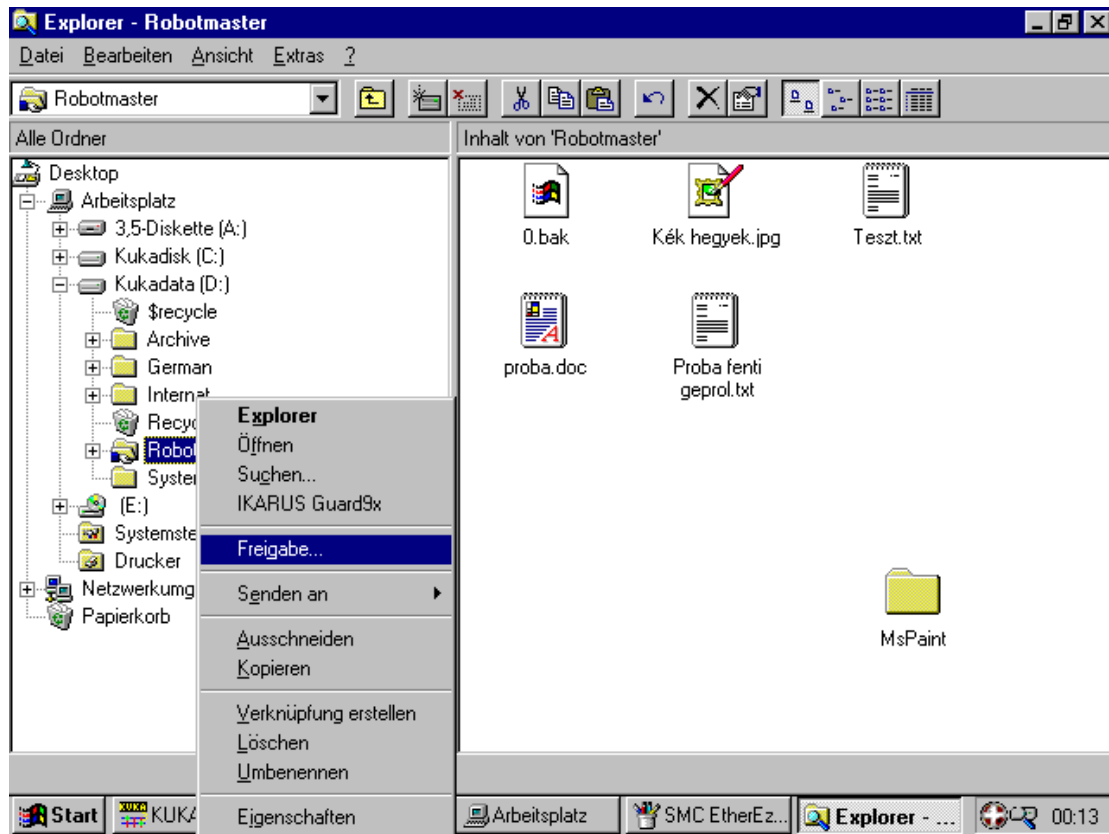
Ezután a robot vezérlőjében a KUKA robotvezérlő szoftver (KukaBof) sikeresen elindult, a rendszerfájlokat a VxWorks hálózatán keresztül letöltötte, és a külső hálózatra sikeresen felcsatlakozott.

Következő lépésként a robot vezérlője és a laptop közötti kommunikáció megvalósítása volt a cél, lehetőleg a következő módon:

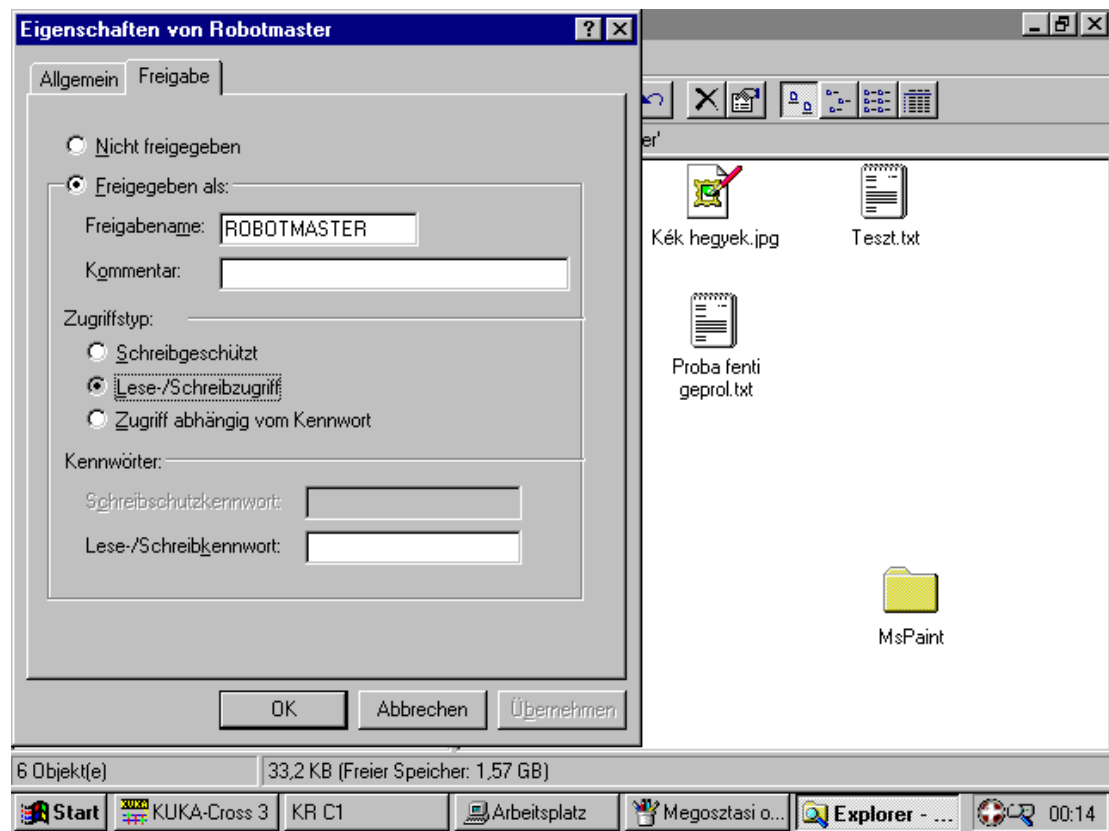
- a robot vezérlőjéből kijövő UTP kábel a falon található Ethernet aljzatba csatlakozzon,
- a laptop vezeték nélkül a Műhelycsarnokban található "SGT" wi-fi hálózatra csatlakozzon.

A cél tehát az volt, hogy további eszközök, pl. switch, további UTP kábelek nélkül a legegyszerűbben, legkevesebb eszközzel létrejöjjön a kommunikáció.

Ahhoz, hogy a robot vezérlőjében hozzá lehessen férni a szükséges adatokhoz, megosztást kellett létrehozni. Az 5. ábrán látható könyvtárat az egér jobb gombjának megnyomása után a „Freigabe” opció megnyitásával, majd a 6. ábrán látható módon be kellett állítani a megosztást. A „Freigegeben als:” és „Lese-/Schreibzugriff” opciók bejelölésével, illetve a megfelelő név megadásával lehetett megvalósítani, hogy olvasás és írásra is, megfelelő néven meg legyen osztva az adott mappa.

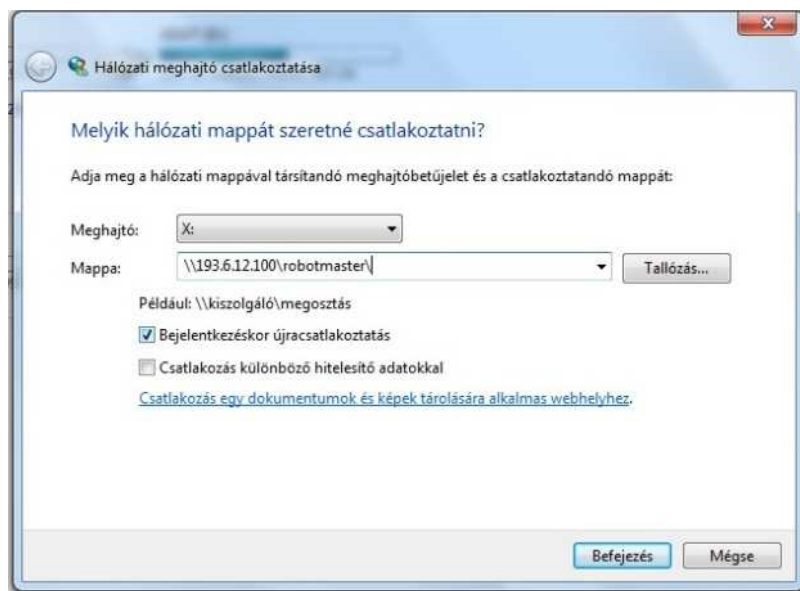


5. ábra: A megosztás lehetőségének elérése



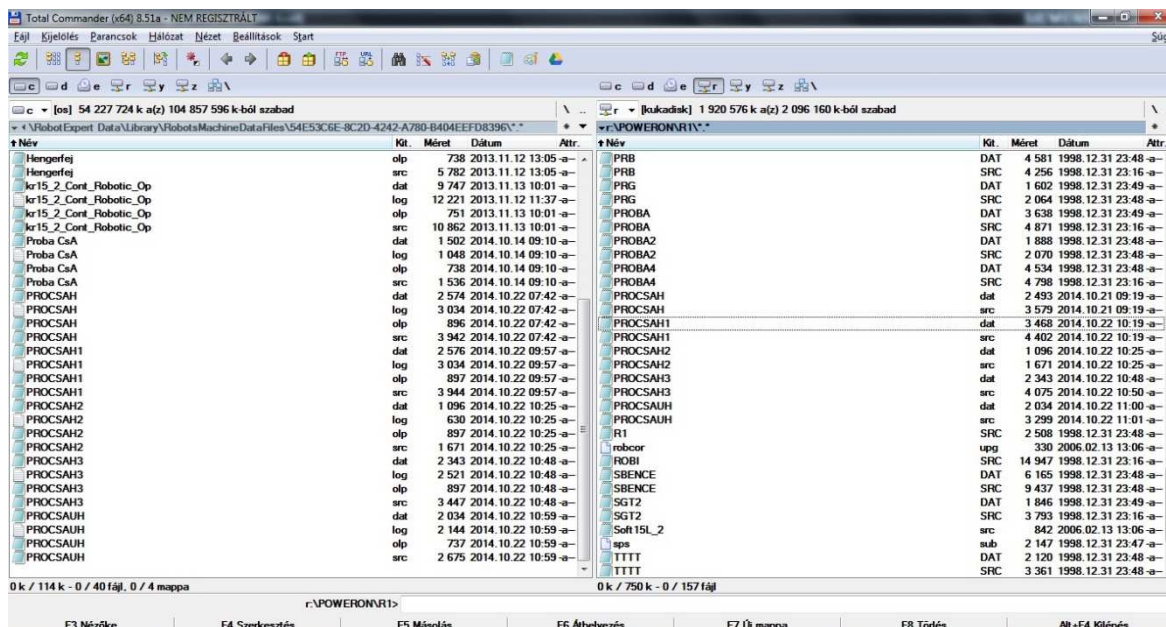
6. ábra: A mappa megosztása olvasásra és írásra

A laptopról a „Hálózati helyek” mappánál nem lehetett közvetlenül elérni a robot vezérlő „Krcpc” nevű gépét, ezért az IP-cím alapján lehetett elérni a vezérlőben megosztott mappát. Az intézőben a hálózati meghajtó csatlakoztatásánál be kellett állítani az „\\IP-cím\megosztott mappa” struktúrát, jelen esetben az 7. ábrának megfelelően „\\193.6.12.100\robotmaster\” címet.



7. ábra: A robot vezérlő elérése távoli meghajtóként

A fentebb leírtak alapján már el lehetett érni azt a mappát a Total Commander fájlkezelő szoftverben, ahová a végrehajtható programokat fel feltölteni az 8. ábrának megfelelően.



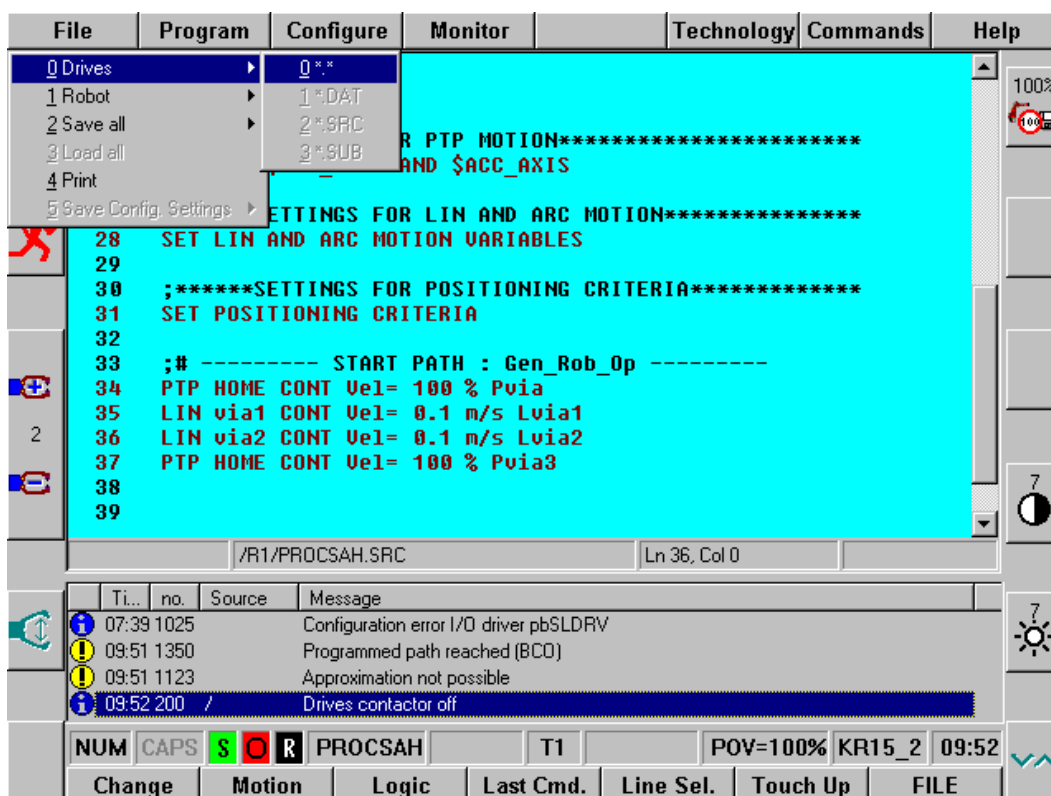
8. ábra: A programokat tartalmazó mappa elérése

4. RobotExpertben készített program feltöltése a robotvezérlőbe

Az [5] szakdolgozatban ismertetett módon lehet létrehozni mozgás-utasításokat, majd ezeket szimulálása is megvalósítható. Az [5] szakdolgozat 4.6 fejezetében vázolt úton lehetséges a modellezett és szimulált robot mozgását megvalósító program KUKA robotra való forrásfájl elkészítése. A létrehozott 4 db fájl közül a *.src és a *.dat kiterjesztésű fájlok a lényegesek. A *.src fájl tartalmazza a forráskódot és abban a mozgásutasításokat. A *.dat fájl tartalmazza a programozott helyzeteket.

Miután a laptopon sikeresen létrejöttek a fájlok, azután a 8. ábra jobb oldalán látható „R1” könyvtárba kell átmásolni a két darab lényeges fájlt. Ekkor még csak a robot vezérlőjében található HDD-re tölti a fájlokat, nem a robot irányításához szükséges meghajtóra.

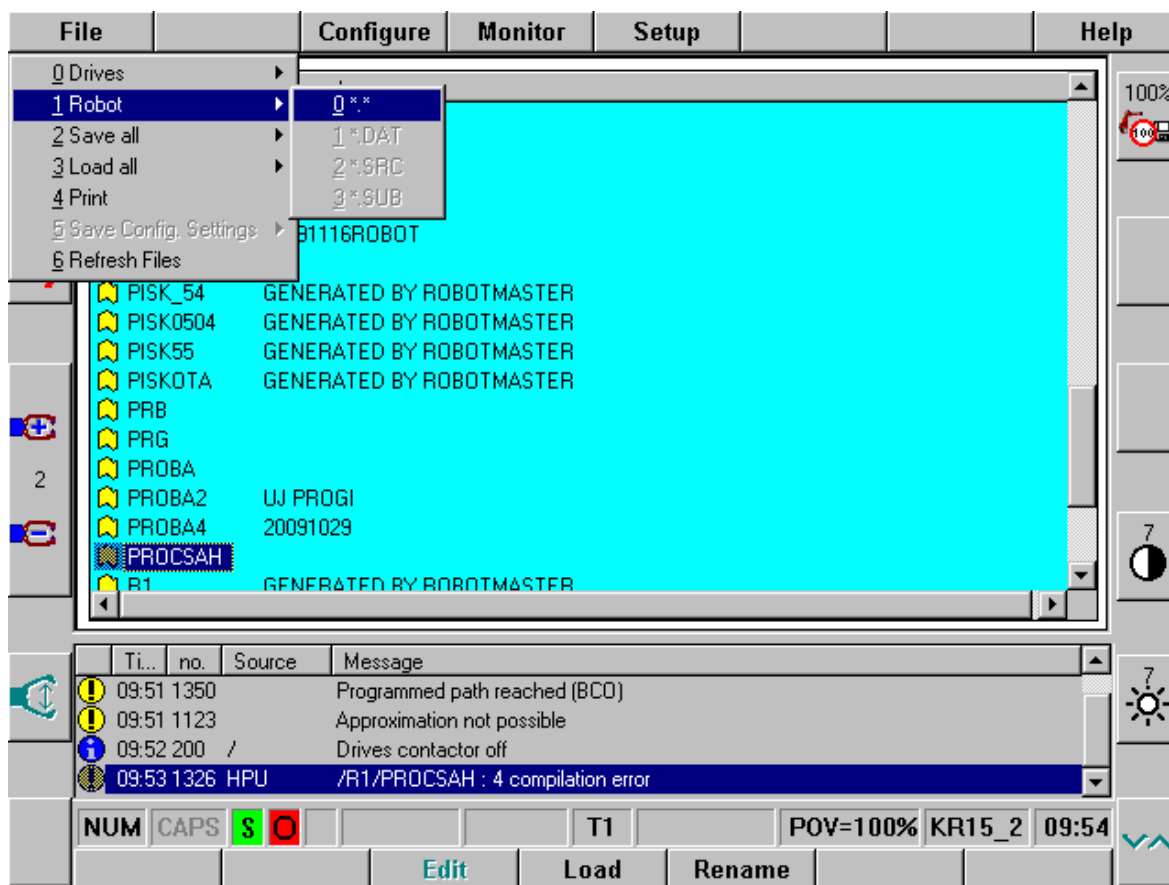
A roboton a 9. ábrán látható „File/0 Drives/0 *.*” opció elérésével tekinthetők meg a „KRC/POWERON/R1” mappában levő fájlok.



9. ábra: A robotvezérlő külső hálózathoz elérhető meghajtó elérése

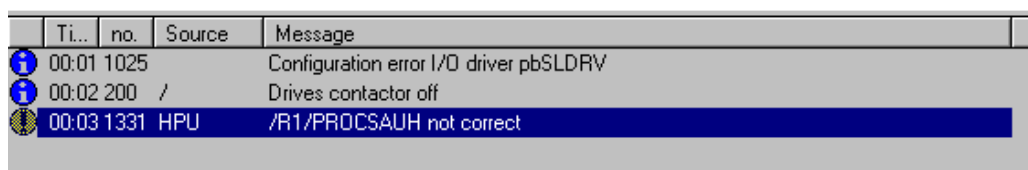
A megfelelő fájl kiválasztásával a 10. ábrán alul látható „Load” parancs segítségével lehetséges a program robotvezérlőjébe való betöltése. Az ábrán sikeresen betöltötte a programot, de 4 fordítói hibát írt ki. Ez a forráskódban levő eltérések miatt jön létre. Ilyen például a tizedesjegyek különbsége: a RobotExpert nem hoz létre tizedesjegyet a mozgásparamétereknél, míg a KUKA vezérlő igényli azt egész szám esetén is.

Sikeres betöltés után a 10. ábrán látható „File/1 Robot/0 *.*” opció segítségével lehet visszatérni a robot irányításához szükséges meghajtóra.



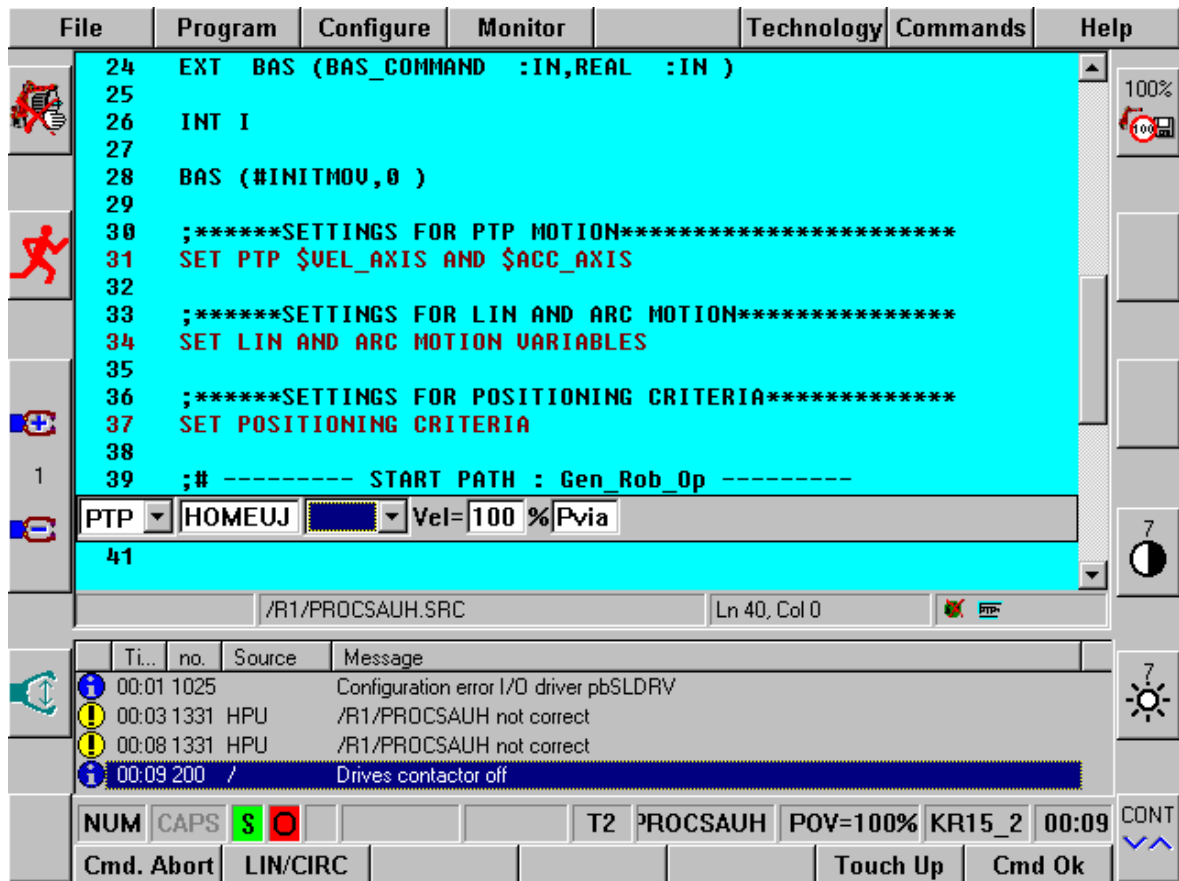
10. ábra: Sikeres betöltés után és a Robot-meghajtóba való átlépési lehetőség

A Robot-meghajtóba való sikeres átlépés után a programot még nem lehet a programozó memóriába tölteni a fordítói hibák miatt, ekkor hibaüzenetet ír ki a 11. ábrának megfelelően.



11. ábra: Program memóriába való betöltésekor megjelenő hibaüzenet

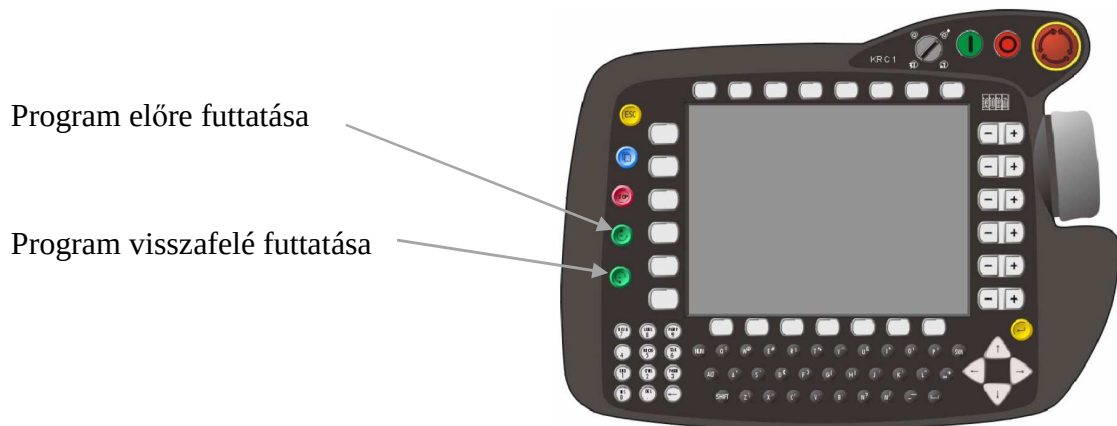
A sikeres betöltéshez módosítani kell a programot. Ezt a 10. ábrán alul látható „Edit” opció felhasználásával tehető meg. Ekkor a 12. ábrán szemléltetett ablak fogad. Itt minden egyes mozgásparancsot az alul látható „Change” opció megnyomásával, majd elfogadásával lehet helyrehozni a fordítói hibát. Itt a KukaBof szoftver a KUKA vezérlőnek megfelelően módosítja a forráskódot.



12. ábra: A mozgásparancsok módosítása

Sikeres módosítás után a programot a szoftver már hibaüzenet nélkül betölti a program memóriába.

A KCP (Kuka Control Panel) a robot kezelőszerve, mely a 13. ábrán látható. A KCP bal oldalán található két darab zöld gomb, ezek közül az egyik a program előrefuttatásáért, a másik a visszafelé futtatásért felelős. A program memóriába való betöltés után ezen gombok nyomva tartásával lehet a programot futtatni.



13. ábra: Kezelőpanel és a programot futtató gombok elérése

5. A program futtatása során felmerült problémák és ezek elhárítása

A program futtatása során a következő figyelmeztető- vagy hibaüzenetek jelentek meg:

- *programmed path reached (BCO)*
- *orientation velocity not programmed*
- *command velocity exceeded*
- *angle status inadmissable*

Ezen kívül felmerült az is, hogy a robot nem olyan orientációval hajtja végre a parancsot, ahogy azt a RobotExpert megjeleníti, vagyis az agregát a valóságban vízszintesen áll, a RobotExpertben megjelenített függőleges helyett.

Ezek figyelembevétele vagy elhárítása szükséges volt ahhoz, hogy zökkenőmentesen végrehajtsa a robot a mozgásokat. Az üzenetek és megoldási javaslatok a [6] hibaüzenetek listáját tartalmazó irodalom alapján lett feldolgozva.

Programmed path reached (BCO): a robot a programozott pályát elérte, a 14. ábrán látható a figyelmeztető üzenet. Ez akkor jelenik meg, ha a robot a programozott pontokhoz képest egy tetszőleges külső pontból közelíti a program első programozott pontját. Habár a „Program előre futtatása” folyamatosan nyomva van tartva, itt mégis megáll a robot mozgása, és csak a gomb felengedése és újbóli megnyomása esetén lehet a robot mozgás-irányítását tovább folytatni.

	Ti...	no.	Source	Message
	09:54	1025		Configuration error I/O driver pbSLDRV
	11:10	1350		Programmed path reached (BCO)

14. ábra: *Programmed path reached (BCO)* figyelmeztető üzenet

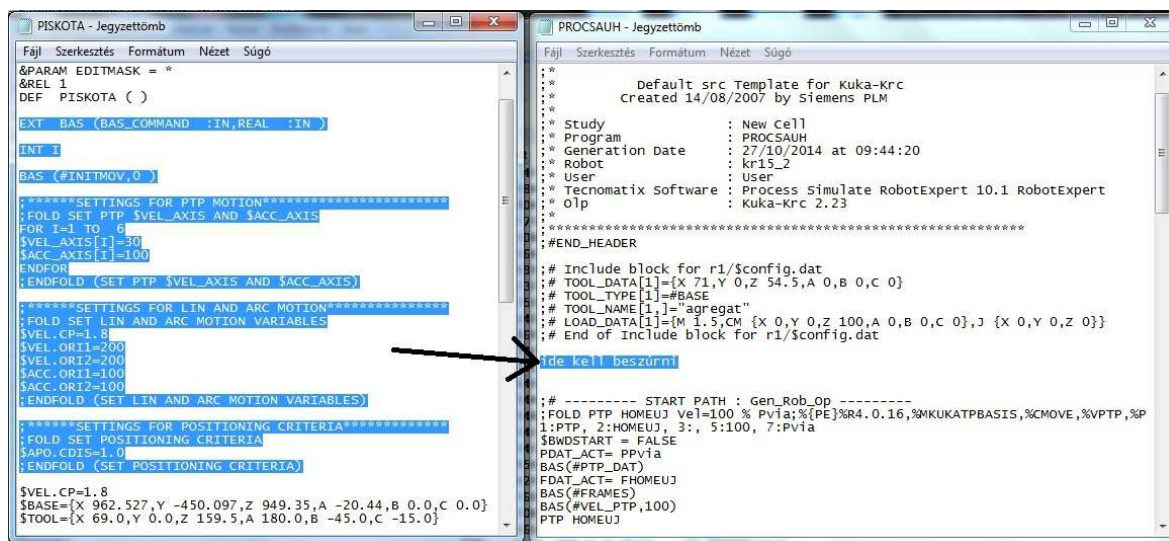
Orientation velocity not programmed: a sebesség orientációja nincs programozva, a hibaüzenetet a 15. ábra szemlélteti. A programban lineáris interpolációval végrehajtott mozgást is programoztunk, a RobotExpert nem állított be orientációt, viszont a robot vezérlője igényli azt.

	Ti...	no.	Source	Message
	09:54	1025		Configuration error I/O driver pbSLDRV
	10:35	1417		Orientation velocity not programmed
	10:35	220	/	Drives contactor off, intermediate circuit loaded.

15. ábra: *Orientation velocity not programmed* hibaüzenet

Itt a robotvezérlőben található korábbi években készített programok közül keresni kellett olyat, ami tartalmazza ezt az orientáció beállítást.

A megfelelő forráskód-részletet át kellett másolni a RobotExpert által generált forráskódba a 16. ábrán illusztráltaknak megfelelően, és ekkor már a vezérlő ismerte a lineáris mozgás orientációját. A forráskód-részletet a Függelék F4 fejezete írja le.



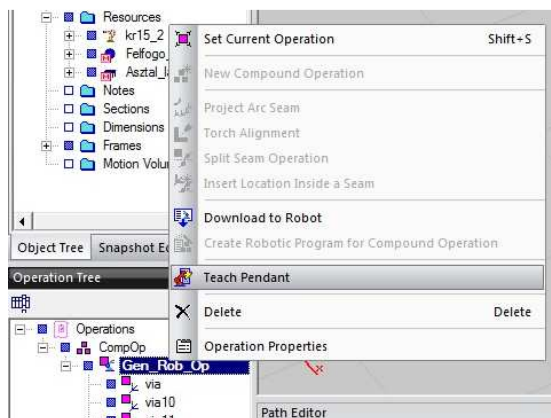
16. ábra: Orientáció programozása

Command velocity exceeded: a programozott sebesség meghaladja a robot képességét, a hibaüzenet a 17. ábrán látható módon tűnik fel. Ekkor a program azonnal megáll, és a dinamikus fék bekapcsol, amit üzenetekben jelez is („Dynamic braking active” és „Active commands inhibited”). A hibajelenség akkor fordult elő, amikor a robotvezérlő által gyárilag beállított HOME alaphelyzetből a robot függőlegesen lefelé, vagyis –Z irányba szeretne menni. Viszont a csukló el szeretne fordulni a másodperc töredéke alatt, amit a robot nem tud végrehajtani. Megoldásként el kell kerülni az ilyen útvonalakat.

09:54 1025	Configuration error I/O driver pbSLDRV
10:20 1350	Programmed path reached (BCD)
10:20 1102	Command velocity exceeded A4
10:20 1138	Dynamic braking cancelled
10:20 200 /	Drives contactor off
10:21 1376 HPU	Active commands inhibited

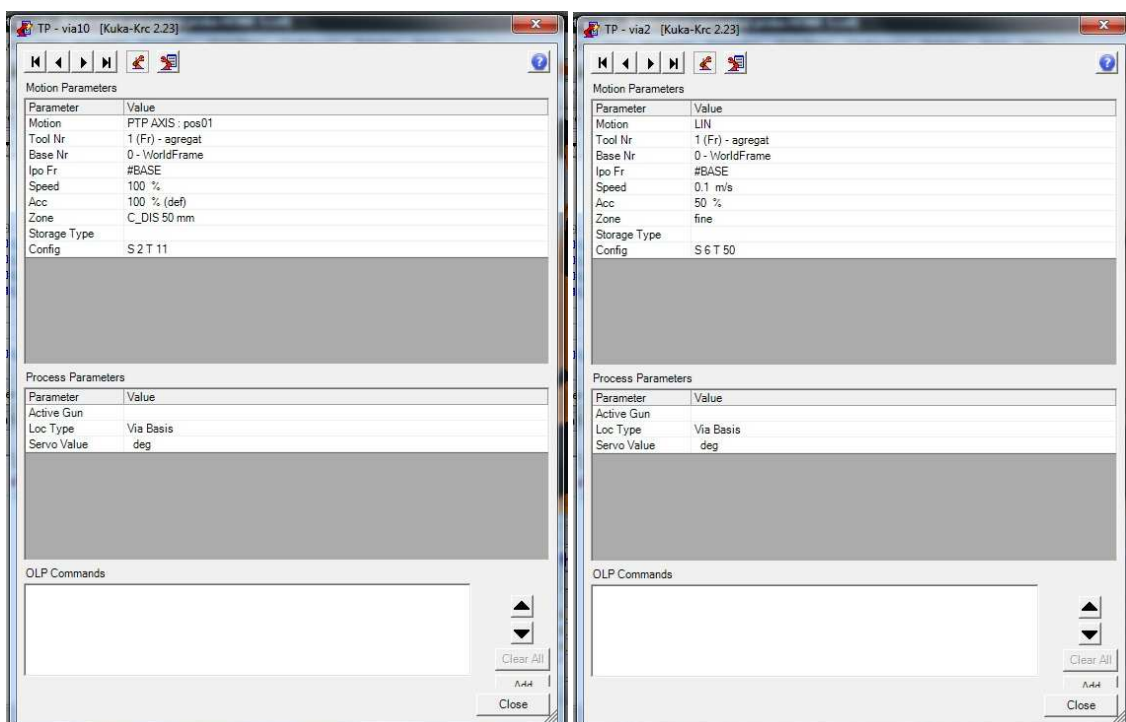
17. ábra: Command velocity exceeded

Angle status inadmissable: a szög státusza elfogadhatatlan. Ez abból adódik, hogy a RobotExpert által létrehozott helyzetek itt még nincsenek betanítva, így a robotvezérlő nem tudja megfelelően kezelni. A megoldáshoz a RobotExpertben kellett további beállításokat létrehozni. A 18. ábrán látható „Teach Pendant” opció segítségével lehet elérni az egyes programozott helyzetek adatait.



18. ábra: A „Teach Pendant” opció elérése

A „Teach Pendant” ablakban minden egyes helyzethez be kell állítani, hogy PTP, azaz pontról pontra/gyorsjárat, vagy LIN, azaz lineáris utasítás legyen a 19. ábrán megfigyelhető módon. A PTP esetén be kell állítani a programozott ponthoz tartozó helyzetet is.

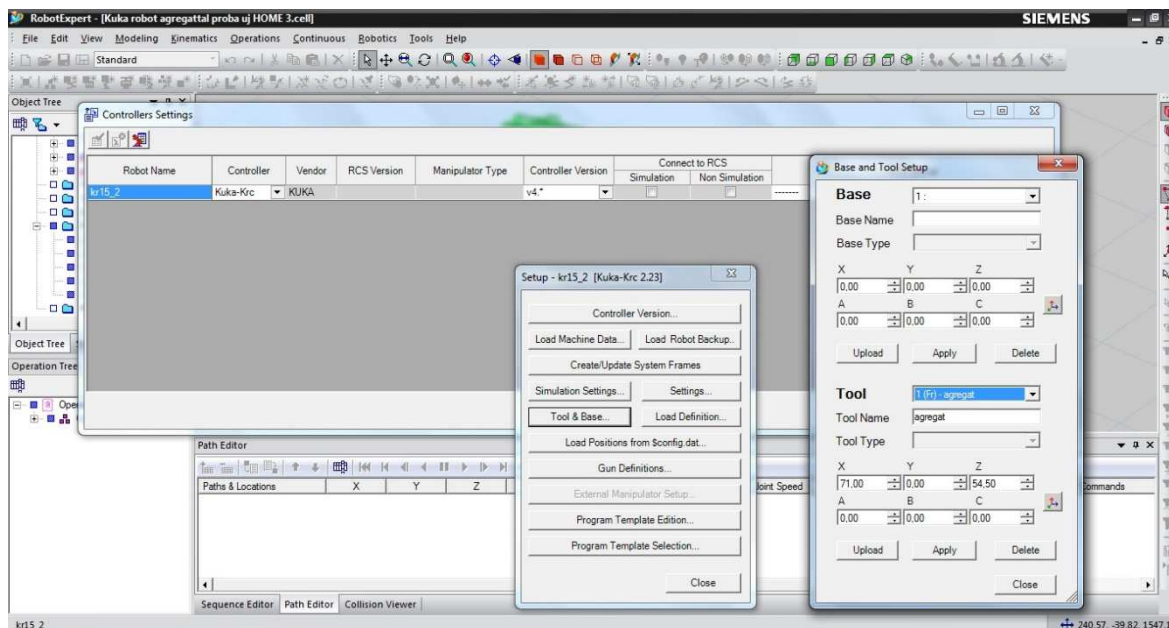


19. ábra: A PTP és LIN utasítások beállítása

A robot nem megfelelő orientációval hajtja végre a parancsot: ez a RobotExpertben és a KUKA robot eltérő koordináta-rendszerei miatt fordult elő. Ennek megoldásához a RobotExpertben kell megfelelően beállítani a koordináta-rendszereket.

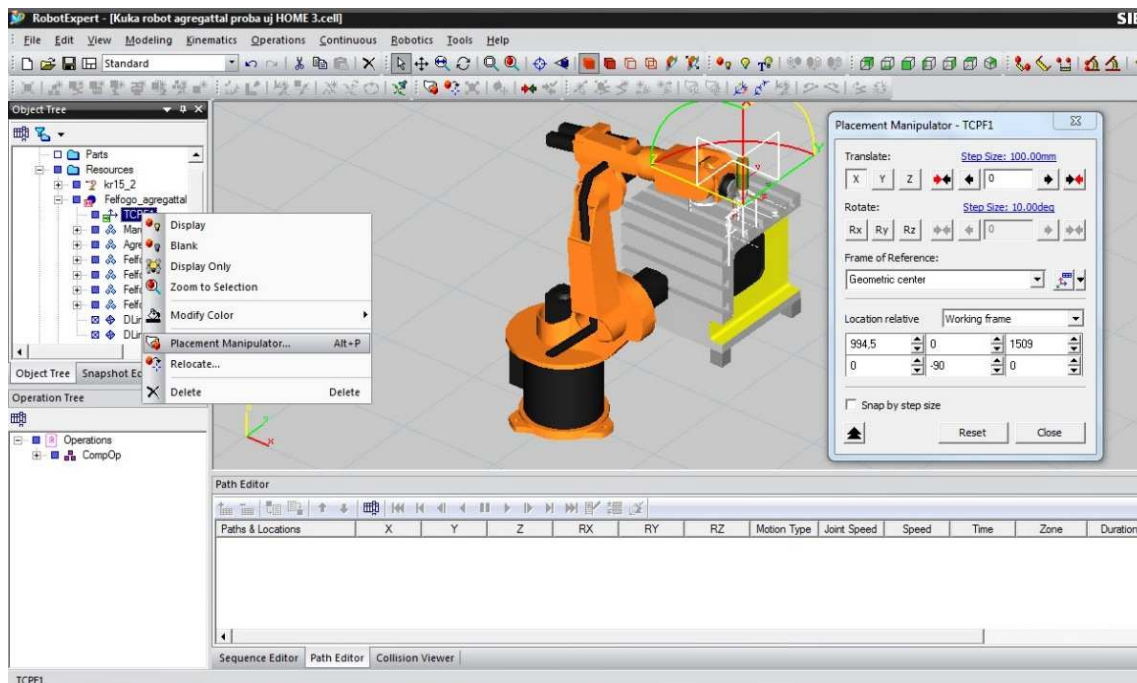
Ehhez először definiálni kellett a robot vezérlő beállítása között az agregátot és ahhoz rendelni kellett egy koordináta-rendszert a 20. ábrának megfelelően.

Az opció elérése a „Robotics/Controller Settings/Robot Setup/Tool & Base” úton lehetséges. Itt a megfogó orientációja A=0, B=0 és C=0 értéket kaptak.



20. ábra: Az agregát definiálása és koordinátarendszer hozzárendelése

Ezután az agregát saját TCPF1 koordinátarendszerét is módosítani kellett a 22. ábrának megfelelően. Ezt a TCPF1-re jobb gombbal kattintva és a Placement Manipulator opciót kiválasztva lehetséges megtenni. Itt az orientáció RX=0, RY=-90 és RZ=0 értékekre lett átállítva. A -90 fok oldotta meg azt a gondot, hogy az agregát a valóságban ne vízszintesen, hanem függőlegesen álljon a 21. ábrán látható módon.



21. ábra: A TCPF1 megfelelő helyzetbe állítása

A hibák elhárítása után a program hibamentesen lefutott. Egy konkrét program is készül, ami az előadás prezentációján lesz bemutatva.

6. Összefoglalás

A TDK dolgozat a KUKA robot kinematikai leírásával és a RobotExpert szoftver KUKA robotra való alkalmazásával foglalkozott.

A szakirodalmakat áttekintve, az alapján kerültek megrajzolásra a csukló koordináta-rendszerek, valamint a Denavit-Hartenberg-féle paraméterek felírása. Ez alapján elkészült egy egyszerű SciLab szimulációs program a robot direkt kinematikai feladatának vizsgálatára.

A dolgozat további része a hálózati elérését-, valamint a generált fájl program-memóriába való megfelelő betöltésének lépéseit ismerteti. Végül a KUKA roboton futtatott program működése során felmerült problémák kerültek feltárássra. A fejezet további része a megoldásokat részletezi.

A TDK dolgozat továbbfejlesztési lehetőségei között szerepel a robot inverz kinematikai leírása, és a szimulációs program ilyen irányú továbbfejlesztése.

Az elméleti vizsgálódásokon túl fontos cél a RobotExpertben egy megmunkáláshoz tartozó bonyolult geometriájú pálya megvalósítása.

A kutatómunka a Miskolci Egyetem stratégiai kutatás területén működő Mechatronikai és Logisztikai Kiválósági Központ keretében valósult meg.

7. Irodalomjegyzék

[1] Király B.: **Ipari robotok kinematikai és dinamikai elemzése**, oktatási segédlet, Miskolc, 1995.

[2] RÓNAI, L. – CSERVENÁK, Á.: **Robotkiszolgáló-rendszer kinematikai szimulációja**. Miskolci Egyetem, TDK konferencia, 2012

[3] KUKA Roboter GmbH: **Robot KR15/2 Technical Data**, Augsburg, 1998

[4] **KUKA Robot System Overview**, http://vvl.fh-reutlingen.de/index.php?lang=en&goto_page=P_GLP_KUKA_OVERVIEW&open=1

[5] Cservenák Á.: **Virtuális robotlabor létrehozása RobotExpert segítségével**, Szakdolgozat, Miskolci Egyetem, 2013

[6] KUKA Roboter GmbH: **Error messages / troubleshooting**, Augsburg, 1998

Link utoljára ellenőrizve: 2015. január 6.

Függelék

F1. Helyzetmátrixok

A 0-1. csukló közötti számítás a következőképpen néz ki. A translációs és rotációs mátrixok felírása:

$$H_{s1} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 675 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{\theta 1} := \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{a1} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) & 0 \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{a1} := \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 300 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Az 1-2. csukló közötti mátrixok:

$$H_{s2} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{\theta 2} := \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{a2} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_2) & -\sin(\alpha_2) & 0 \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{a2} := \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 650 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A 2-3. csukló közötti mátrixok:

$$H_{s3} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{\theta 3} := \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{a3} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_3) & -\sin(\alpha_3) & 0 \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{a3} := \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 155 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A 3-4. csukló közötti mátrixok:

$$H_{s4} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 600 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{\theta 4} := \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{a4} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_4) & -\sin(\alpha_4) & 0 \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{a4} := \begin{bmatrix} 1 & 0 & 0 & a_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A 4-5. csukló közötti mátrixok:

$$H_{s5} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{\theta 5} := \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{a5} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_5) & -\sin(\alpha_5) & 0 \\ 0 & \sin(\alpha_5) & \cos(\alpha_5) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{a5} := \begin{bmatrix} 1 & 0 & 0 & a_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Az 5-6. csukló közötti mátrixok:

$$H_{s6} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 200 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{\theta 6} := \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{a6} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_6) & -\sin(\alpha_6) & 0 \\ 0 & \sin(\alpha_6) & \cos(\alpha_6) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H_{a6} := \begin{bmatrix} 1 & 0 & 0 & a_6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

F2. Homogén koordináták és további helyzetmátrixok

„ H_{02} ” helyzetmátrix:

$$H_{02} := H_{01} \cdot H_{12} = \begin{bmatrix} 0 & -1 & 0 & 300 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1325 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

„ H_{03} ” helyzetmátrix:

$$H_{03} := H_{01} \cdot H_{12} \cdot H_{23} = \begin{bmatrix} 0 & 0 & 1 & 300 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1480 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

„ H_{04} ” helyzetmátrix:

$$H_{04} := H_{01} \cdot H_{12} \cdot H_{23} \cdot H_{34} = \begin{bmatrix} 0 & -1 & 0 & 900 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1480 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

„ H_{05} ” helyzetmátrix:

$$H_{05} := H_{01} \cdot H_{12} \cdot H_{23} \cdot H_{34} \cdot H_{45} = \begin{bmatrix} 0 & 0 & 1 & 900 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1480 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Az „ R_{01} ” homogén koordináta:

$$R_{01} := H_{01} \cdot R_0 = \begin{bmatrix} 300 \\ 0 \\ 675 \\ 1 \end{bmatrix}$$

Az „ R_{02} ” homogén koordináta:

$$R_{02} := H_{02} \cdot R_0 = \begin{bmatrix} 300 \\ 0 \\ 1325 \\ 1 \end{bmatrix}$$

Az „ R_{03} ” homogén koordináta:

$$R_{03} := H_{03} \cdot R_0 = \begin{bmatrix} 300 \\ 0 \\ 1480 \\ 1 \end{bmatrix}$$

Az „ R_{04} ” homogén koordináta:

$$R_{04} := H_{04} \cdot R_0 = \begin{bmatrix} 900 \\ 0 \\ 1480 \\ 1 \end{bmatrix}$$

Az „ R_{05} ” homogén koordináta:

$$R_{05} := H_{05} \cdot R_0 = \begin{bmatrix} 900 \\ 0 \\ 1480 \\ 1 \end{bmatrix}$$

F3. A SciLab szoftverben megírt program forráskódja

```
//
```

```
clear
```

```
function[H]=helyzet(s,teta,alfa,a)
```

```
    H=zeros(4,4)
```

```
    H(1,1)=cos(teta)
```

```
    H(1,2)=-sin(teta)*cos(alfa)
```

```
    H(1,3)=sin(teta)*sin(alfa)
```

```
    H(1,4)=a*cos(teta)
```

```
    H(2,1)=sin(teta)
```

```
    H(2,2)=cos(teta)*cos(alfa)
```

```
    H(2,3)=-cos(teta)*sin(alfa)
```

```
    H(2,4)=a*sin(teta)
```

```
    H(3,1)=0
```

```
    H(3,2)=sin(alfa)
```

```
    H(3,3)=cos(alfa)
```

```
H(3,4)=s
H(4,1)=0
H(4,2)=0
H(4,3)=0
H(4,4)=1
endfunction

tetmax=0.5
j=0
for t=0 : 0.1 : 2.1*%pi
    j=j+1
    teta1=tetmax*sin(t)
    teta2=tetmax*sin(t)
    teta3=tetmax*sin(2*t)
    teta4=tetmax*sin(2*t)
    teta5=2*tetmax*sin(2*t)
    teta6=tetmax*sin(2*t)
    H01=helyzet(675,teta1,%pi/2,300)
    H12=helyzet(0,%pi/2+teta2,0,650)
    H23=helyzet(0,teta3, %pi/2, 155)
    H34=helyzet(600,teta4,-%pi/2,0)
    H45=helyzet(0,teta5, %pi/2, 0)
    H56=helyzet(200,teta6,0,0)
    H01=H01
    H02=H01*H12
    H03=H01*H12*H23
    H04=H01*H12*H23*H34
    H05=H01*H12*H23*H34*H45
    H06=H01*H12*H23*H34*H45*H56
    R0=zeros(4,1)
    R0(4,1)=1
    R01=H01*R0
    R02=H02*R0
    R03=H03*R0
```

R04=H04*R0

R05=H05*R0

R06=H06*R0

X=zeros(7+1)

Y=zeros(7+1)

Z=zeros(7+1)

X(2)=0

Y(2)=0

Z(2)=300

X(2+1)=R01(1,1)

Y(2+1)=R01(2,1)

Z(2+1)=R01(3,1)

X(3+1)=R02(1,1)

Y(3+1)=R02(2,1)

Z(3+1)=R02(3,1)

X(4+1)=R03(1,1)

Y(4+1)=R03(2,1)

Z(4+1)=R03(3,1)

X(5+1)=R04(1,1)

Y(5+1)=R04(2,1)

Z(5+1)=R04(3,1)

X(6+1)=R05(1,1)

Y(6+1)=R05(2,1)

Z(6+1)=R05(3,1)

X(7+1)=R06(1,1)

Y(7+1)=R06(2,1)

Z(7+1)=R06(3,1)

param3d(X,Y,Z)

xp(j)=X(7+1)

yp(j)=Y(7+1)

zp(j)=Z(7+1)

end

param3d(xp,yp,zp)

F4: Az orientációhoz szükség forráskód-részlet

```
EXT BAS (BAS_COMMAND :IN,REAL :IN )
INT I
BAS (#INITMOV,0 )

;*****SETTINGS FOR PTP MOTION*****
;FOLD SET PTP $VEL_AXIS AND $ACC_AXIS
FOR I=1 TO 6
$VEL_AXIS[I]=30
$ACC_AXIS[I]=100
ENDFOR
;ENDFOLD (SET PTP $VEL_AXIS AND $ACC_AXIS)

;*****SETTINGS FOR LIN AND ARC MOTION*****
;FOLD SET LIN AND ARC MOTION VARIABLES
$VEL.CP=1.8
$VEL.ORI1=200
$VEL.ORI2=200
$ACC.ORI1=100
$ACC.ORI2=100
;ENDFOLD (SET LIN AND ARC MOTION VARIABLES)

;*****SETTINGS FOR POSITIONING CRITERIA*****
;FOLD SET POSITIONING CRITERIA
$APO.CDIS=1.0
;ENDFOLD (SET POSITIONING CRITERIA)
$VEL.CP=1.8
$BASE={X 962.527,Y -450.097,Z 949.35,A -20.44,B 0.0,C 0.0}
$TOOL={X 69.0,Y 0.0,Z 159.5,A 180.0,B -45.0,C -15.0}
```