

TDK DOLGOZAT



Jakab Dávid

2012.

KÖSZÖNETNYILVÁNÍTÁS

Ezúton szeretném tisztaszívvel köszönetemet nyilvánítani családomnak, hogy az elmúlt időszak feszült pillanataiban is, türelmükkel és segítségükkel támogatták munkámat. Főként, hogy a felmerült problémák keltette heves reakcióknak sokszor részesei voltak.

Kiemelt köszönettel tartozom Burkus Ervin, és Dr. Odry Péter konzulenseimnek a dolgozat során adott hasznos ötleteikért, jó tanácsaikért, s hogy nem csak szakmailag, hanem emberileg is támogattak tanulmányom elkészítése alatt. Hálával tartozom Tóth Árpád tanáromnak, ki eltérő szakterületétől függetlenül, hozzájárult és tanácsaival támogatta dolgozatom elkészültét. Köszönettel tartozom továbbá a Dunaújvárosi Főiskolán e félévben (2012/2013/1) tanító tanárainak, hogy elnézően, és türelemmel fogadták, hogy olykor óráikon és vizsgáikon nem jelentem meg a tanulmány kivitelezésének időszaka alatt.

Elismeréssel tartozom munkahelyem kollégáinak, hogy türelemmel és megértéssel fogadták, hogy olykor a munka rovására ment dolgozatom iránti szenvedélyem.

TDK DOLGOZAT EREDETISÉGI NYILATKOZAT

Alulírott Jakab Dávid (Neptun-kód: GTDFFB) kijelentem, hogy az „**Öt szabadságfokú robotkar vezérlése Matlab GUI platformon, SimMechanics szimulációval**” című TDK dolgozatomban foglaltak saját munkám eredményei, és csak forrásokat (szakirodalom, eszközök stb.) használtam fel. A szakmai irodalomtól eltekintve, - mely főként az általam nagy elismeréssel bíró Dr. Mester Gyula könyveiből és publikációiból ered -, a dolgozatom matematikai hátterét önállóan oldottam meg, főként a robot manipulátorok egyediségére tekintettel. A hivatkozásomban fellelhető Mohammed Reyad AbuQassem, a Gázai Egyetemen írt Mester dolgozata, mely hasonlóságokat mutathat; habár az általa írt dolgozatot tekintve, inkább javítást igényel, mind matematikai, mind irodalmi tekintetben. Éppen ezért, és a dolgozat egyediségére törekedve, próbáltam eltérő szimulációt, egyéb programfelépítést valamint vezérlést megvalósítani.

Jelen TDK dolgozatomat a hatályos szerzői jogi szabályok betartásával készítettem.

Dátum: Dunaújváros, 2012.11.04.

Jakab Dávid

KIVONAT

A dolgozat részletesen bemutatja egy 5 szabadságfokú robotkar valós idejű vezérlését Matlab program segítségével, SimMechanics szimulációval. A robotkar vezérlése Matlab GUI platformon keresztül vezérelhető, ahol a szükséges matematikai számításokat - mint az Inverz és Direkt Kinematika, Trajektória, Jacobi mátrix és soros port kommunikáció -, Matlab m scriptek és függvények segítségével kivitelezzi. A robotkar 3D szimulációját egy SolidWorks-ben készített, majd SimMechanics környezetbe integrált környezetben mutatja be. A dolgozat részletesen tárgyalja a robotkar vezérlésének matematikai hátterét, a valós idejű vezérlés és szimuláció kivitelezését, valamint a megvalósítás alatt felmerülő problémákat és azok megoldásait. A dolgozat bemutatja továbbá, a fent említett 5 szabadságfokú robotkar valós idejű útvonaltervezését az Inverz kinematika és Direkt kinematika tekintetében. A matematikai háttér, az útvonaltervezés, valamint a valós idejű vezérlés és szimuláció bemutatásához, a robotkar kivitelez egy, a készítő által leprogramozott algoritmust, melynek során a robotkar önállóan megoldja a Hanoi-torony feladványt.

ABSTRACT

The essay presents in detail a 5 degrees of freedom robotic arm's control with the help of Matlab program and with SimMechanics simulations. The arm can be controlled through a Matlab GUI platform, where the necessary mathematical calculations such as are executed with 'Matlab m scripts and functions' Inverse and Forward Kinematics, Trajectories, Jacobian matrix and serial port communications. The 3D simulation of the robot arm is presented in a SolidWorks platform which was integrated into a SimMechanics environment. The paper, also introduces the arm's mathematical background, and the arm's real-time execution and control. Then again the arising problems and their solutions during the implementation. Also discussed the above-mentioned 5 degrees of freedom robotic arm's real-time Trajectory movement in the sight of Inverse and Forward kinematics. To show the mathematical background of the route planning and the real-time control and simulation; the arm is carrying out an algorithm by itself (which was pre-programmed by the designer) and it is able to solve the Hanoi tower problem anonymously.



„Nem születhetnek zsenik és szentek, ha nem léteznek a normát messze túllépő emberek, és nem látom be, milyen jogon kíván a normákat csak az egyik irányba túllépő egyéneket. Valamiféle szimmetriának lennie kell.”

Isaac Asimov

NÓMENKLATÚRA

3D	<i>Three-Dimensional Graphics</i>	<i>Három-Dimenziós Grafika</i>
DH	<i>Denavit-Hartenberg</i>	<i>Denavit-Hartenberg transzformáció</i>
DOF	<i>Degrees of Freedom</i>	<i>Szabadságfok</i>
FK	<i>Forward Kinematic</i>	<i>Direkt Kinematika</i>
GUI	<i>Graphical User Interface</i>	<i>Grafikus Felhasználói Felület</i>
IK	<i>Inverse Kinematic</i>	<i>Inverz Kinematika</i>
PC	<i>Personal Computer</i>	<i>Személyi Számítógép</i>
RRT	<i>Revolute – Revolute – Prismatic</i>	<i>Rotációs – Rotációs – Transzlációs</i>
RRR	<i>Revolute – Revolute – Revolute</i>	<i>Rotációs – Rotációs – Rotációs</i>
TP	<i>Trajectory Planning</i>	<i>Útvonaltervezés</i>

RÖVIDÍTÉSEK

T_i	<i>Homogeneous Transformation Matrix</i>	<i>homogén transzformációs mátrix</i>
a_i	<i>Link Length</i>	<i>alkatrész hosszúság</i>
α_i	<i>Link Twist of Two Joints</i>	<i>alkatrészek közti szögelfordulás</i>
d_i	<i>Link Offset</i>	<i>alkatrész eltolás</i>
c_i	<i>Cosine Θ_i</i>	<i>koszinusz Θ_i</i>
s_i	<i>Sine Θ_i</i>	<i>szinus Θ_i</i>
Θ_i, q_i	<i>Joint Angle</i>	<i>összekötő szög</i>
R	<i>Rotation Matrix</i>	<i>rotációs mátrix</i>
Θ_{234}	$\Theta_2 + \Theta_3 + \Theta_4$	$\Theta_2 + \Theta_3 + \Theta_4$
$q(t)$	<i>Angle Move</i>	<i>szögelfordulás</i>
$\dot{q}(t)$	<i>Angle Velocity</i>	<i>szögsebesség</i>
$\ddot{q}(t)$	<i>Angle Acceleration</i>	<i>szöggyorsulás</i>
q_0	<i>Initial Angle Value</i>	<i>kezdeti szögérték</i>
q_f	<i>Final Angle Value</i>	<i>végző szögérték</i>
v_0	<i>Initial Angle Velocity</i>	<i>kezdeti szögsebesség</i>
v_f	<i>Final Angle Velocity</i>	<i>végző szögsebesség</i>
t_0	<i>Initial Time Value</i>	<i>kezdeti időérték</i>
t_f	<i>Final Time Value</i>	<i>végző időérték</i>
Rad	<i>Radian</i>	<i>radián</i>
Deg	<i>Degrees</i>	<i>fok</i>

TARTALOMJEGYZÉK

1.	BEVEZETÉS	1
1.1	Történelmi áttekintés.....	1
1.2	Célkitűzés.....	3
2.	ELMÉLETI HÁTTÉR	4
2.1	Robotcsuklók.....	4
2.2	Robotszegmensek.....	5
2.3	Robot manipulátorok konfigurációi	5
2.4	Robot manipulátorok munkaterei.....	6
2.5	Effektor	8
2.6	Csuklókoordináták	9
2.7	Világkoordináták.....	10
3.	AZ AL5B ROBOTKAR KINEMATIKÁJA	11
3.1	A Denavit-Hartenberg (DH) mátrix	11
3.2	Az AL5B robotkar DH mátrixa.....	13
3.3	Direkt kinematika.....	15
3.4	Az effektor orientációja.....	15
3.5	Inverz kinematika.....	18
3.6	Pályatervezés (Trajektória)	20
4.	AZ AL5B ROBOTKAR VEZÉRLÉSE.....	23
4.1	Hardware	23
4.2	Software	25

4.2.1	SolidWorks	25
4.2.2	SimMechanics.....	27
4.2.3	Matlab GUI.....	29
4.2.4	Robotics Toolbox.....	34
4.2.5	Soros-port kommunikáció.....	35
5.	ÖSSZEFOGLALÁS	37
6.	IRODALOMJEGYZÉK	38
7.	FÜGGELÉK	39

TÁBLÁZATJEGYZÉK

1. táblázat <i>Homogén transzformációs mátrixok</i>	11
2. táblázat <i>AL5B DH paraméter táblázata</i>	14
3. táblázat <i>AL5B DH mátrixok csuklókként</i>	14
4. táblázat <i>Szervo motorok impulzustáblázata</i>	23
5. táblázat <i>Matlab - változók deklarálása a Global Workspace-re</i>	29
6. táblázat <i>Az AL5B robotkar paraméterei</i>	39
7. táblázat <i>Az iRob program fájljai</i>	40

ÁBRAJEGYZÉK

1. ábra UNIMATION PUMA 500 robot	2
2. ábra Robot manipulátor rotációs és translációs csuklóval	4
3. ábra Robot manipulátor szegmensei és csuklói	5
4. ábra Antropomorf manipulátor munkatere	6
5. ábra Gömbi manipulátor munkatere	6
6. ábra Hengeres manipulátor munkatere	7
7. ábra Descartes manipulátor munkatere	7
8. ábra Effektor pozicionálás	8
9. ábra 6-szabadságfokú robotkar csuklókoordinátái	9
10. ábra Euler szögek	10
11. ábra Homogén transzformáció bemutatása	12
12. ábra AL5B robotkar koordinátái	13
13. ábra AL5B robotkar DH koordinátái	13
14. ábra "Roll" forgatás a z tengely körül	15
15. ábra "Pitch" bólintás az y tengely körül	16
16. ábra "Yaw" csavarás az x tengely körül	16
17. ábra Az AL5B robotkar geometriája az x,y koordinátákban	19
18. ábra Az AL5B robotkar geometriája az r,z koordináták alapján	19
19. ábra Útvonaltervezés a szög és idő tekintetében	22
20. ábra Az AL5B robotkar	23
21. ábra Szervomotorok pozicionálása	24
22. ábra SSC-32 Servo Controller	24
23. ábra AL5B robotkar SolidWorks modellje	25
24. ábra SolidWorks rotációs kapcsolat (coincident - concentric)	26
25. ábra SolidWorks modell fajsúly definiálása	26
26. ábra SolidWorks SimMechanics regisztrálása	27
27. ábra AL5B modell SimMechanics kapcsolati ábrája 1.	28



DUNAÚJVÁROSI FŐISKOLA

28. ábra <i>AL5B</i> modell <i>SimMechanics</i> kapcsolati ábrája 2.	28
29. ábra <i>SimMechanics</i> - <i>Gain</i> vezérlése <i>Matlab GUI</i> -ból.....	32
30. ábra <i>SimMechanics</i> - <i>Joint Actuator</i>	32
31. ábra Az <i>AL5B SimMechanics</i> modell-je	33
32. ábra <i>A</i> robotkar vezérlőfelülete	33
33. ábra <i>Függelék</i> - <i>Robotkar</i> paraméterei	39

1. BEVEZETÉS

1.1 Történelmi áttekintés

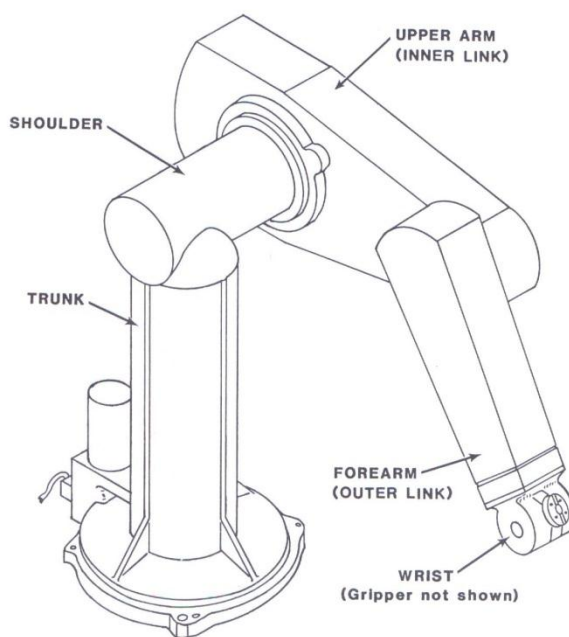
Jelen világunkban, az emberek számára természetessé vált, hogy robotok veszik őket körbe mindenhol. Olyannyira megszokottá vált már, hogy sok esetben észre sem veszik, környezetükben mi mindent intelligens rendszerek és robotok irányítanak. Még kevesebben tudják, hogy történelmünket tekintve, már a legrégebbi írásaink is említést tesznek róluk. A legkorábbi irodalmunkban, a Bibliában is megjelennek már Ádám és Gólem történetében. Az ember mindig is vágyott arra a törekvésére, hogy teremtsen, s alkosson. Ezen célkitűzéseit pedig már a legkorábbi időszakban is próbálta kivitelezni. Kezdvé Tarentumi Arkhütasz fagalambjával, Han császár zenekarával, Hérón automatáin keresztül egészen a középkorig, Möngke kán szökőkútjáig és Leonardo da Vinci madármodelljéig (Anonymous, 2012). A civilizáció fejlődésével pedig a robotok iránti vágyakozás is egyre inkább nőtt. A modernkori robotika elterjedését mégis Karel Capek cseh íróhoz, s az általa, az 1920-as években írt „Rossum Univerzális Robotjai” utópikus művéhez kötik. Víziójában az ember által készített gépeket hívták robotnak, a cseh „robota”, vagyis „munka” szó alapján, amely az akkori kor nyelvtani értelmezésének megfelelően, sokkal inkább „rabszolgamunkát” jelentett. A robotika elterjedésének másik nagy előmozdítója pedig Isaac Asimov, orosz származású amerikai író és biokémikus. Asimov amellet, hogy népszerűvé tette a tudományos-fantasztikus irodalmat, megalkotta a robotika három törvényét, melyet a későbbiek folyamán egy nulladik törvénnyel is kiegészített (Nocks, 2001). Asimov a robotika törvényeit a következőképpen írta le:

0. A robot nem árthat az emberiségnek, és nem nézheti tétlenül, ha az emberiséget veszély fenyegeti.
1. A robotnak nem szabad kárt okoznia emberi lényben vagy tétlenül túrnie, hogy emberi lény bármilyen kárt szenvedjen.
2. A robot engedelmeskedni tartozik az emberi lények utasításainak, kivéve, ha ezek az utasítások az első törvény előírásaiba ütköznének.

3. A robot tartozik saját védelméről gondoskodni, amennyiben ez nem ütközik az első és második törvény előírásaiba.

Asimov és Capek művei, valamint az újonnan kialakult filmgyártás egyértelmű teret biztosított a robotoknak a népszerűséghez és fejlődéshez.

Ennek következményeképpen 1971-ben kifejlesztették a villamos hajtású Stanford, majd 1975-ben pedig a PUMA robotkart (Unimation), melyek meghozták az igazi áttörést (Nocks, 2001).



1. ábra UNIMATION PUMA 500 robot

Forrás: [http://helix.gatech.edu/students/siouxwill/I2\)21\)us.htm](http://helix.gatech.edu/students/siouxwill/I2)21)us.htm)

Mára már a technika ugrásszerű fejlődésének köszönhetően széles körben használják a robotokat szinte minden olyan munkaterületen, ahol a megbízhatóság és az igen nagy pontosság kiemelkedő fontosságú. A robotok meghatározó kellékei az űrkutatásnak és az ipari termelésnek, fokozatosan teret hódítanak az oktatásban és háztartásokban és egyre többet alkalmaznak őket a harcászatban, gyógyászatban is. Napjainkban a robotok alkalmazásában és fejlesztésében Ázsia magasan kiemelkedik a többi kontinens közül. Ázsián belül is Japán az, ahol a legtöbb robotot alkalmaznak. A kontinenseket

figyelembe véve a robotok alkalmazásában Európa áll a második helyen, míg a harmadik helyet Észak Amerika foglalja el.

Ma a világban csaknem 1 millió ipari robot működik, mely méltó példája az emberi találékonyságnak és a robotok iránti szenvedélyének.

A robot fogalmát a Nemzetközi Szabványügyi Hivatal a (ISO „Manipulating Robots”, ISO8373:1996) a következő módon állapítja meg (Mester, Robotika, 2011):

„A robot automatikusan vezérelt, újra programozható, három vagy több tengelyű mozgásra és sokoldalú beavatkozásra képes eszköz. A robot lehet rögzített vagy mozgó eszköz.”

1.2 Célkitűzés

Jelen dolgozat egy Lynxmotion gyártmányú AL5B típusú robotkaron keresztül mutatja be, a robotvezérlés matematikai hátterét. Az 5 szabadságfokú robotkar reprezentálásához, először SolidWorks-ben fogjuk megtervezni a robotkar 3D virtuális mását, hol az alkatrészenként összeépített karnak definiáljuk az egyes alkatrészekhez tartozó fajsúlyát is a későbbi matematikai számításokhoz. A SolidWorks-ben épített robotkart, Matlab/Simulink programba vezetjük át, hol SimMechanics környezetben fogjuk azt vezérelni, az általunk készített Matlab GUI felhasználói felülettel. A Matlab programhoz írt, Peter Corke általi Robotics Toolbox csomagja fogja biztosítani számunkra, a Denavit-Hartenberg (DH), az inverz kinematika, direkt kinematika valamint útvonaltervezés matematikai modellezését. A Matlab GUI, közvetlenül fogja irányítani a Robotics Toolbox által, mind a SimMechanics környezetbe integrált 3D modellt, mind pedig az AL5B robotkart soros porton keresztül. Az útvonaltervező algoritmus századmásodperc pontosan megtervezi az egyes csuklókhöz tartozó szögeket, sebességeket és gyorsulásokat. Ezen adatokat, valós-időben közvetítjük, a soros porton keresztül csatlakoztatott robotkarhoz, valamint a SimMechanics 3D modellhez. A robotkar vezérlésének bemutatásához, pedig egy komplett mozgássort fogunk reprezentálni.

2. ELMÉLETI HÁTTÉR

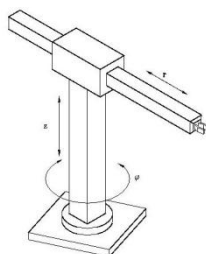
A robot manipulátorok, mint mechanikus szerkezetek meghatározott számú szegmensekből és csuklókból épülnek fel, melyek rendelkeznek tulajdonságokkal, paraméterekkel és pozíciókkal. Az alábbiakban ezeket fogjuk tárgyalni.

2.1 Robotcsuklók

A robot manipulátorokat, mint összetett szerkezeteket, mindig bizonyos számú szegmensek és robotcsuklók alkotják. A robotcsuklók feladata, hogy a szegmensek egymáshoz viszonyított helyzetét meghatározzák és biztosítsák. Ezen szegmenseket a későbbi matematikai számítások végett, a Descartes-féle (x, y, z) koordináta rendszerhez viszonyítva készítjük el. Így határozva meg a tengelyek menti elmozdulást és elfordulást. Egy csukló funkcióját tekintve, lehet:

- a) rotációs
- b) transzlációs (v. prizmatikus)

A rotációs csuklók (R) biztosítják egy szegmens, a másik szegmenshez viszonyított forgó mozgását, mely csukló sematikus ábrázolása a henger. A transzlációs csuklók (T) tekintetében pedig az alap szegmenshez viszonyítva egyenes vonalú elmozdulása van a másik szegmensnek, ezen csuklók sematikus ábrázolása a hasáb.

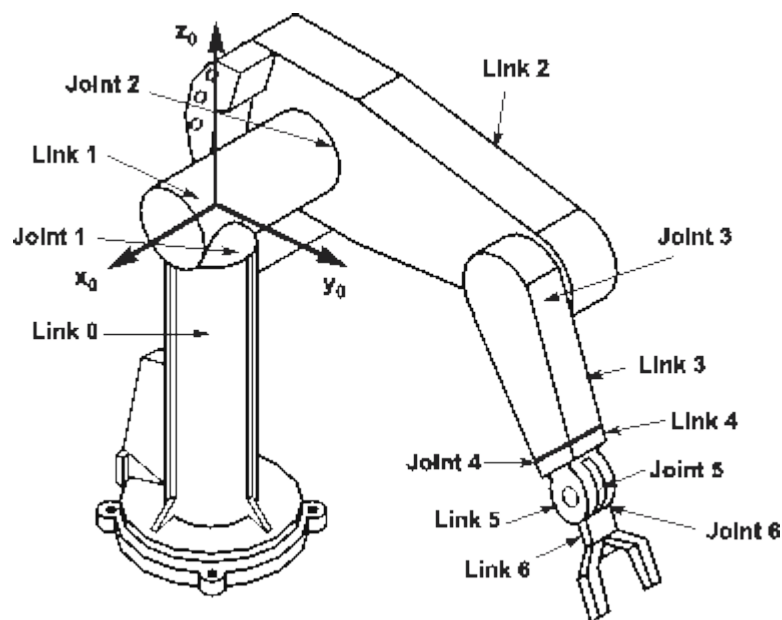


2. ábra Robot manipulátor rotációs és transzlációs csuklóval

Forrás: <http://www.brighthubengineering.com/robotics/29395-base-bodies-of-robots-cylindrical-base-robot/>

2.2 Robotszegmensek

A robotcsuklókat összekötő merev testeket robotszegmenseknek nevezzük. Ezek a robotszegmensek rendelkeznek, mind kinematikai, mind dinamikai paraméterekkel. A kinematikai paraméterek tartalmazzák a szegmensek hosszát (Link), valamint a robotcsuklók és szegmensek egymáshoz viszonyított dőlésszögét (Joint). Ezeket a kinematikai paramétereket a Denavit-Hartenberg féle eljárás alapján határozzuk meg. A dinamikai paraméterek pedig a szegmensek tömegét és tehetetlenségi nyomatékait tartalmazzák.



3. ábra Robot manipulátor szegmensei és csuklói

Forrás: <http://icosym-nt.cvut.cz/odl/partners/fuh/EXAMPLES/eqs2/index.html>

2.3 Robot manipulátorok konfigurációi

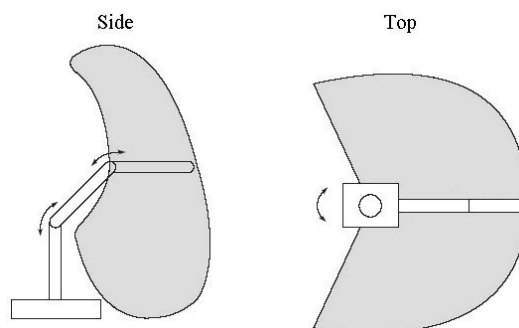
A robot manipulátorokat a rögzített alapzattól kiindulva, n szabadságfokkal rendelkeznek. A szabadságfokok lehetnek rotációsak és translációsak, amelyek 2^n típust határozhatnak meg. Vizsgáljunk egy 3-szabadságfokú manipulátort, hol minden szabadságfok, rotációs vagy translációs. Így a lehetséges variációk száma: $2^3 = 8$.

2.4 Robot manipulátorok munkaterai

A különböző alapkonfigurációk, egyedi munkatereket, azaz bejárható térnagyságot teremtenek, amely természetesen függ a szegmensek hosszától, valamint a csuklók szögelfordulásának minimális és maximális értékétől. A végtelen számú lehetőség közül, jelenleg az öt legelterjedtebb típusú munkateret mutatjuk be (Tukora, 2004):

a) Antropomorf manipulátor (RRR)

A három rotációs csuklóval rendelkező manipulátor munkateré torzított gömb alakú.

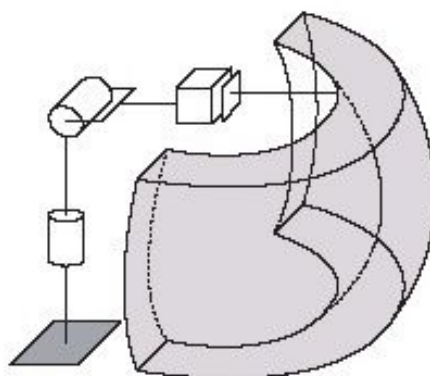


4. ábra Antropomorf manipulátor munkateré

Forrás: Google képtár

b) Gömbi manipulátor (RRT)

A két rotációs csuklóval és egy transzlációs csuklóval rendelkező manipulátor munkateré üreges gömb alakú.

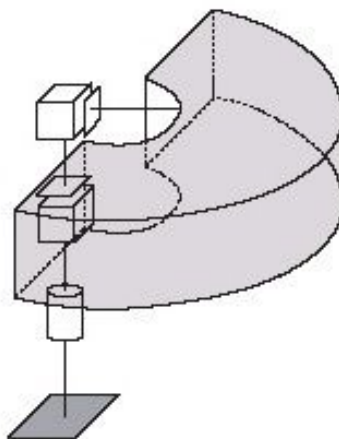


5. ábra Gömbi manipulátor munkateré

Forrás: Google képtár

c) Hengeres manipulátor (RTT)

A két transzlációs csuklóval és egy rotációs csuklóval rendelkező manipulátorok hengergyűrű munkateret biztosítanak.

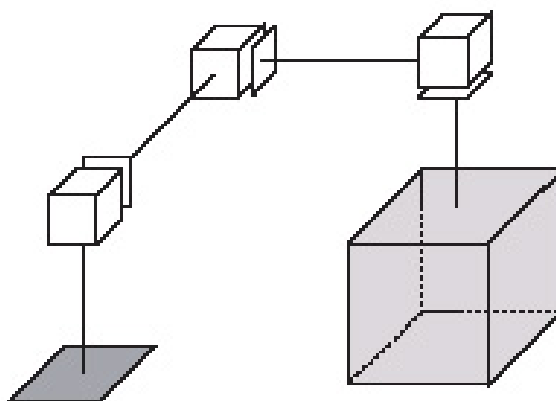


6. ábra Hengeres manipulátor munkatere

Forrás: Google képtár

d) Descartes manipulátor (TTT)

A három transzlációs csuklóval rendelkező manipulátorok, a Descartes-féle koordináta rendszerben könnyen meghatározható, hasáb alakú munkateret biztosítanak.



7. ábra Descartes manipulátor munkatere

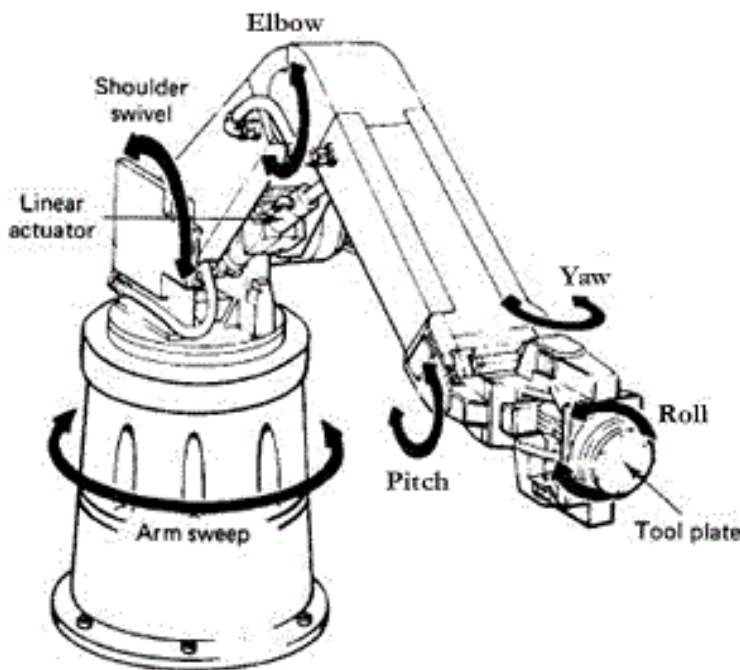
Forrás: Google képtár

e) Párhuzamos manipulátor

A párhuzamos manipulátorok kettő vagy több translációs csuklóval rendelkeznek, melyek mindannyian az effektor mozgását befolyásolják. Általában a nagy terheléseknek kitett manipulátorok esetében használják ezen típust.

2.5 Effektor

Az effektor, a robotkar és munkadarab közvetlen összeköttetését biztosító végszerszám. Az effektor helymeghatározása a munkatérben a DH mátrix alapján számolható. Orientáció tekintetében az effektor pozicionálását értjük a manipulátor munkatérében, a 3 térbeli szög (ψ , θ , ϕ) szerinti elhelyezésben (Mester, Robotmanipulátorok kinematikája, 2003).



8. ábra Effektor pozicionálás

Forrás: <http://www.robotmatrix.org/ArticulateRobotic.htm>

Egy 4-szabadságfokú robot manipulátor, 3 szabadságfokkal el tudja végezni az (x, y, z) pozicionálást a munkatérben, a negyedik szabadságfokkal pedig egy szög szerinti

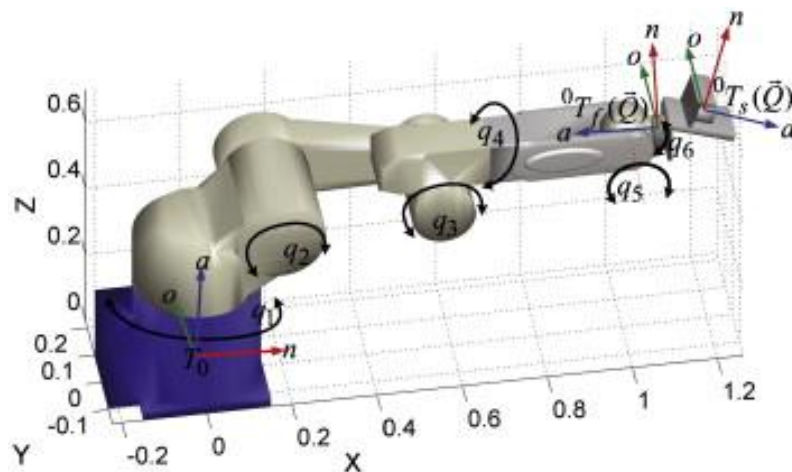
orientációját (ψ). Ez egy 5-szabadságfokú manipulátor esetében az (x, y, z) pozicionálást követően, a negyedik és ötödik szabadságfokkal a (ψ , θ) szögek szerinti orientációt. Egy 6-szabadságfokú manipulátor az első három szabadságfokkal elvégzett x, y, z pozicionálását követően, a 4.-ik, 5.-ik és 6.-ik szabadságfokkal elvégezheti a teljes orientációt (ψ , θ , ϕ) (Mester, Robotika, 2011).

2.6 Csuklókoordináták

A csuklókoordináták (q_i , ahol $i=1,2,...,n$ a csuklók száma) meghatározzák egy csuklóhoz csatlakozó két szegmens egymáshoz viszonyított helyzetét. Rotációs csukló tekintetében ez megegyezik az elforgatási szöggel, míg translációs csuklónál a csukló tengelye mentén történő egyenes vonalú elmozdulással. A csuklókoordináták definiálhatóak az alábbi vektorral:

$$\mathbf{q} = [q_1 \ q_2 \ q_3 \ \dots \ q_n]^T \quad (2.1)$$

Ahol a vektor elemei megegyeznek a szabadságfokok számával.



9. ábra 6-szabadságfokú robotkar csuklókoordinátái

Forrás: <http://www.sciencedirect.com/science/article/pii/S0926580512001483>

A csuklókoordináták minden esetben bizonyos határok között mozognak:

$$q_{i \min} \leq q_i \leq q_{i \max}$$

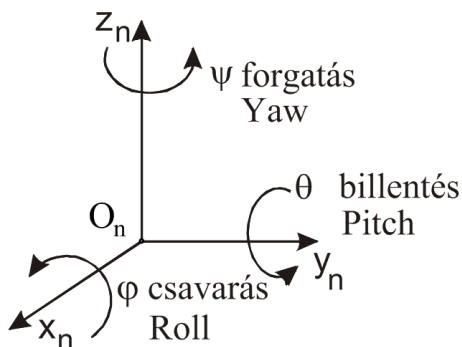
2.7 Világkoordináták

A világkoordináták meghatározzák a manipulátor effektorának pozícióját és orientációját a Descartes-féle koordináta rendszerben, ahol a manipulátor földhöz rögzített pozíciója (x_0, y_0, z_0) , az effektor pozíciója (x_n, y_n, z_n) koordinátákkal, míg az effektor orientációja a módosított Euler-szögekkel (ψ, θ, φ) írható le. Ebben az esetben a világkoordináták vektora (**s**), így írható fel:

$$\mathbf{s} = [x \ y \ z \ \psi \ \theta \ \varphi]^T \quad (2.2)$$

A módosult Euler szögek elnevezései:

ψ – forgatási szög (YAW), θ – billenési szög (PITCH), φ – csavarási szög (ROLL),



10. ábra Euler szögek

Forrás: <http://folder.vts.su.ac.rs/exams/download/oper/no-render/file/Intelligens%20robotok%20es%20rendszerek.pdf>

Hol, a forgatási szög (ψ) a nyugvó koordinátarendszer z tengelye körüli szögelfordulását, a billenési szög (θ), a z tengely körüli szögelfordulást követő új helyzetbe került y tengely körüli szögelfordulást, míg a csavarási szög (φ) az y tengely körüli szögelfordulást követő új helyzetben, az x tengely körüli szögelfordulást határozza meg (Mester, Robotika, 2011).

3. AZ AL5B ROBOTKAR KINEMATIKÁJA

3.1 A Denavit-Hartenberg (DH) mátrix

A Denavit-Hartenberg mátrix meghatározásához, szükséges ismernünk a homogén koordináta-transzformációkat. A homogén koordináta-transzformációk leírják két derékszögű koordinátarendszer közti rotációt valamint a két koordinátarendszer origójának egymástól való távolságát. A homogén koordináta-transzformáció alapján tehát értelmezhetjük a robotkar két egymás melletti szegmenseinek egymáshoz viszonyított helyzetét, így elérve a manipulátor rögzített helyzetétől az effektor helyzetéig. A transzformációs mátrixok (3.1) koordináták szerinti elfordulások és eltolások alapján, az alábbi táblázatban láthatóak:

1. táblázat Homogén transzformációs mátrixok

$$\alpha : \begin{bmatrix} c\alpha & -s\alpha & 0 & 0 \\ s\alpha & c\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(x tengely menti elfordulás)

$$\delta : \begin{bmatrix} c\delta & -s\delta & 0 & 0 \\ s\delta & c\delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(y tengely menti elfordulás)

$$\theta : \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(z tengely menti elfordulás)

$$\mathbf{x} : \begin{bmatrix} 0 & 0 & 0 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(x tengely menti eltolás)

$$\mathbf{y} : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

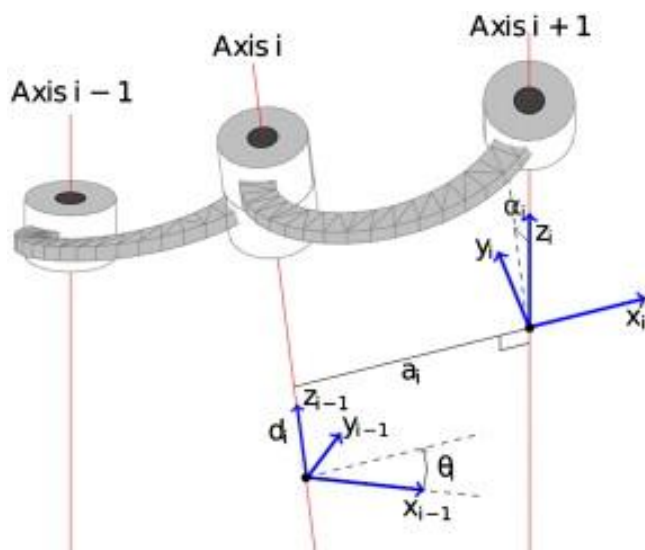
(y tengely menti eltolás)

$$\mathbf{z} : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(z tengely menti eltolás)

Ahol $c = \cos$, és $s = \sin$ (például: $c\theta = \cos\theta$).

Ezen mátrixok alapján meghatározható a robotkar DH mátrixa. Az alábbi kép bemutatja a csuklóhoz tartozó transzformációt:



11. ábra Homogén transzformáció bemutatása

Forrás: <http://www.sciencedirect.com/science/article/pii/S073658451100010X>

Hol a csuklóhoz tartozó mátrix:

$$T_{i+1}^i = Rot(z, \theta_i) Trans(z, d_i) Trans(x, a_i) Rot(x, \alpha_i) \quad (3.8)$$

$$T_{i+1}^i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

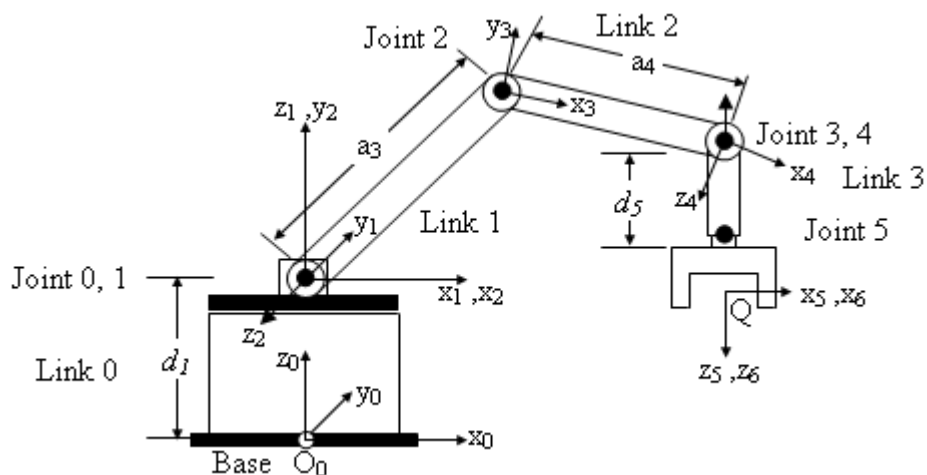
$$T_{i+1}^i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Amennyiben csuklókként meghatároztuk a transzformációs mátrixokat, létrehozhatjuk a robotkart leíró Denavit-Hartenberg mátrixot, a csukló mátrixokat összeszorozva:

$$T_i^0 = T_1^0 T_2^1 \dots T_i^{i-1} \quad (3.11)$$

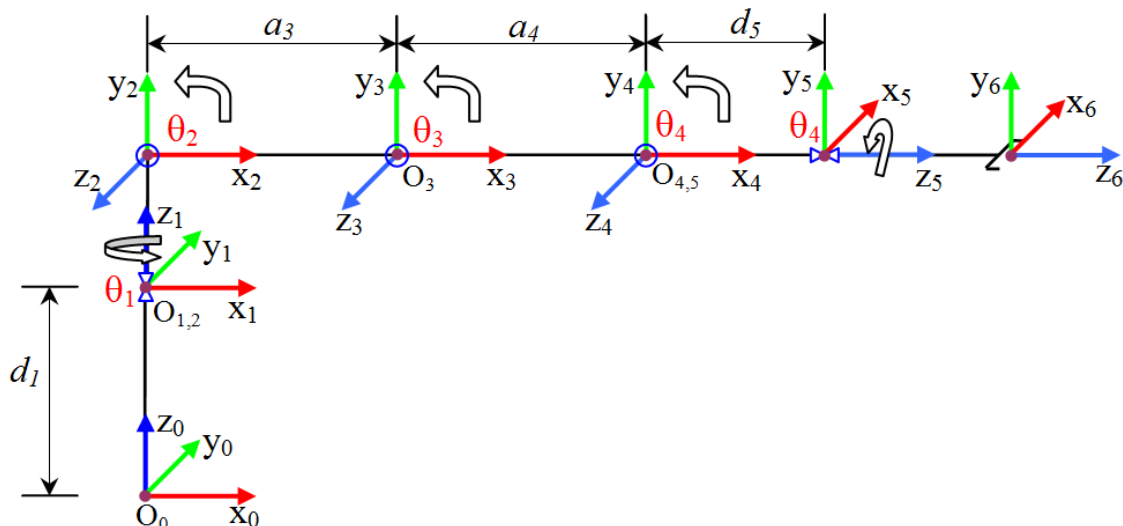
3.2 Az AL5B robotkar DH mátrixa

Az imént tárgyalt meghatározás alapján létrehozhatjuk robotkarunk DH transzformációs mátrixát. Az alábbi ábrákon láthatóak az AL5B robotkar csuklóihoz tartozó koordináták, valamint szegmenseinek hossza (AbuQassem, 2010).



12. ábra AL5B robotkar koordinátái

Forrás: <http://library.iugaza.edu.ps/thesis/92207.pdf>



13. ábra AL5B robotkar DH koordinátái

Forrás: <http://library.iugaza.edu.ps/thesis/92207.pdf>

Ezek alapján felírhatjuk a Denavit-Hartenberg paramétereket:

2. táblázat *AL5B DH paraméter táblázata*

i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	0	0
2	θ_2	0	0	$\pi/2$
3	θ_3	0	a_3	0
4	$(\theta_4 - \pi/2)$	0	a_4	0
5	θ_5	d_5	0	$-\pi/2$
6	θ_6	0	0	0

Ezáltal felírva a DH mátrixokat csuklókként:

3. táblázat *AL5B DH mátrixok csuklókként*

$$\begin{aligned}
 T_1^0 &= \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_2^1 &= \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ \sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_3^2 &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & a_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_4^3 &= \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & a_4 \cos \theta_4 \\ \sin \theta_4 & \cos \theta_4 & 0 & a_4 \sin \theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_5^4 &= \begin{bmatrix} \cos \theta_5 & 0 & -\sin \theta_5 & 0 \\ \sin \theta_5 & 0 & \cos \theta_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_G &= \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

A Denavit-Hartenberg mátrixból ki kell hagyni az effektor nyitására és zárására szolgáló (T_G) mátrixot, így az eredő DH-t a következőképpen határozhatjuk meg:

$$T_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 \quad (3.12)$$

3.3 Direkt kinematika

A Direkt kinematikai számításoknál a manipulátor csuklóinak ismeretében határozzuk meg az effektor pozícióját a világkoordináta rendszerben. Tehát, ha ismerjük a szegmensek egymáshoz viszonyított elforgását (θ_i) akkor meghatározhatjuk az effektor helyzetét a Descartes koordinátarendszerben (x, y, z). Ehhez a manipulátor DH mátrixát használjuk fel. Jelen robotkarunk DH mátrixát a (3.12) egyenletben határoztuk meg, melyet kiszámolva kapjuk a keresett (x, y, z) értékeket.

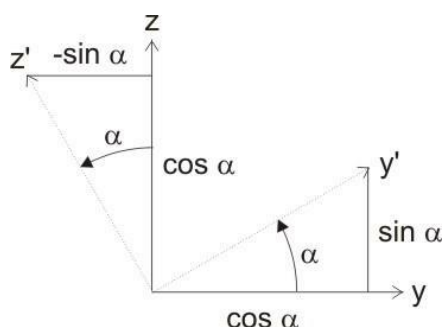
$$T_5^0 = T_1^0 T_2^1 T_3^2 T_2^3 T_2^1 = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & d_x \\ e_{1y} & e_{2y} & e_{3y} & d_y \\ e_{1z} & e_{2z} & e_{3z} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

Mátrixunkban a keresett (x, y, z) koordinátákat a (d_x, d_y, d_z) értékek adják vissza.

3.4 Az effektor orientációja

Az effektor orientációjának meghatározásához, a három koordináta körüli forgás, három különböző forgatómátrixot határoz meg, a Roll-Pitch-Yaw transzformációs mátrixot (Mester, Robotika, 2011).

A „Roll” forgatás, a z tengely körül történik α szöggel, míg x és y tengelyek a síkhoz képest változatlanok maradnak. A z tengely pedig a lapból kifelé mutat.



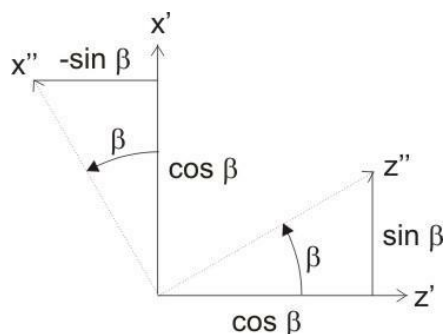
14. ábra "Roll" forgatás a z tengely körül

Forrás: http://tananyagfejlesztés.mik.uni-pannon.hu/images/stories/vegleges_tananyagok/MESTER_ROBOTIKA/Robotika.pdf

Az ezt leíró forgatómátrix pedig:

$$\mathbf{R}(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

A „Pitch” bólintás, a y tengely körül történik β szöggel, míg x és z tengelyek a síkhoz képest változatlanok maradnak. Az y tengely pedig a lapból kifelé mutat.



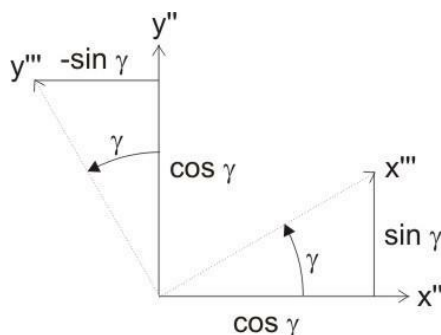
15. ábra "Pitch" bólintás az y tengely körül

Forrás: http://tananyagfejlesztés.mik.uni-pannon.hu/images/stories/vegleges_tananyagok/MESTER_ROBOTIKA/Robotika.pdf

A forgatómátrix pedig:

$$\mathbf{R}(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (3.15)$$

Végül a „Yaw” csavarás, az x tengely körül történik γ szöggel, míg y és z tengelyek a síkhoz képest változatlanok maradnak. Az x tengely pedig a lapból kifelé mutat.



16. ábra "Yaw" csavarás az x tengely körül

Forrás: http://tananyagfejlesztés.mik.uni-pannon.hu/images/stories/vegleges_tananyagok/MESTER_ROBOTIKA/Robotika.pdf

Így a forgatómátrix:

$$\mathbf{R}(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \quad (3.16)$$

A fent bemutatott három rotáció alapján kiszámítható a transzformációs mátrix-szorzat:

$$\mathbf{R}_n^0 = \mathbf{R}(\psi)\mathbf{R}(\theta)\mathbf{R}(\varphi) \quad (3.17)$$

ahol $\alpha = \psi$, $\beta = \theta$, $\gamma = \varphi$.

Az $\mathbf{R}_n^0$ transzformációs mátrixot pedig megkaphatjuk a DH mátrixból, hiszen:

$$\mathbf{T}_5^0 = \mathbf{T}_1^0 \mathbf{T}_2^1 \mathbf{T}_3^2 \mathbf{T}_2^3 \mathbf{T}_2^1 = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & d_x \\ e_{1y} & e_{2y} & e_{3y} & d_y \\ e_{1z} & e_{2z} & e_{3z} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_n^0 & \mathbf{d}_n^0 \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (3.18)$$

$$\mathbf{R}_n^0 = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \quad (3.19)$$

Ezt követően meghatározhatjuk a transzformációs mátrixot, valamint a rotációk értékét:

$$\mathbf{R}_n^0 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \quad (3.20)$$

$$\mathbf{R}_n^0 = \begin{bmatrix} \cos\psi \cos\theta & \cos\psi \sin\theta \sin\varphi - \sin\psi \cos\varphi & \cos\psi \sin\theta \cos\varphi + \sin\psi \sin\varphi \\ \sin\psi \cos\theta & \sin\psi \sin\theta \sin\varphi + \cos\psi \cos\varphi & \sin\psi \sin\theta \cos\varphi - \cos\psi \sin\varphi \\ -\sin\theta & \cos\psi \sin\varphi & \cos\theta \cos\varphi \end{bmatrix} \quad (3.21)$$

A mátrixegyenlet egyes elemei tehát:

$$e_{1x} = \cos \psi \cos \theta \quad (3.22)$$

$$e_{1y} = \sin \psi \cos \theta \quad (3.23)$$

$$e_{1z} = -\sin \theta \quad (3.24)$$

$$e_{2x} = \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi \quad (3.25)$$

$$e_{2y} = \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi \quad (3.26)$$

$$e_{2z} = \cos \psi \sin \varphi \quad (3.27)$$

$$e_{3x} = \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \quad (3.28)$$

$$e_{3y} = \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \quad (3.29)$$

$$e_{3z} = \cos \theta \cos \varphi \quad (3.30)$$

A ψ , θ , φ szögek ezek alapján a következőképpen határozhatóak meg:

$$\psi = \arctg \frac{e_{1y}}{e_{1x}} + 2k\pi \quad (3.31)$$

$$\theta = \arctg \left[\frac{-e_{1z}}{e_{1x} \cos \psi + e_{1y} \sin \psi} \right] + 2k\pi \quad (3.32)$$

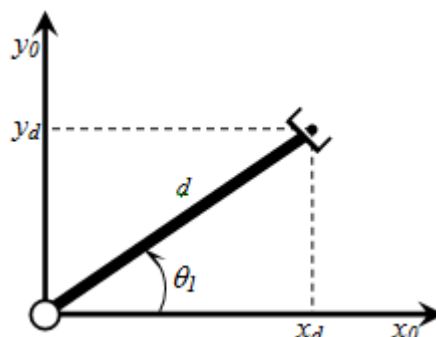
$$\varphi = \arctg \frac{e_{2z}}{e_{3z}} + 2k\pi \quad (3.33)$$

3.5 Inverz kinematika

Az Inverz kinematikai számításoknál a manipulátor effektor pozíciója alapján határozzuk meg a manipulátor csuklóinak egymáshoz viszonyított elforgását a világkoordináta rendszerben. Tehát, ha ismerjük az effektor helyzetét a Descartes

koordinátarendszerben (x, y, z) akkor meghatározhatjuk a szegmensek egymáshoz viszonyított helyzetét (θ_i) (AbuQassem, 2010).

Ezek alapján tehát:

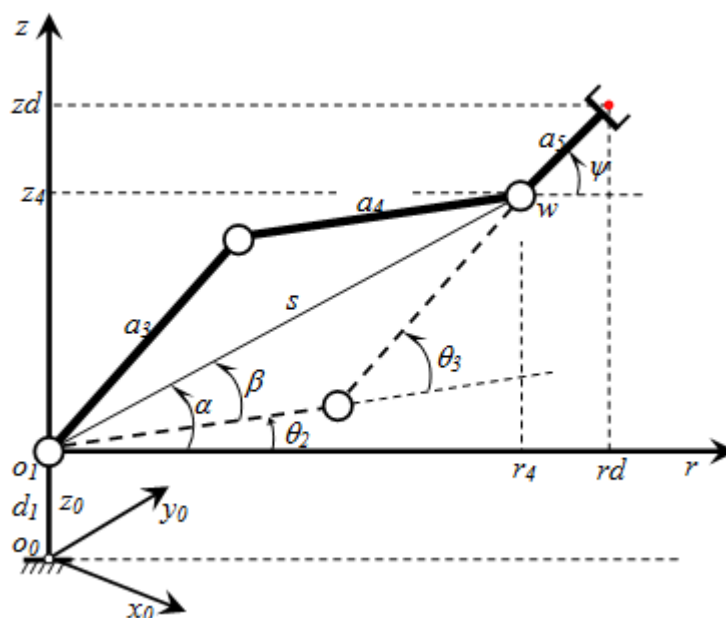


17. ábra Az AL5B robotkat geometriája az x,y koordinátákban

Forrás: <http://library.iugaza.edu.ps/thesis/92207.pdf>

$$d = r = \sqrt{(x_d^2 + y_d^2)} \quad (3.34)$$

$$\theta_1 = \arctg \frac{x_d}{y_d} \quad (3.35)$$



18. ábra Az AL5B robotkat geometriája az r,z koordináták alapján

Forrás: <http://library.iugaza.edu.ps/thesis/92207.pdf>

$$\psi = \theta_2 + \theta_3 + \theta_4 \quad (3.36)$$

$$r_4 = r_d - d_5 \cos(\psi) \quad (3.37)$$

$$z_4 = z_d - d_5 \sin(\psi) \quad (3.38)$$

$$\beta = \text{atan2}(s^2 + a_3^2 - a_4^2, 2a_3s) \quad (3.39)$$

$$\alpha = \text{atan2}(z_4 - d_1, r_4) \quad (3.40)$$

$$s = \sqrt{(z_4 - d_1)^2 + r_4^2} \quad (3.41)$$

$$\theta_2 = \alpha \pm \beta \quad (3.42)$$

$$\theta_3 = \text{atan2}(s^2 - a_3^2 - a_4^2, 2a_3a_4) \quad (3.43)$$

$$\theta_4 = \psi - \theta_2 - \theta_3 \quad (3.44)$$

Ezzel meghatároztuk az x , y , z koordináták és a ψ szög ismeretében a manipulátor csuklóíhoz tartozó θ_1 , θ_2 , θ_3 , θ_4 szögeket.

3.6 Pályatervezés (Trajektória)

A robot manipulátorok pályatervezése magába foglalja a pályatervezést világ- és csuklókoordinátákban, melynek célja az adott pontok közti pálya meghatározása. A pályatervezés a robot manipulátor munkafolyamatától függ, s megvalósítható világkoordinátákban és csuklókoordinátákban (Mester, Robotika, 2011).

A pályatervezés alatt a kezdő és végpontok (A, B) közti útszakaszt határozzuk meg a rendelkezésre álló idő (τ) tekintetében.

$$\begin{aligned} \mathbf{s}(t) &= \mathbf{s}^A + \lambda(t)(\mathbf{s}^B - \mathbf{s}^A) \\ 0 &\leq t \leq \tau \end{aligned} \quad (3.45)$$

$\lambda(t)$ függvény az effektor sebességének törvényszerűségét határozza meg.

A világkoordinátákban meghatározott pályatervezést követően, szükséges azt a csuklókoordinátákban is kivitelezni, melyet a következőképpen tehetünk meg:

$$\begin{aligned} \mathbf{q}(t) &= \mathbf{q}^A + \lambda(t)(\mathbf{q}^B - \mathbf{q}^A) \\ 0 &\leq t \leq \tau \end{aligned} \quad (3.46)$$

Ahol τ a rendelkezésre álló idő, mely alatt $\mathbf{q}^A$ csuklóvektorból $\mathbf{q}^B$ csuklóvektorba kell jutnia az adott csuklónak.

Ezek tekintetében meghatározhatjuk robotkarunk specifikus pályatervezését, az alábbiakban:

$$\begin{aligned} \mathbf{q}(t) &= \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3 \\ &\quad (a \text{ távolság definiálása}) \end{aligned} \quad (3.47)$$

$$\begin{aligned} \dot{\mathbf{q}}(t) &= \mathbf{a}_1 + 2\mathbf{a}_2 t + 3\mathbf{a}_3 t^2 \\ &\quad (a \text{ sebesség definiálása}) \end{aligned} \quad (3.48)$$

$$\begin{aligned} \ddot{\mathbf{q}}(t) &= 2\mathbf{a}_2 + 6\mathbf{a}_3 t \\ &\quad (a \text{ gyorsulás definiálása}) \end{aligned} \quad (3.49)$$

Ezek alapján meghatározhatjuk a kezdő és végpozíció pályáját, valamint a kezdő és végsebességet:

$$\mathbf{q}_0 = \mathbf{a}_0 + \mathbf{a}_1 t_0 + \mathbf{a}_2 t_0^2 + \mathbf{a}_3 t_0^3 \quad (3.50)$$

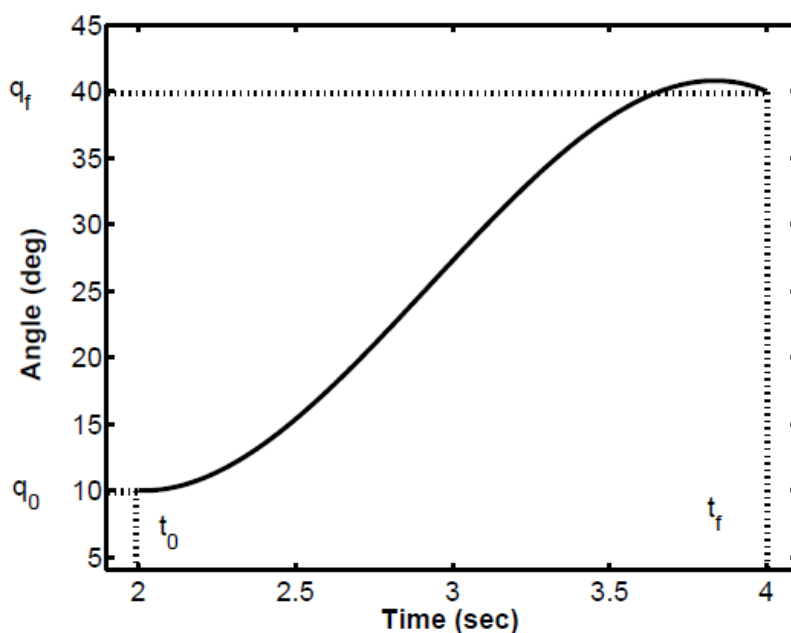
$$\mathbf{v}_0 = \mathbf{a}_1 + 2\mathbf{a}_2 t_0 + 3\mathbf{a}_3 t_0^2 \quad (3.51)$$

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \quad (3.52)$$

$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 \quad (3.53)$$

Így egy négy egyenletből és négy ismeretlenből álló egyenletrendszert kapunk, melyet, az alábbi módon oldhatunk meg:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ q_f \\ v_f \end{bmatrix} \quad (3.54)$$



19. ábra Útvonaltervezés a szög és idő tekintetében

Forrás: <http://library.iugaza.edu.ps/thesis/92207.pdf>

Az AL5B robotkarra ezzel meghatároztunk minden alapvető számítást, hogy azt vezérelni tudjuk. Nincs más hátra, mint a szimuláció, s hogy a vezérlést programba ágyazzuk (AbuQassem, 2010).

4. AZ AL5B ROBOTKAR VEZÉRLÉSE

4.1 Hardware

Az AL5B robotkar mozgását szervo motorok biztosítják. A szervo motorok vezérlését pedig a Lynxmotion cég által forgalmazott SSC-32 vezérlő, mely a számítógéppel, soros porton keresztül kommunikál.



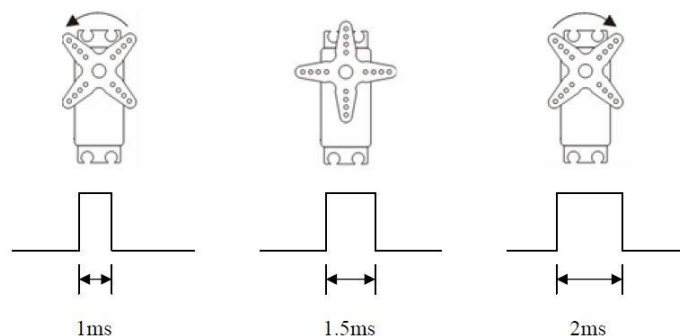
20. ábra Az AL5B robotkat

Forrás: <http://www.robotshop.com/lynxmotion-al5b-robot-arm-combo-kit-4-dof.html>

A szervomotorok pozicionálását az impulzusok hossza határozza meg, melyek estünkben az alábbi táblázatban láthatóak:

4. táblázat Szervo motorok impulzustáblázata

-90°	-45°	0°	45°	90°
500 μ s	1000 μ s	1500 μ s	2000 μ s	2500 μ s



21. ábra Szervomotorok pozicionálása

Forrás: <http://chiayeak.wordpress.com/2010/10/23/how-to-control-the-servo-motor/>

A szervo motorokat az SSC-32 kontrolleren keresztül, az alábbiak szerint vezérelhetjük:

#1 P2000 T1000

Hol a kettős keresztet követően a szervo SSC-32 vezérlő csatornához csatlakoztatott száma szerepel, majd P-t (*Pulse*) követően (2000) az impulzusszámot határozzuk meg. Lényegében a szervo motor pozícióját. Végül pedig T-t (*Time*), mellyel a teljesítés idejét lehet meghatározni ($1000\text{ ms} = 1\text{ sec}$).



22. ábra SSC-32 Servo Controller

Forrás: <http://www.lynxmotion.com/p-395-ssc-32-servo-controller.aspx>

Amennyiben több szervót szeretnénk egyidejűleg mozgatni, úgy, hogy azok időben ugyanabban a pillanatban ériék el végső pozíciójukat, akkor az alábbi kódot alkalmazzuk:

#1 P2000 #3 P1800 T1000

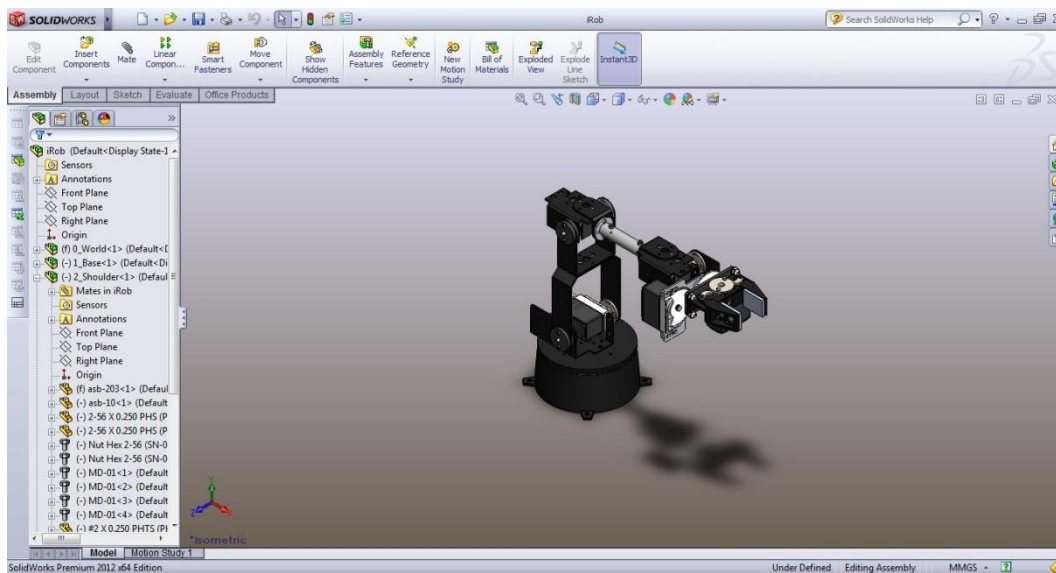
Tehát a két csatorna, s a két impulzus meghatározását követően definiáljuk az időt.

4.2 Software

A robotkar hardware jellemzőit követően a megvalósítás és kivitelezés bemutatása következik. Jelen fejezetben bemutatásra kerül a SolidWorks programban elkészített modell, majd ennek exportálása a SimMechanics programba. A SimMechanics program összeköttetése a Matlab GUI-val, hogy a robotkar valós idejű vezérlése bemutatásra kerüljön. Ezt követően Peter Corke, Robotics Toolbox eszköztárának rövid jellemzése és implementálása következik, valamint a soros port kommunikációnak kivitelezése.

4.2.1 SolidWorks

A robotkar szimulációja SolidWorks programban lett kivitelezve. A SolidWorks-ben külön tekintettel kell lenni, hogy a koordinátarendszer középpontjában helyezkedjen el a robotkar. Ez lesz az exportálást követően a világkoordináta rendszer origója, mely a pontos kivitelezéshez szükséges, hogy az origóban helyezkedjen el.

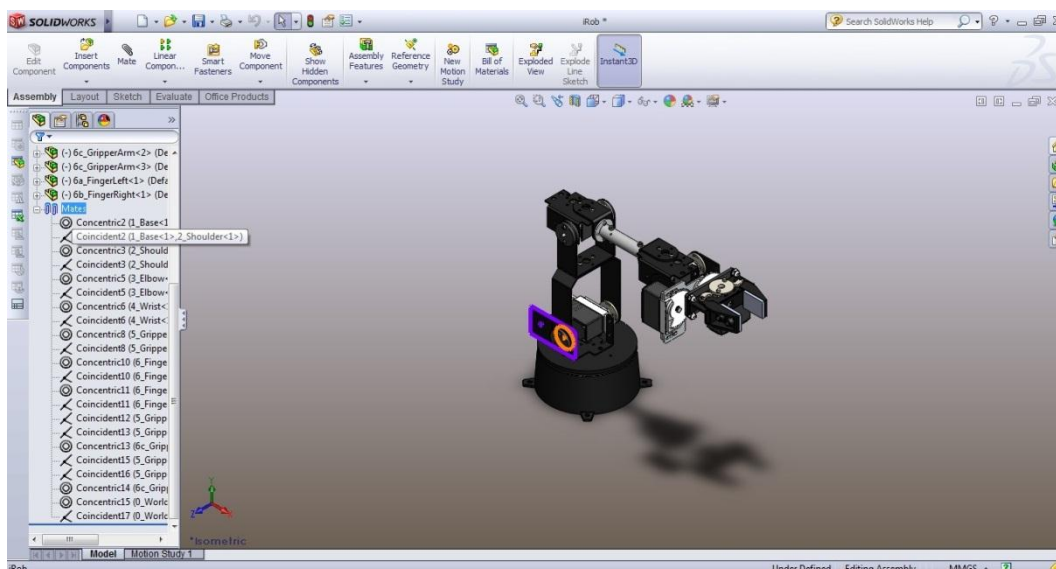


23. ábra AL5B robotkar SolidWorks modellje

Forrás: Saját készítésű kép

A *.sldprt kiterjesztésű részegységeket összeépítve *.sldasm kiterjesztésű blokkokat (szegmenseket) kapunk, melyek összekapcsolásával meghatározhatjuk, hogy milyen irányú elmozdulásokat tudjon kivitelezni robotkarunk. Fontos, hogy a SimMechanics

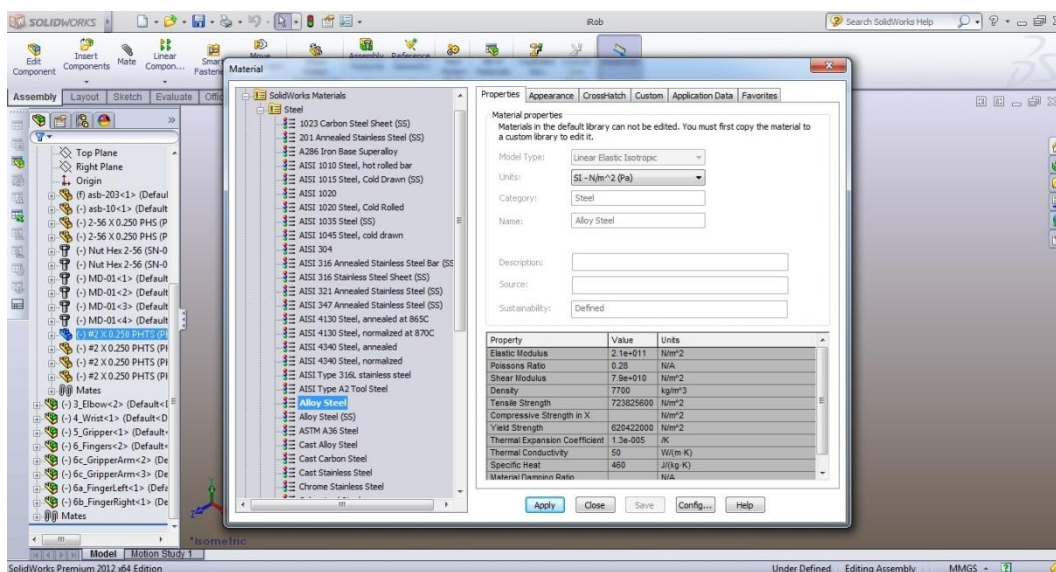
exportáláshoz a SolidWorks modellünkben csak a blokkok (szegmensek) és a köztük lévő kapcsolat legyen a főkönyvtárban létrehozva. A kapcsolatok tekintetében a Coincident – Concentric kapcsolat hozza létre a kinematikában tárgyalt rotációs kapcsolatot, míg a Coincident – Parallel pedig a translációs kapcsolatot (Co., 2002).



24. ábra SolidWorks rotációs kapcsolat (coincident - concentric)

Forrás: Saját készítésű kép

Létrehozva a robotkar szegmenseit és a köztük lévő kapcsolatokat, megadhatjuk az egyes alkatrészek fajsúlyát (Reyes, 2012).



25. ábra SolidWorks modell fajsúly definiálása

Forrás: Saját készítésű kép

Ez kiváló segítséget nyújt, hiszen automatikusan kiszámolja az inerciákat valamint a tömegközéppontokat. A SolidWorks-ben rengeteg anyagfajta van definiálva, így könnyedén kivitelezhető a fajsúlyok megadása az egyes részegységeknek.

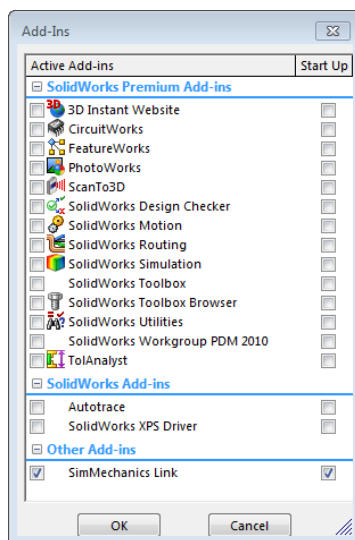
4.2.2 SimMechanics

A SolidWorks-ben elkészített modell-t a SimMechanics-ba exportáljuk. Ehhez le kell töltenünk a MathWorks weboldaláról az exportáláshoz szükséges programokat¹. A telepítéshez a Matlab Command Window-ban az alábbi parancsokat szükséges futtatni:

```
path(path,'download_file_directory')
```

```
install_addon('zip_file_name')
```

Ahol először a letöltött fájlok mappáját állítjuk be, majd a zip fájlt telepítjük. A telepítést követően a SolidWorks-öt regisztrálni szükséges a Matlab programban (MathWorks, 2002). Ezt az `smlink_linksw` paranccsal kivitelezhetjük. A SolidWorks programot megnyitva az *Eszközök/Add-Ins* menüben a párbeszédablakban a SimMechanics Link Add-Ins-t bejelölve, készen áll a SolidWorks az exportálásra.



26. ábra SolidWorks SimMechanics regisztrálása

Forrás: Saját készítésű kép

¹ <http://www.mathworks.com/help/physmod/sm/cad-import.html>

4.2.3 Matlab GUI

A Matlab GUI vezérlőfelületen keresztül irányíthatjuk a SimMechanics modellt valamint soros porton keresztül az AL5B robotkart. A GUI összeköttetése a SimMechanics modellel biztosítja a Direkt és Inverz kinematikai feladatok bemutatását. Ahhoz, hogy ezen összeköttetés valós idejű legyen szükséges néhány beállítást elvégezni (Simakov, 2005). A Matlab-ban deklarálnunk kell a Global WorkSpace-re a SimMechanics-ot irányító változókat, melyeket a GUI-val azonos elnevezésű *.m kiterjesztésű fájlban tehetünk meg a *function iRob_GUI_OpeningFcn(hObject, eventdata, handles, varargin)* résznél (MathWorksInc, 2012). A Global WorkSpace-re az alábbi parancsokkal hozhatjuk létre a változókat:

5. táblázat Matlab - változók deklarálása a Global WorkSpace-re

```
assignin('base','iRob_GUI_handles',handles)

assignin('base','FK_Start_hObject',handles.FK_Start)

assignin('base','IK_Start_hObject',handles.IK_Start)

assignin('base','Basis_Button_hObject',handles.Basis_Button)

assignin('base','Hanoi_Button_hObject',handles.Hanoi_Button)

assignin('base','RT_Button_hObject',handles.RT_Button)
```

Miután ezt megtettük, a SimMechanics modellünkben is be kell állítani, hogy elérje a Global WorkSpace változókat. Ezt a *File/Model Properties* menüben tehetjük meg, hol az egyes eseményekhez létrehozzuk az adott funkciókat.

PostLoadFCN:

```
% Load the GUI
h_iRob_GUI = iRob_GUI;
% Adjust its position
%set(h_iRob_GUI,'Position',[20 20 20 20])
```

StartFCN:

```
% The user may have closed the GUI
try
% Get the string of the button.
%
mystringIK = get(findobj(h_iRob_GUI,'Tag','IK_Start'),'String');
mystringFK = get(findobj(h_iRob_GUI,'Tag','FK_Start'),'String');
mystringBasis = get(findobj(h_iRob_GUI,'Tag','Basis_Button'),'String');
mystringHanoi = get(findobj(h_iRob_GUI,'Tag','Hanoi_Button'),'String');
mystringRT = get(findobj(h_iRob_GUI,'Tag','RT_Button'),'String');
%
% See if the string is "Start"
% If it is, that implies that the start wasn't driven by the GUI.
% Thus we need to update the GUI by making a callback to the Start/Stop function.
%
if strcmp(mystringIK,'Start')
iRob_GUI('IK_Start_Callback',IK_Start_hObject,[],iRob_GUI_handles)
end
%
if strcmp(mystringFK,'Start')
iRob_GUI('FK_Start_Callback',FK_Start_hObject,[],iRob_GUI_handles)
end
%
if strcmp(mystringBasis,'Start')
iRob_GUI('Basis_Button_Callback',Basis_Button_hObject,[],iRob_GUI_handles)
end
%
if strcmp(mystringHanoi,'Start')
iRob_GUI('Hanoi_Button_Callback',Hanoi_Button_hObject,[],iRob_GUI_handles)
end
%
if strcmp(mystringRT,'Start')
iRob_GUI('RT_Button_Callback',RT_Button_hObject,[],iRob_GUI_handles)
end
end
```

StopFCN:

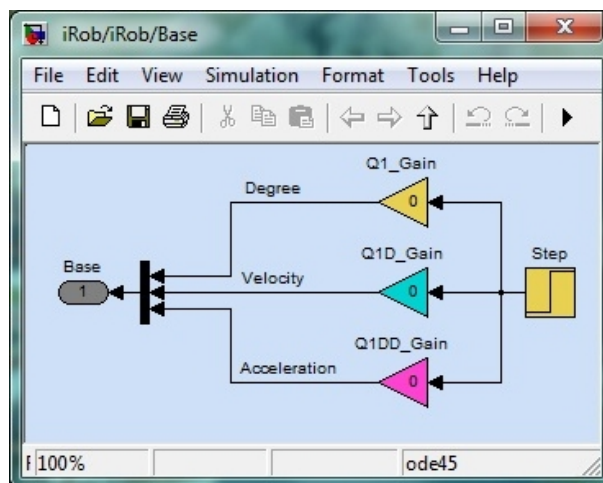
```
% The user may have closed the GUI
try
%
% Get the string of the button.
%
mystringIK = get(findobj(h_iRob_GUI,'Tag','IK_Start'),'String');
mystringFK = get(findobj(h_iRob_GUI,'Tag','FK_Start'),'String');
mystringBasis = get(findobj(h_iRob_GUI,'Tag','Basis_Button'),'String');
mystringHanoi = get(findobj(h_iRob_GUI,'Tag','Hanoi_Button'),'String');
mystringRT = get(findobj(h_iRob_GUI,'Tag','RT_Button'),'String');
%
```

```
% See if the string is "Stop"
% If it is, that implies that the stop wasn't driven by the GUI.
% Thus we need to update the GUI by making a callback to the startstop function.
%
if strcmp(mystringIK,'Stop')
iRob_GUI('IK_Start_Callback',IK_Start_hObject,[],iRob_GUI_handles)
end
if strcmp(mystringFK,'Stop')
iRob_GUI('FK_Start_Callback',FK_Start_hObject,[],iRob_GUI_handles)
end
if strcmp(mystringBasis,'Stop')
iRob_GUI('Basis_Button_Callback',Basis_Button_hObject,[],iRob_GUI_handles)
end
if strcmp(mystringHanoi,'Stop')
iRob_GUI('Hanoi_Button_Callback',Hanoi_Button_hObject,[],iRob_GUI_handles)
end
%
if strcmp(mystringRT,'Stop')
iRob_GUI('RT_Button_Callback',RT_Button_hObject,[],iRob_GUI_handles)
end
%
end
```

CloseFCN:

```
% Try to close the GUI
try
close(h_iRob_GUI)
end
```

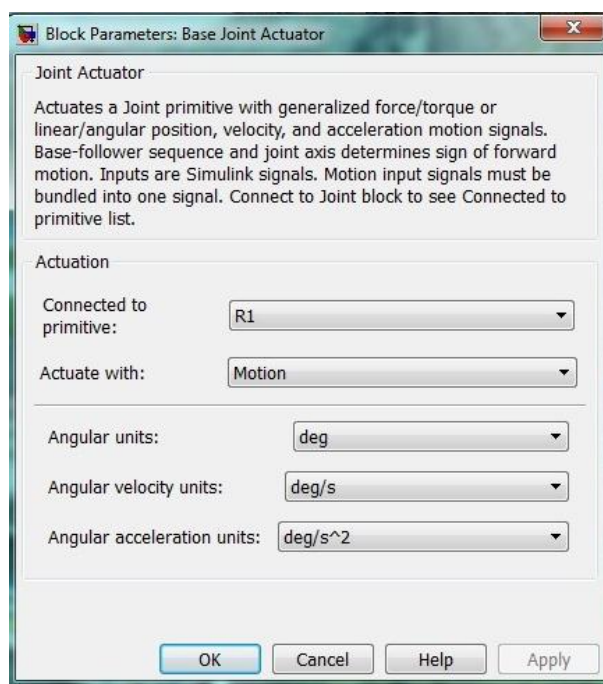
Következő lépésben, a Matlab-ban kapott információkat szükséges a SimMechanics Gain moduljaihoz juttatni, mely a Step lépés módosításával, a robotkar szimulációt a megfelelő pozícióba vezérli.



29. ábra SimMechanics - Gain vezérlése Matlab GUI-ból

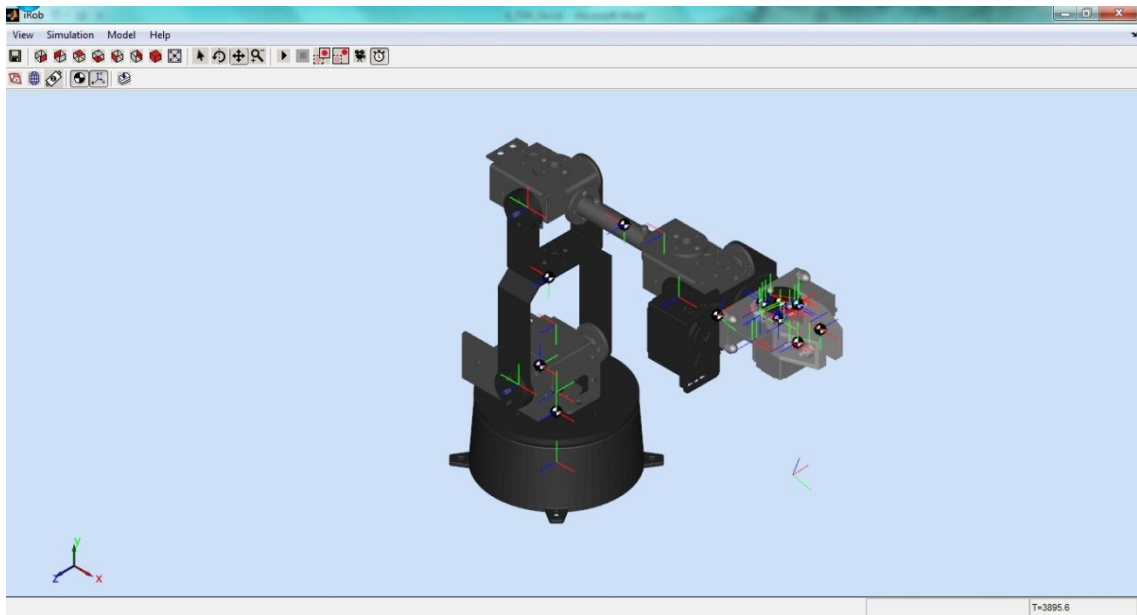
Forrás: Saját készítésű kép

A *Step* értéke alapértelmezetten 1-re van állítva, míg *Gain* a szorzó, melynek értékét a GUI-ban határozzuk meg. A szimulációban szükséges, hogy folyamatosan értéket kapjanak az egyes csuklók a gravitáció végett, ehhez használtuk a *Step* lépést. A *Gain* értéke a csuklókhoz van közvetítve (*Joint Actuator*).



30. ábra SimMechanics - Joint Actuator

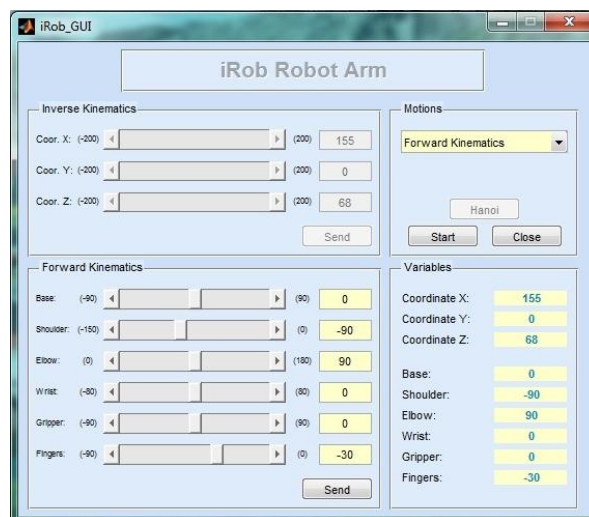
Forrás: Saját készítésű kép



31. ábra Az AL5B SimMechanics modell-je

Forrás: Saját készítésű kép

Az összeköttetést megvalósítva a GUI kialakítása következett. Jelen dolgozat, az Inverz és Direkt kinematikát mutatja be elsődlegesen. Így a Matlab GUI a következőképpen lett kialakítva:



32. ábra A robotkar vezérlőfelülete

Forrás: Saját készítésű kép

A GUI *.m fájl-ban, az értékek beállítását követően a „Send” gombra kattintva küldhetjük el az adatokat a modellnek a alábbi paranccsal:

```
set_param([bdroot 'iRob/Base/Q1_Gain'],'Gain',value);
```

Nincs más hátra, mint definiálni a Matlab-ban a robotkart, melyhez Peter Corke Robotics Toolbox csomagját használjuk fel.

4.2.4 Robotics Toolbox

Peter Corke Robotics Toolbox (Corke, 2011) kiegészítője, segít definiálni robotunkat a Matlab programban. A szegmensek alapján létrehozza a robotot, a Denavit-Hartenberg paramétereket, Direkt és Inverz kinematikát, valamint az útvonaltervezést.

A robot szegmenseinek létrehozását az alábbi módon kivitelezhetjük:

```
assignin('base','d1',68);
assignin('base','a3',119);
assignin('base','a4',128);
assignin('base','d5',90);
clear L
%%
%          th      d      a      alpha
L(1) = Link([ q1      d1      0      pi/2      0], 'standard');
L(2) = Link([ q2      0      a3      0      0], 'standard');
L(3) = Link([ q3      0      a4      0      0], 'standard');
L(4) = Link([ q4      0      0      -pi/2      0], 'standard');
L(5) = Link([ q5      d5      0      0      0], 'standard');
L(6) = Link([ q6      0      0      0      0], 'standard');
```

Hol az *assignin* paranccsal létrehozott Global WorkSpace változókat, a *Link* paranccsal a DH paraméterekké alakítjuk. Ezt követően, a szegmensek alapján létrehozzuk a robotunkat:

iRob = SerialLink(L, 'name', 'iRob', 'manufacturer', 'Unimation', 'comment', 'viscous friction; params')

A robot létrehozása után, az alábbi parancsokkal vezérelhetjük robotkarunkat:

Direkt kinematikai számítás:

$$Tf = f_{\text{kine}}(iRob, qz)$$

Ahol Tf visszaadja az $iRob$ robot, qz vektorában található csuklók szögei által az x, y, z koordinátákat a Descartes-féle koordináta rendszerben.

Inverz kinematikai számítás:

$$T1 = \text{transl}(217, 217, 177);$$

$$q = i_{\text{kine}}(iRob, T1, q0, M)$$

Ahol q a kiszámított szögek vektora, $T1$ a translációs mátrix, $q0$ a kiszámított szögekhez tartozó tolerancia (általában 0), valamint M a hat szabadságfoknál kevesebb szabadságfokú robotkarok csuklóinak meghatározása, a számítási feladat tekintetében.

Például: $M = [1 \ 1 \ 1 \ 1 \ 1 \ 0]$ hol a számításba az első öt szabadságfokot beleérti, a hatodikat nem. Mivel a hatodik jelen robotunkban a Gripper nyitását és zárását szolgálja.

Útvonaltervezés:

$$q = j_{\text{traj}}(qz, qr, t);$$

Mely függvénynél q vektor visszaadja a qz és qr kezdő és végpontok közti szögeket, sebességet és gyorsulást a t időpillanatban. qz és qr szögvektorok, melyek tartalmazzák a kezdő és végpozíció csuklóértékeit.

4.2.5 Soros-port kommunikáció

A Robotics Toolbox alkalmazásai segítettek, hogy meghatározzuk a robotkar matematikai hátterét, viszont mindezt közvetíteni szükséges robotkarunknak. A

hardware résznél tárgyaltuk, hogy milyen paranccsal közvetíthető a jel a szervók felé. Előbb definiálnunk, s megnyitnunk szükséges a soros portot:

```
Serial = serial('COM3');  
set(Serial,'Terminator','CR');  
set(Serial,'BaudRate',115200);  
set(Serial,'DataBits',8);  
set(Serial,'Parity','none');  
set(Serial,'StopBits',1);  
set(Serial,'FlowControl','none');  
fopen(Serial);  
get(Serial);
```

Ezt követően, az útvonaltervezés előzetes meghatározása alapján, folyamatosan küldjük az új adatokat (másodpercenként 100-szor):

```
while i < ((ts / lepes)+1),  
    i = i+1;  
    serv = round(q(i,2));  
    s0 = num2str(1500 + (serv * 10));  
    stest = strcat( '#0 P' , s0)  
    fprintf(Serial,'%s\n',stest);  
    pause(lepes);  
end
```

A program bezárását megelőzően pedig lezárjuk a portot:

```
fclose(serial)  
delete(Serial)  
clear Serial
```

5. ÖSSZEFOGLALÁS

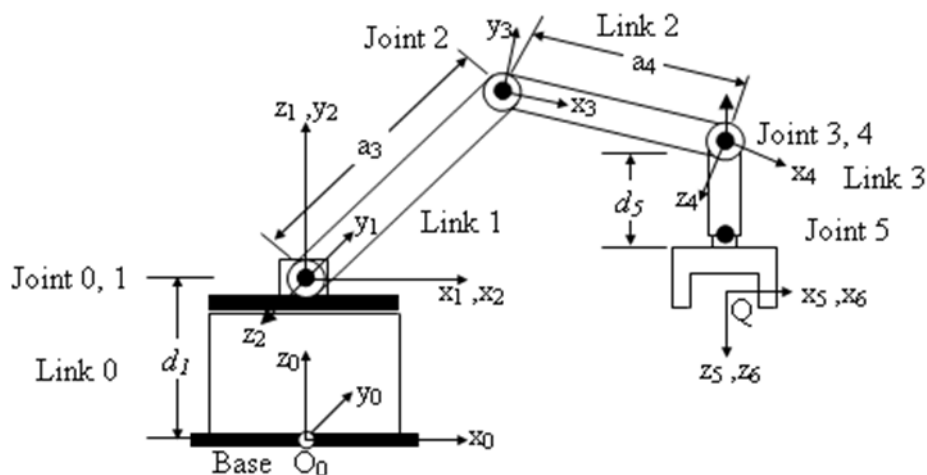
Dolgozatom lényege, hogy elsajátítsam, s példával bemutassam egy robotkar kinematikai hátterét, szimulációjának kiépítését, valamint vezérlését. A dolgozat alatt felmerült problémák megoldása további fejlődésre sarkallott, de összességében tekintve, az eltervezett feladatokat kivitelezettnek mondhatom. Habár sok esetben a megvalósítási folyamat alatt eltévelyedtem, s általam megvalósíthatatlannak tartott problémákkal találkoztam, végül sikerült mindenre megoldást találni. A végső konklúzió, további munkára ösztönöz, így a jövőre való tekintettel, szeretném összekötni robotkaromat egy kamerával, hogy a digitális képfeldolgozás témakörét is mélyebb szinten elsajátíthassam. Ezáltal a robotkar, a különböző színű tárgyakat felismerve, azokat szelektálná, s egy előre meghatározott helyre szállítaná. A Robotika témaköre rengeteg lehetőséget nyújt a fejlődésre, hisz megköveteli a matematikai, a fizikai, az informatikai és villamosságtani ismereteket. A jövő szerves része a Robotika, mely számos újítást igényel még, habár rohamosan fejlődik a technológia. A dolgozat megerősítette bennem, hogy a jövő Robotika kutatásainak szerves részévé szeretnék válni, mely további tanulásra sarkall. Összességében tekintve, jelenlegi első dolgozatom jó néhány új dolgozat témát alakított ki bennem, a digitális képfeldolgozáson keresztül, a hangvezérlésen át, egészen az infobionikai alkalmazásterületig. Úgy vélem nem ez volt az utolsó dolgozatom.

6. IRODALOMJEGYZÉK

- AbuQassem, M. R. (2010). *Simulation and Interfacing of 5 DOF Educational Robot Arm*. Islamic University of Gaza.
- Anonymous. (2012). *The History of Computing Project*. Forrás: <http://www.thocp.net/reference/robotics/robotics.html>
- Co., S. (2002). *SolidWorks Student Workbook*.
- Corke, P. (2011). *Robotics, Vision & Control*. Springer.
- MathWorks. (2002). *SimMechanics*.
- MathWorksInc. (2012). *Creating Graphical User Interfaces*.
- Mester, G. (2003). *Robotmanipulátorok kinematikája*. Egyetemi jegyzet.
- Mester, G. (2011). *Robotika*. Typotex.
- Nocks, L. (2001). *A Brief History of Robotics since 1950: An entry from Gale's Science and Its Times*.
- Reyes, A. (2012). *Beginner's Guide to SolidWorks*.
- Tukora, B. (2004). *Robotok irányítása*. Főiskolai jegyzet.

7. FÜGGELÉK

Az AL5B robotkar matematikai háttérének megvalósításához az alábbi szegmens méretekkel és csuklók szöghatáraival dolgoztam:



33. ábra Függelék - Robotkar paraméterei

Forrás: <http://library.iugaza.edu.ps/thesis/92207.pdf>

6. táblázat Az AL5B robotkar paraméterei

d1	68 mm
a3	119 mm
a4	128 mm
d5	100 mm
q1 (Base)	$-90^\circ \leq q1 \leq 90^\circ$
q2 (Shoulder)	$-60^\circ \leq q1 \leq 90^\circ$
q3 (Elbow)	$-90^\circ \leq q1 \leq 90^\circ$
q4 (Wrist)	$-90^\circ \leq q1 \leq 90^\circ$
q5 (Gripper)	$-90^\circ \leq q1 \leq 90^\circ$
q6 (Gripper open/close)	$-90^\circ \leq q1 \leq 0^\circ$

A robotkar programja az alábbi fájlokat használja és tartalmazza:

7. táblázat *Az iRob program fájljai*

iRob_GUI.fig	Matlab GUI grafikus paramétereit tartalmazza
iRob_GUI.m	Matlab GUI funkcióit tartalmazza
iRob.mdl	SimMechanics modell
iRob.xml	SolidWorks fájl, amely a SimMechanics paramétereit tartalmazza
*.STL	SimMechanics modell szegmenseinek paramétereit tartalmazza
deg2rad.m	Szögből radiánba konvertál
rad2deg.m	Radiánból szögbe alakít
link.m	Robotics Toolbox – szegmenseket hoz létre
SerialLink.m	Robotics Toolbox – létrehozza a robotot a szegmensek által
fkine.m	Direkt kinematikát számol
ikine.m	Inverz kinematikát számol
jtraj.m	Trajektóriát számol
transl.m	transzlációs mátrixot alakít át