

Debreceni Egyetem
Műszaki Kar
Mechatronika Tanszék



FANUC hibrid delta-parallel ipari robot kinematikai analízise és szingularitást detektáló programkód fejlesztése MATLAB környezetben

Szerző(k) **Lakatos Lili Sára BSc hallgató**

Témavezető(k): **Korsoveczki Gyula**
Tanszékvezető: **Prof. Dr. Husi Géza**

Debrecen
2023.

Tartalomjegyzék

Bevezetés	3
1. Irodalomkutatás	5
1.1. Szabadságfokok számítása	5
1.1.1. Csuklókoordináták típusai	5
1.1.2. Robot szabadságfokainak meghatározása.....	6
1.2. Robot koordináták típusai.....	6
1.2.1. Csuklókoordináták jellemzői	7
1.2.2. Világkoordináták jellemzői	7
1.3. Transzformációs mátrix felírása	8
1.3.1. Denavit-Hartenberg paraméterek	10
1.3.2. Denavit-Hartenberg transzformációs mátrix.....	11
1.4. Szingularitás fogalma	12
2. Feladatmegvalósítás.....	14
2.1. Direkt irányú kinematika	14
2.1.1. Delta rész direkt irányú kinematikája.....	14
2.1.2. Hibrid rész direkt irányú kinematikája	24
2.2. Inverz irányú kinematika	26
2.2.1. Hibrid rész inverz irányú kinematikája.....	26
2.2.2. Delta rész inverz irányú kinematikája.....	29
3. Mérési eredmények	32
3.1 Kinematikai számítási és mérési eredmények összevetése.....	33
3.1.1 Első pozíció	33
3.1.2 Második pozíció.....	35
3.1.3 Harmadik pozíció	35
3.1.4 Negyedik pozíció.....	36
3.1.5 Ötödik pozíció	37
3.1.6 Képek a robotpozíciókról.....	38
3.2 Szingularitás detekció tesztelése	39
3.2.1 Szingularitás: Direkt kinematika delta rész $z_{g1}=z_{g3}$	39
3.2.2 Szingularitás: Direkt kinematika delta rész $z_{g2}=z_{g3}$	40
3.2.3 Szingularitás: Direkt kinematika delta rész $z_{g1}=z_{g2}=z_{g3}$	40

3.2.4	Szingularitás: Direkt kinematika hibrid rész.....	41
3.2.5	Szingularitás: Inverz kinematika hibrid rész.....	42
3.2.6	Képek a szingularitási pozíciókról.....	43
4.	Összefoglaló	45
5.	Irodalom jegyzék	47
	Mellékletek.....	49
1.	melléklet: Direkt kinematikai feladat	49
2.	Melléklet: Inverz kinematikai feladat.....	54

BEVEZETÉS

A tudomány fejlődéseknek köszönhetően az elmúlt évek során az ipari technológiák gyökerestől megváltoztak. A túlnépesedés és a tömeggyártás megjelenése miatt az ipar nem tudja teljes mértékben kielégíteni az fogyasztást és a megnövekedő igényeket tisztán emberi munkaerő használatával. Ebből kifolyólag ezen probléma megoldása érdekében mára egyre több gyártósoron és cégnél alkalmaznak ipari robotokat a munkafolyamatok gyorsabb, pontosabb és költséghatékonyabb elvégzése érdekében.

Az ipari robotok két típusát különböztetjük meg, a sorosan- és a párhuzamosan láncolt kinematikájú manipulátorokat. A sorba kapcsolt robotok bár nagy munkatérrel rendelkeznek és viszonylagos felépítésükhöz képes nagy tömegek mozgatására alkalmasak, azonban a maximálisan elérhető sebességük alacsony, mint ahogy az ismétlési pontosságuk is. Ezzel szemben a párhuzamos kinematikájú úgynevezett parallel robotok a jelenlegi ipari igényeket sokkal jobban kielégítő felépítéssel bírnak. Bár kisebb a munkatérük és a terhelhetőségük, ezzel szemben annál nagyobb precizitásra és gyorsaság elérésére alkalmasak [1]. Ezek a tulajdonságok, ahogyan az előbb taglalva volt, a gyártásifolyamatok gyorsítása szempontjából alkalmazva hatalmas előnyt nyújtanak és egyre népszerűbbé válhatnak.

A párhuzamos manipulátorok két fő típusát különböztetjük meg, vannak alap és hibrid jellegű robotok. A hibrid típusúak, ahogy arra a nevük is utal, egy párhuzamosan és egy sorba kapcsolt kinematikai részből tevődnek össze. Emiatt egy elég összetett és matematikailag bonyolult direkt és inverz kinematikai rendszerrel rendelkeznek, aminek nehézsége miatt eddig nem sokan foglalkoztak a tervezésével és modellezésével. Azonban ennek ellenére a Debreceni Egyetem Műszaki Karának tulajdonában van egy hibrid parallel robot, amit lehetőségem van szakdolgozatom során tanulmányozni és szimulálni. Úgy vélem, hogy ennek a robot típusnak a részletes ismerete és a működésének tanulmányozása és szimulálása a szakdolgozatomat hiánypótlóvá és kiemelkedővé teszi robotipar és a mechatronika területén belül.

Ezenfelül az Ipar 4.0-ban a manipulátorok munkaterületének és viselkedésének vizsgálata adott szimulációs környezetben egy modern gyártósor fejlesztése vagy elkészítése előtt a főlegesen plusz költségek és hibák elkerülése érdekében elvárt. Ez azonban nem csak az

ipari környezetre vonatkozólag érvényes, hanem egy robot gyártó cég esetében egy új manipulátor prototípus legyártása előtt is elengedhetetlen, hogy szimulációk segítségével megismerjük annak kinematikáját és működését.

A robotok kinematikai modelljének ismerete és a szerkezetek mozgás predikciója nem csak a mozgási terület meghatározása miatt fontos, hanem a használat közbeni problémás pozíciók, úgynevezett szingularitási pontok elkerülése érdekében. Közismert tény, hogy a robotok úgy nevezett ktrecekben vannak egy gyártóterületen alkalmazva hiszen, ha a gépek bizonyos helyzetekbe kerülnek a mozgásuk kiszámíthatatlanná fog válni. Ez mai napig egy nagyon komoly probléma, ami miatt az ipari robotok még a cellák alkalmazásával is veszélyesen tudnak működni, azonban az ilyen szingularitási helyek elemzésével és kiszűrésével csökkenthető az ipari balesetek száma.

A fent említett okokból következőleg látható, hogy a szakdolgozatban taglalásra kerülő tematika elengedhetetlen kompetencia a 21. században. Mint a kutatás területén, mint pedig az iparban létfontosságú szerepe van a robotmodellek szakszerű szimulációjára. Véleményem szerint a szakdolgozatba kifejtett feladat az általam több éve tanulmányozott mechatronika teljes mértékű kiteljesedése. A robotikában jelen van mindaz, ami a mechatronikát felépíti és egyedivé teszi. Egy manipulátorban megtalálhatóak a gépészeti elemek, mint maga a gép karosszériája, a villamos alkatrészek, ami a rendszer elektronikájában játszik szerepet, és az informatika, ami a rendszer irányításáért és vezérléséért felel, Azonban a felsorolt tudomány területek önmagukban nem elegek ahhoz, hogy egy olyan összetett rendszert alkossanak, mint egy robot, szükséges hozzájuk a matematika és a logika, ami kapcsolatot teremt a részek között és segítségével létre hozhatóak megfelelő szimulációk és validált robotmodellek. Mivel jelenleg sok helyen a gyártósorok robotizációja még csak folyamatban van, ezért ebből következőleg a közel jövőben egyre több helyen lesz elvárt az iparban olyan emberek alkalmazása, akik képesek ezen kihívások megoldására.

1. IRODALOMKUTATÁS

1.1. SZABADSÁGFOKOK SZÁMÍTÁSA

A szabadságfokok meghatározzák, hogy egy adott pont vagy egy kiterjedt test milyen irányokba fordulhat, illetve mozdulhat el, adott koordináta rendszerek esetén [2]. Jelen esetben a kutatás során vizsgált robotot megfeleltethető kiterjedt testnek, amit a térben, azaz az x , y és z koordináták mentén vizsgálunk. A térben egy kiterjedt test szabadságfokainak száma maximálisan 6 lehet, mivel három irányba végezhet transzlációs, illetve rotációs mozgást [3]. A szabadságfokainak meghatározásához Grubler-Kutzbach egyenletet szükséges alkalmazni [4]. A megfelelő adatok használatával lehetőség van az (1.) egyenletbe való behelyettesítés után kiszámítani a manipulátor szabadságfokait [5].

$$DOF = \lambda(l - j - 1) + \sum f_i - f_{passzív} \quad (1.)$$

Ahol az egyenletben szereplő ismeretlenek rendre:

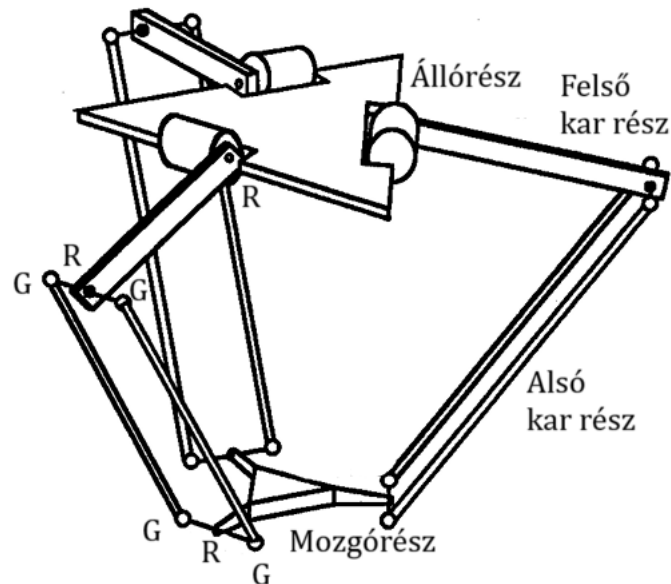
- λ : Merev test szabadságfoka, ami síkban $\lambda = 3$ térben pedig $\lambda = 6$
- l : A rendszerben elhelyezkedő szegmensek száma (beleszámítva a bázist is)
- j : A rendszerben megtalálható csuklók száma
- $\sum f_i$: Az összes csuklókhoz tartozó szabadságfokok számának összegzése
- $f_{passzív}$: A passzív szabadságfokok száma

1.1.1. CSUKLÓKOORDINÁTÁK TÍPUSAI

A robotokban elhelyezkedő csuklók között megkülönböztetünk aktív és passzív csuklókat, illetve ezeken belül még különféle típusokat. Az aktív csukló, azaz az a csukló, amelyet közvetlenül vezérlünk lehet transzlációs és rotációs, vagy esetenként akár univerzális is. A passzív, vagyis a nem közvetlenül mozgatott csuklók rendre lehetnek gömb, rotációs és akár transzlációs vagy univerzális csuklók is [6] [7]. A rotációs és transzlációs csuklók egy szabadságfokkal rendelkeznek, míg a gömb csuklók 3-mal.

A kutatásom során egy három karú hibrid delta robotot modelleztem, aminek delta része a RGG típusú kinematikai láncsal rendelkezik, azaz az aktív csuklói rotációs csuklók lesznek, míg a paralelogrammát mozgató részei, tehát a passzív elemei a gömb csuklók lesznek. A hibrid rész esetén három sorba kapcsolt kinematikai láncban elhelyezkedő

rotációs csukló található. A kutatás során használt hibrid delta robot aktív csuklóit minden esetben rotációs csuklók alkotják. A rotációs csuklók esetén a robot egyik szegmense körül el fog tudni fordulni a másik csuklóhoz tartozó szegmens. Ezt követően a dokumentumban az előzőekben meghatározott manipulátor típusra vonatkozóan lesznek a különféle adatok és jellemzők számítva.



1. ábra: Delta robot felépítése [6]

1.1.2. ROBOT SZABADSÁGFOKAINAK MEGHATÁROZÁSA

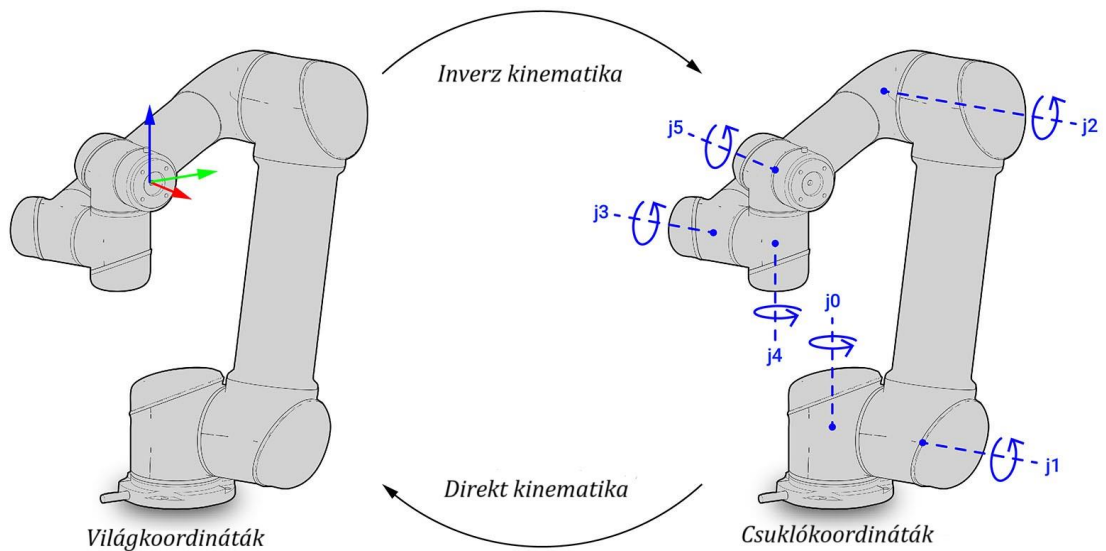
A kutatásban alkalmazott delta robotra vonatkozó adatok a következők: $\lambda = 6$, $l = 14$, $j = 18$, $\sum f_i = 3 \cdot 1 + 12 \cdot 3 + 3 \cdot 1 = 42$ és $f_{passzív} = 6$ (a hat darab alsó kar rész miatt) így pedig

$$DOF = 6(14 - 18 - 1) + 42 - 6 = 6 \quad (2.)$$

Ebből fakadóan a használt manipulátor 6 szabadságfokkal fog rendelkezni, tehát a térbeli Descartes koordináta rendszer mindhárom tengelye mentén képes elmozdulni és elfordulni is.

1.2. ROBOT KOORDINÁTÁK TÍPUSAI

Két típusú robotkoordináta típust különböztetünk meg egymástól ezek a csukló- és a világhkoordináták, melyek egymás inverzei és számítható az egyik értéke, ha ismert a másiké [8].



2. ábra: Direkt és inverz kinematika összehasonlítása [9]

1.2.1. CSUKLÓKOORDINÁTÁK JELLEMZŐI

A csuklókoordináták meghatározzák egy adott kinematikai pár két szegmensének egymáshoz viszonyított helyzetét. Ahogy előzőleg említve volt ez az érték rotációs csuklók esetén az elfordulás szögével, transzlációs csuklók esetén pedig az adott tengelyen történő elfordulással egyezik meg [10]. A csuklókoordináták jele: q_i . A csuklókoordináták vektorának elemeinek száma megegyezik a robot csuklóinak számával:

$$q = [q_1 \ q_2 \ q_3 \ \dots \ q_n]^T \quad (3.)$$

Az általam vizsgált robotnál a hat rotációs csukló esetén hat darab szögelfordulásból fog állni ez a vektor: $q = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6]^T$ Fontos megjegyezni még, hogy minden csuklókoordináta két adott, előre meghatározott érték között változhat.

1.2.2. VILÁGKOORDINÁTÁK JELLEMZŐI

A világkoordináták értékei meghatározzák a robot effektorjának, vagy szerszámközéppontjának helyét és helyzetét egy a robot álló részének középpontjához rögzített Descartes féle derékszögű koordinátarendszerben. Maga az effektor helye meghatározható az x , y és z koordináták segítségével. Az effektor helyzetének leírása a módosított Euler szögek segítségével történik. A módosított szögek rendre ψ , θ és φ . A forgatási szög ψ , (angolul: *Yaw*) megmutatja, hogy az álló koordináta rendszer z tengelye

körül milyen szöggel van elfordítva a szerszámhoz tartozó koordináta-rendszer. A második rotáció az úgynevezett billentési szög θ (angolul: *Pitch*), ami az új helyzetbe pozícionált referenciakoordináta-rendszer y tengelye körüli forgatását jelenti. A harmadik és egyben utolsó rotáció szögét szokás csavarási szögnek φ (angolul: *Roll*) nevezni. Ezzel az előzőekben már módosított orientációjú koordináta-rendszer x tengelye körüli forgatást szokás jelölni

A világkoordináták vektora általánosan felírható a következőképpen:

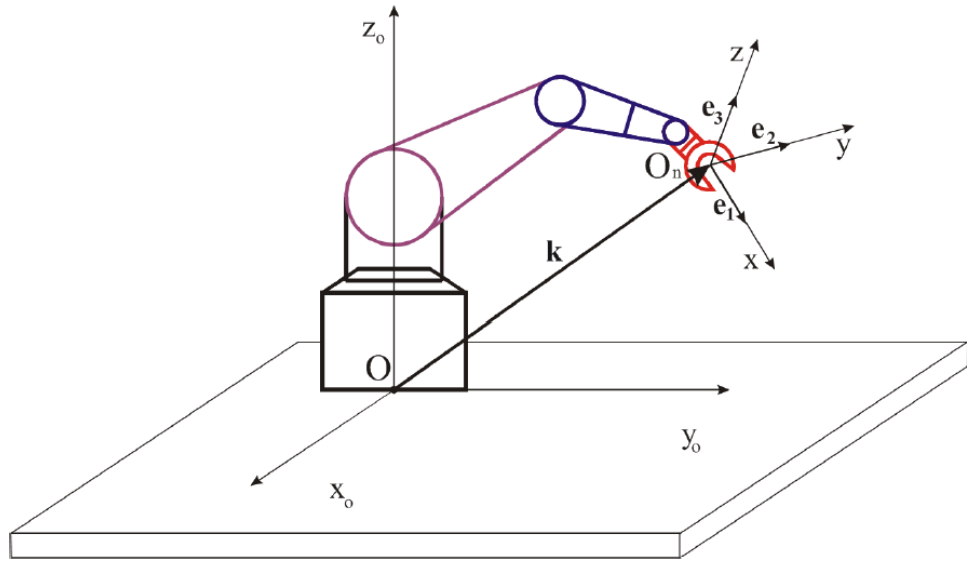
$$s = [x \ y \ z \ \psi \ \theta \ \varphi]^T \quad (4.)$$

1.3. TRANSZFORMÁCIÓS MÁTRIX FELÍRÁSA

Sorba kapcsolt kinematikájú robotok esetében a referenciakoordináta rendszer és a szerszám koordináta rendszer között a Homogén transzformációs mátrix adja meg a kapcsolatot [10]. Ez meghatározza a rotációt és a két koordináta-rendszer origójának távolságát [11]. A Rotációs mátrixnak nevezzük, azt mátrixot, amely tartalmazza szerszámhoz rendelt mozgó koordináta-rendszer egység vektorjainak merőleges vetületét a referencia koordináta-rendszerben:

$${}^nR_0 = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \quad (5.)$$

Ha veszek egy adott $\bar{p}$ vektort, ami egyértelműen meghatározza egy általam választott P pont helyzetét a mozgó koordináta rendszerben, akkor a következőképpen írható fel az $\bar{r}$



3. ábra: Referencia koordinátarendszer meghatározása [10]

vektor, ami a P pont helyzetét a referenciakoordináta rendszerben jelöli:

$$\bar{r} = {}^nR_0 \cdot \bar{p} + \bar{k} \quad (6.)$$

Ahol:

- $\bar{r}$: A P pont helyzetvektora a referenciakoordináta rendszerben
- $\bar{p}$: A P pont helyzetvektora a mozgó koordináta rendszerben
- $\bar{k}$: A mozgókoordináta rendszer origójának helyzetvektora a referenciakoordináta rendszerben
- nR_0 : A rotációs mátrix

Az egyenlet felírva a megfelelő behelyettesítésekkel:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \quad (7.)$$

Az egyszerűbb felírás érdekében bevezetem a Homogén transzformációs mátrixot:

$$H = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.)$$

Az (59.) egyenletbe visszahelyettesítve a következő alakhoz jutok:

$$\bar{r} = H \cdot \bar{k} \quad (9.)$$

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ 1 \end{bmatrix} \quad (10.)$$

A Homogén transzformációs mátrix kapcsolatot teremt a két koordinátarendszer között, megadja azok egymáshoz viszonyított helyét és helyzetét. A mátrix meghatározásához szükség van az egymást követő csuklók között kapcsolatot teremtő Denavit-Hartenberg transzformációs mátrixok előállításához.

1.3.1. DENAVIT-HARTENBERG PARAMÉTEREK

A Denavit-Hartenberg módszer alapja, hogy bármely két koordinátarendszer két forgó és két haladó mozgással egymásba vihető. Ezeket a lépéseket szokás Denavit-Hartenberg paraméterek nevezni. Ennek megfelelően minden robotcsuklóra rá tudok ültetni egy derékszögű koordináta rendszert a következő szabály alapján [12]:

- z_i tengely minden esetben a csukló irányában helyezkedjen el
- x_i tengelyt úgy veszem fel, hogy az $i-1.$ és az $i.$ csukló közös normálisába essen és az $i-1.$ csuklóból az $i.$ csukló irányába mutasson
- y_i tengelyt úgy választom, hogy kielégítse a feltételt, hogy a koordinátarendszer jobbsodrású

Ahhoz, hogy két csukló koordináta rendszert egymásnak meg tudják feleltetni az előbb említett Denavit-Hartenberg paramétereket kell alkalmaznom. Jelen esetben minden csukló rotációs így a jelölések az alábbiak felelnek meg [13]:

- d_i : Az $i-1.$ és $i.$ csuklótengelyek normálisai között az $i-1.$ csukló z tengelye mentén mért távolság
- a_i : Az $i-1.$ és $i.$ csuklótengelyek közös normálisának az $i.$ csukló x tengelyen mért hosszúság

- α_i : Az i . csukló x tengelye körül jobbsavar irányú forgatás mértéke, azaz $i-1$. és i . csuklók z tengelyei által bezárt szög értéke
- q_i : Rotációs csuklók esetén az $i-1$. csukló z tengelye körül jobbsavar irányú forgatás mértéke, azaz $i-1$. és i . csuklók x tengelyei által bezárt szög értéke

A Denavit-Hartenberg paraméterek és lépések az alábbi homogén koordináta transzformációs mátrixokkal fognak megegyezni:

$$D(q_i) = \begin{bmatrix} \cos q_i & -\sin q_i & 0 & 0 \\ \sin q_i & \cos q_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.)$$

$$D(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12.)$$

$$D(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13.)$$

$$D(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14.)$$

1.3.2. DENAVIT-HARTENBERG TRANSZFORMÁCIÓS MÁTRIX

A négy műveletet összeszorozva megkapom a Denavit-Hartenberg transzformációs mátrixát az adott csuklóknak:

$${}^{i-1}D_i = D(q_i) \cdot D(d_i) \cdot D(a_i) \cdot D(\alpha_i) \quad (15.)$$

Tehát a beszorzást elvégezve és a mátrixokat behelyettesítve a Denavit-Hartenberg transzformációs mátrix két egymást követő rotációs csukló esetén a következőképpen fog kinézni [14]:

$${}^{i-1}D_i = \begin{bmatrix} \cos q_i & -\sin q_i \cos \alpha_i & \sin q_i \sin \alpha_i & a_i \cos q_i \\ \sin q_i & \cos q_i \cos \alpha_i & -\cos q_i \sin \alpha_i & a_i \sin q_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16.)$$

Ez az összefüggés felírható jelen feladat esetében a három rotációs csuklóra, így három transzformációs mátrixot fogok kapni ebben az esetben. Ezeket a mátrixokat összeszorozva megkapom a Homogén transzformációs mátrixot, ami kapcsolatot teremt a referencia koordináta rendszer és a szerszám koordináta rendszer között. Fontos, hogy a szorzás sorrendje nem mindegy, az egymást követő koordinátarendszerekhez tartozó mátrixokat kell szigorúan ebben a sorrendben összeszorozni [15]:

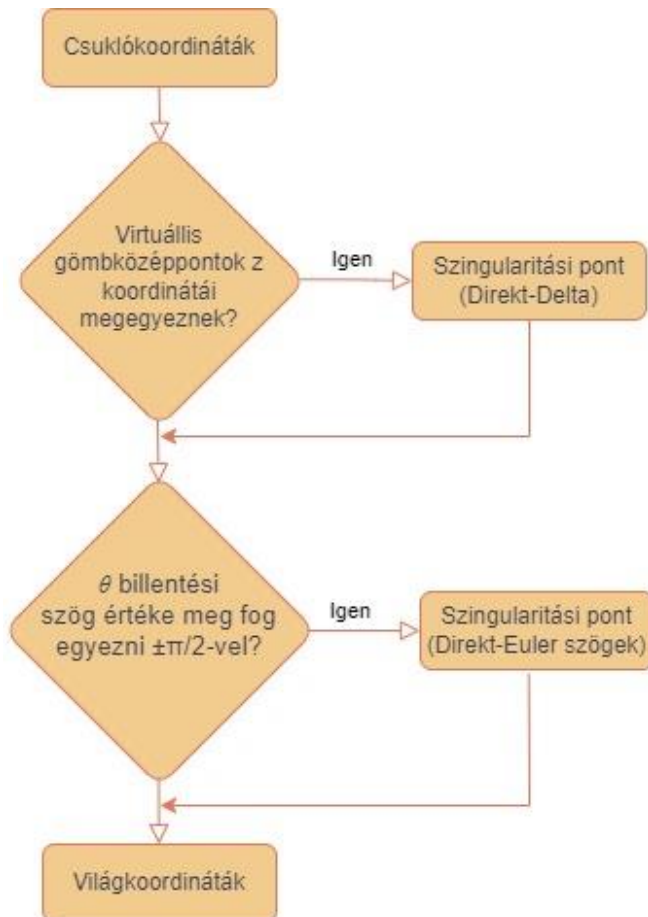
$${}^{i-1}T_n = {}^0D_1 {}^1D_2 \dots {}^{n-2}D_{n-1} {}^{n-1}D_n \quad (17.)$$

Ez a mátrix is 4×4 -es lesz, mivel az összes mátrix, amit összeszorozunk, hogy megkapjuk, az is minden esetben 4×4 formátumú. Ez a mátrix definiálni fogja a szerszámhoz tartozó koordinátarendszer pozícióját és orientációját. Ahogy arról már kezdetben is beszéltem, a Homogén transzformációs mátrix első három sora és oszlopa a koordináták helyzetét a mátrix utolsó oszlopának pedig első három sora az origónak a helyét határozza meg.

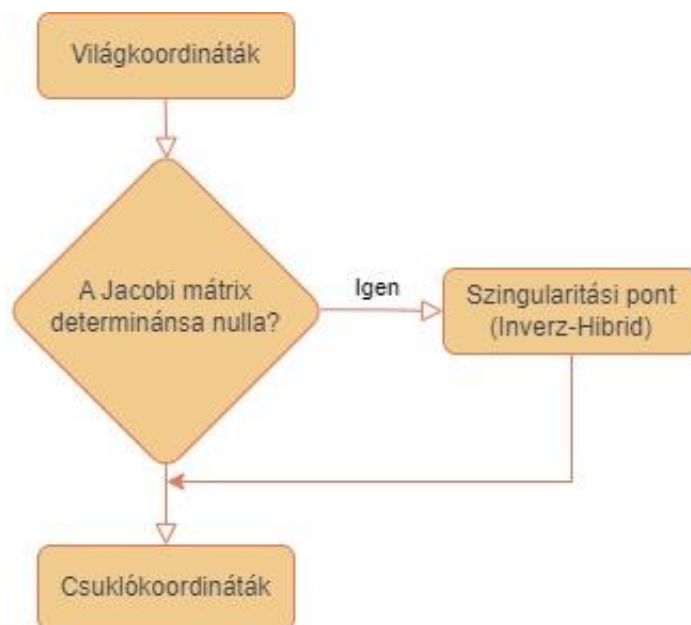
1.4. SZINGULARITÁS FOGALMA

A szingularitás alatt a robotikában egy olyan eshetőséget értünk, amikor egy adott pillanatban az effektor szabadságfoka eltérő lesz a rendszer szabadságfokának számától. Tehát a rendszer veszteni fog a szabadságfokaiból, és nem lesz képes bizonyos típusú mozgások végrehajtására [16]. A szingularitás matematikai értelemben a robotot működtető programkód jellegéből adódik. Általánosan egy program működésében hibát a nullával való osztás idézhet elő, amit a kód nem tud végrehajtani, emiatt a robot mozgása koordinálhatatlanná és veszélyessé válhat.

Az általam fejlesztett program kód egyedisége, hogy képes detektálni a szingularitást, majd ezt követően pedig képes felismerni, hogy a robot melyik részén és mi okozhatta a hibát. Ezeken felül pedig a szingularitás jelenléte mellett képes tovább számolni és meghatározni a kívánt koordináta értékeket.



4. ábra: Direkt rész folyamatábrája [Saját forrás]



5. ábra: Inverz rész folyamatábrája [Saját forrás]

A fentebb látható ábrák a programkód folyamatábráit jelenítik meg.

2. FELADATMEGVALÓSÍTÁS

A feladatmegvalósítás célja, hogy az elméleti háttér ismeretek segítségével létrehozzak egy olyan matematikai modellt a MATLAB program használatával, ami tökéletesen leképezi az általam használt FANUC M-11A 0.5A típusú robot direkt és inverz kinematikai számítását és mozgásformáját.

2.1. DIREKT IRÁNYÚ KINEMATIKA

A direkt kinematikai feladat megoldásának célja, hogy a robot hat aktív rotációs csuklójának elmozdulásából meghatározza effektor végpontjának koordinátáit a világkoordinátarendszerben:

$$s = f(q) \quad (18.)$$

A delta rész esetén egy olyan egyenlethez jutok, aminek a megoldása nem lineáris, tehát több megoldása is létezik [17]. Ezzel szemben a hibrid rész kiszámítása könnyen és gyorsan elvégezhető a Denavit-Hartenberg transzformációs mátrix kiszámításával.

2.1.1. DELTA RÉSZ DIREKT IRÁNYÚ KINEMATIKÁJA

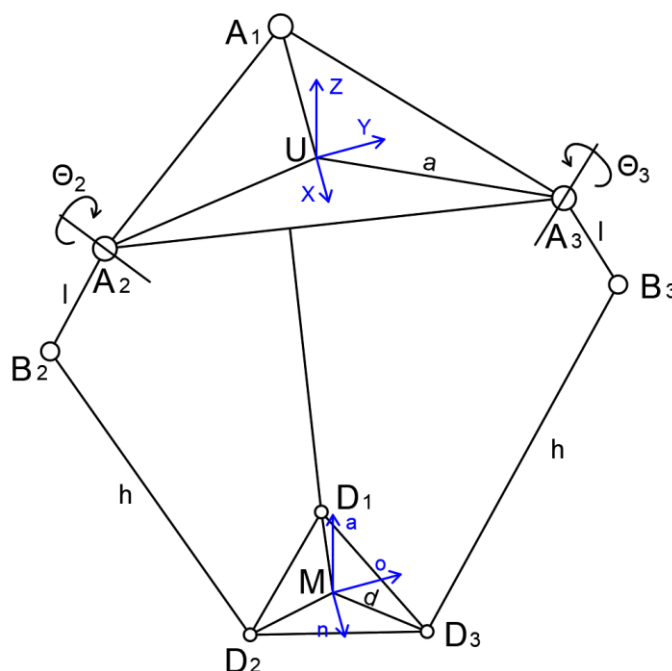
A delta robot egy párhuzamosan kapcsolt kinematikával rendelkező manipulátor. Jelen esetben a kutatásom során vizsgált hibrid robot delta része RSS kinematikával rendelkezik, ahol az R azért van alá húzva, mert a manipulátor aktív csuklójára utal. Mivel ezek a típusú robotok általában valamilyen felső részhez vagy plafonhoz rögzített ezért a rendszer világkoordinátájának origóját az álló rész középpontjával tekintik azonosnak és $U(x,y,z)$ pontként hivatkoznak rá. Az általam vizsgált Fanuc robot x koordinátája úgy van beállítva, hogy az első csukló irányával ellentétes irányba mutasson, a z tengely pedig úgy, hogy a mozgó platformmal ellentétes irányba, azaz felfelé mutasson. A mozgó koordinátarendszert a mozgó platform középpontjára helyezem és az origóját $M(n,o,a)$ pontnak definiálom. Ebben az esetben n tengely x -el, az a pedig z tengellyel lesz párhuzamos és azonos értelmű, tehát a két koordináta rendszer orientációja meg fog egyezni.

A (3.) ábrán látható θ_i szögek az aktív rotációs csukló pillanatnyi helyzetét mutatják szögben, maguk a csuklók pedig A betűvel vannak jelölve. Ezek a direkt kinematikai feladat során bemeneti értékeként fognak jelen lenni. Maga a θ akkor lesz nulla

pozícióban, ha felső kar rész vízszintessel 65° -os szöget zár be. Forgások pozitív irányának a z tengely felé történő mozgást definiálják a robot felhasználó kézikönyvében.

A (3.) ábrán látható egyéb jelölések a következők:

- a : az aktív csuklók távolsága a világkoordináta rendszer origójától
- d : az alsó gömb csukló távolsága a mozgó koordinátarendszer origójától
- l : a felső kar rész hossza
- h : az alsó kar rész hossza

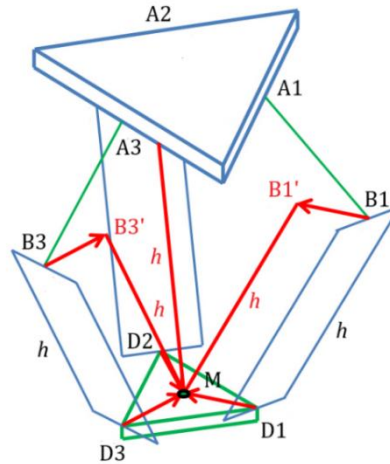


6. ábra: Delta rész kialakítása [Saját forrás]

A delta rész direkt kinematikájának megoldása során a cél a mozgó platformon értelmezett mozgó koordinátarendszer orientációjának és pozíciójának meghatározása. Fontos megjegyezni, hogy jelen esetben elég a mozgókoordinátarendszer helyzetét meghatározni, mivel a párhuzamos kialakításnak köszönhetően az orientációja nem fog változni a referenciakoordinátarendszerhez képest.

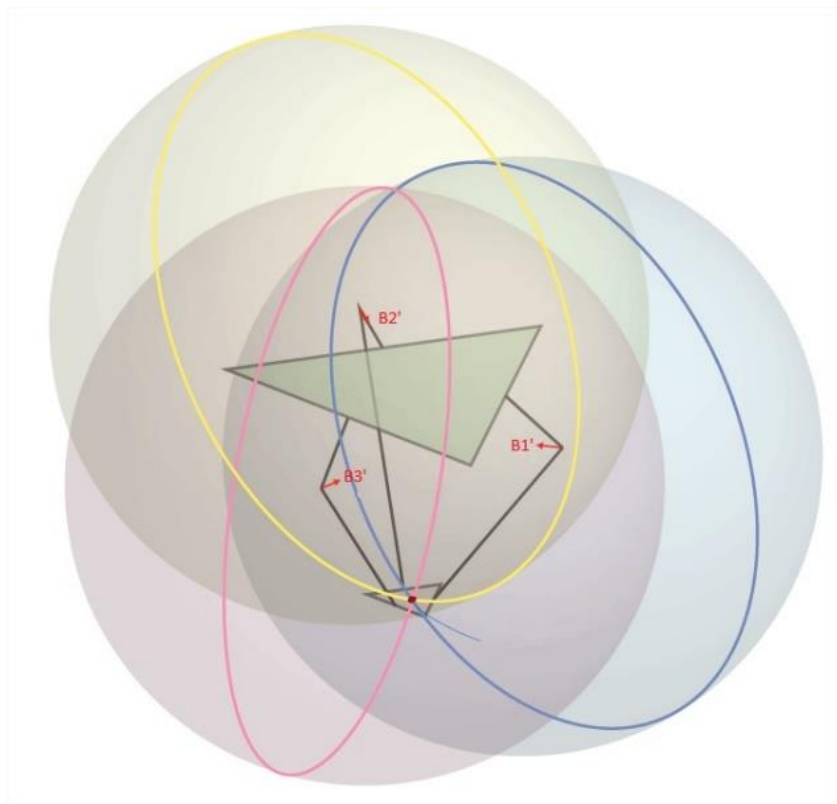
A direkt feladat kiszámítása azon alapszik, hogy ahogy az a (4.) ábrán is látszik minden kar esetében az alsó kar részénél a D_i csuklók egy adott gömb felülete mentén mozognak, aminek a rádiusza h és a középpontja pedig B_i . Bevezetem a B'_i pontokat úgy, hogy a B_i, B'_i, D_i és M pontok egy virtuális paralelogrammát képezzenek [18]. Mivel a $\overline{B_i B'_i}$ vektorok párhuzamosak lesznek az $\overline{M D_i}$ vektorokkal így a $\overline{B'_i M}$ vektorok hossza meg fog egyezni minden pillanatban h -val, azaz az alsó karrész hosszával. Mivel a delta robot rész

esetén a direkt kinematikai feladat célja meghatározni mozgó platform középpontjának, vagyis az M pontnak a helyzetét.



7. ábra: A direkt kinematikai feladat megoldási elve [17]

Az általam definiált B_i' pontok segítségével, úgynevezett virtuális gömb középpontokat hozhatok létre, amelyeknek sugara itt is h -val lesz egyenlő. Amennyiben létre hozom mindhárom virtuális gömb középpontot és megtalálom azt a pontot, ahol ezek metszik egymást, akkor igazából így az M pontot fogom definiálni [19].



8. ábra: Virtuális gömbök [20]

A feladat megoldását azzal kezdem, hogy meghatározom a világkoordináta-rendszer origójából a B_i pontokba mutató vektorokat. Ehhez első lépésként mindhárom esetben felírható a következő összefüggés:

$$\overrightarrow{UM} + \overrightarrow{MD_i} = \overrightarrow{UA_i} + \overrightarrow{A_iB_i} + \overrightarrow{B_iD_i}, \quad i = 1; 2; 3 \quad (19.)$$

Mivel jelen esetben csak a θ_i szögek értékei fogják befolyásolni a pozíciót, ezért, ha $\overrightarrow{B_iD_i}$ vektorokat fejezem ki a (19.) egyenletből azzal el tudom kerülni, hogy számolni kelljen az alsó és a felső kar rész közötti szögek értékeivel:

$$\overrightarrow{B_iD_i} = (\overrightarrow{UM} + \overrightarrow{MD_i}) - (\overrightarrow{UA_i} + \overrightarrow{A_iB_i}), \quad i = 1; 2; 3 \quad (20.)$$

A zárójelekkel való csoportosítás értelme, hogy szemléltetve legyen, hogy az összekapcsolt elemek az U_i és a D_i , illetve az U_i és a B_i pontok közötti vektorokat határozzák meg. Jelen problémánál nekem leginkább az $\overrightarrow{UB_i}$ és $\overrightarrow{MD_i}$ vektorok lesznek szükségesek a virtuális gömbközepppontok meghatározása érdekében. Mivel a koordináták nem egyeznek meg a három esetben ezért a három kar esetében külön-külön, más értékekkel kell megoldani az alapból azonos elven felépülő kinematikai feladatot.

Méréssel meghatároztam a mozgó platform középpontjának és a hozzákapcsolódó gömb csuklók távolságát, azaz $\overrightarrow{MD_1} = [-d; 0; 0]^T$. Ezt követően kiszámítom a virtuális $\overrightarrow{UB_i}$ gömb középpontok meghatározásához szükséges $\overrightarrow{UB_i}$ vektorokat.

$$\overrightarrow{UB_1} = \overrightarrow{UA_1} + \overrightarrow{A_1B_1} = \begin{bmatrix} -a \\ 0 \\ -5 \end{bmatrix} + \begin{bmatrix} -l \cos(-\theta_1 + 50) \\ 0 \\ -l \sin(-\theta_1 + 50) \end{bmatrix} = \begin{bmatrix} -a - lC_1 \\ 0 \\ -5 - lS_1 \end{bmatrix} \quad (21.)$$

ahol C_1 egyenlő $\cos(-\theta_1 + 50)$ -el S_1 pedig $\sin(-\theta_1 + 50)$ -el. A z koordináta mentén azért van -5-tel történő eltolás, mert a világkoordináta-rendszer origója 5 milliméterrel fentebb van, mint a delta rész csuklóit alkotó szabályos háromszög középpontja.

A második vektor esetén a számítás folyamata megegyezik az előzőekben elvégzettekkel, azzal a különbséggel, hogy az eddig használt vektorok $+120^\circ$ fokkal el vannak forgatva a világkoordináta-rendszer z tengelye körül. Ennek következtében a vektorok meg lesznek szorozva egy rotációs mátrixszal:

$$\overrightarrow{MD_2} = Rot(z, 120) \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 & -0.866 & 0 \\ 0.866 & -0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5d \\ -0.866d \\ 0 \end{bmatrix} \quad (22.)$$

$$\overrightarrow{UA_2} = Rot(z, 120) \begin{bmatrix} -a \\ 0 \\ -5 \end{bmatrix} = \begin{bmatrix} -0.5 & -0.866 & 0 \\ 0.866 & -0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -a \\ 0 \\ -5 \end{bmatrix} = \begin{bmatrix} 0.5a \\ -0.866a \\ -5 \end{bmatrix} \quad (23.)$$

$$\begin{aligned} \overrightarrow{A_2B_2} &= Rot(z, 120) \begin{bmatrix} -l\cos(-\theta_2 + 50) \\ 0 \\ -l\sin(-\theta_2 + 50) \end{bmatrix} = \begin{bmatrix} -0.5 & -0.866 & 0 \\ 0.866 & -0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -lC_2 \\ 0 \\ -lS_2 \end{bmatrix} \\ &= \begin{bmatrix} 0.5lC_2 \\ -0.866lC_2 \\ -lS_2 \end{bmatrix} \end{aligned} \quad (24.)$$

Ez alapján az $\overrightarrow{UB_2}$ vektor:

$$\overrightarrow{UB_2} = \overrightarrow{UA_2} + \overrightarrow{A_2B_2} = \begin{bmatrix} 0.5a \\ -0.866a \\ -5 \end{bmatrix} + \begin{bmatrix} 0.5lC_2 \\ -0.866lC_2 \\ -lS_2 \end{bmatrix} = \begin{bmatrix} 0.5(lC_2 + a) \\ -0.866(lC_2 + a) \\ -5 - lS_2 \end{bmatrix} \quad (25.)$$

Ez az eset nagyban hasonló a második kinematikai lánc esetén fellépővel, azzal a különbséggel, hogy jelen esetben az z tengely körüli forgatás értéke -120° lesz, a rotációs mátrix így pedig $Rot(z, -120)$. A harmadik karhoz tartozó vektor értékek:

$$\overrightarrow{MD_3} = Rot(z, -120) \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 & 0.866 & 0 \\ -0.866 & -0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5d \\ 0.866d \\ 0 \end{bmatrix} \quad (26.)$$

$$\overrightarrow{UA_3} = Rot(z, -120) \begin{bmatrix} -a \\ 0 \\ -5 \end{bmatrix} = \begin{bmatrix} -0.5 & 0.866 & 0 \\ -0.866 & -0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -a \\ 0 \\ -5 \end{bmatrix} = \begin{bmatrix} 0.5a \\ 0.866a \\ -5 \end{bmatrix} \quad (27.)$$

$$\begin{aligned} \overrightarrow{A_3B_3} &= Rot(z, -120) \begin{bmatrix} -l\cos(-\theta_3 + 50) \\ 0 \\ -l\sin(-\theta_3 + 50) \end{bmatrix} = \begin{bmatrix} -0.5 & 0.866 & 0 \\ -0.866 & -0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -lC_3 \\ 0 \\ l - S_3 \end{bmatrix} \\ &= \begin{bmatrix} 0.5lC_3 \\ 0.866lC_3 \\ -lS_3 \end{bmatrix} \end{aligned} \quad (28.)$$

Az $\overrightarrow{UB_3}$ vektor kiszámítva:

$$\overrightarrow{UB_3} = \overrightarrow{UA_3} + \overrightarrow{A_3B_3} = \begin{bmatrix} -0.5a \\ 0.866a \\ 0 \end{bmatrix} + \begin{bmatrix} -0.5lC_3 \\ 0.866lC_3 \\ lS_3 \end{bmatrix} = \begin{bmatrix} 0.5(lC_3 + a) \\ 0.866(lC_3 + a) \\ -5 - lS_3 \end{bmatrix} \quad (29.)$$

A virtuális gömb középpontok kiszámítása a paralelogramma felírásának segítségével általánosan:

$$\overrightarrow{UB_i'} = \overrightarrow{UB_i} - \overrightarrow{MD_i}, \quad i = 1; 2; 3 \quad (30.)$$

Ez alapján felírom a középpontokat a három kar esetén:

$$\overrightarrow{UB_1'} = \begin{bmatrix} -a - lC_1 \\ 0 \\ -5 - lS_1 \end{bmatrix} - \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -a + d - lC_1 \\ 0 \\ -5 - lS_1 \end{bmatrix} \quad (31.)$$

$$\overrightarrow{UB_2'} = \begin{bmatrix} 0.5(lC_2 + a) \\ -0.866(lC_2 + a) \\ -5 - lS_2 \end{bmatrix} - \begin{bmatrix} 0.5d \\ -0.866d \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5(a - d + lC_2) \\ -0.866(a - d + lC_2) \\ -5 - lS_2 \end{bmatrix} \quad (32.)$$

$$\overrightarrow{UB_3'} = \begin{bmatrix} 0.5(lC_3 + a) \\ 0.866(lC_3 + a) \\ -5 - lS_3 \end{bmatrix} - \begin{bmatrix} 0.5d \\ 0.866d \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5(a - d + lC_3) \\ 0.866(a - d + lC_3) \\ -5 - lS_3 \end{bmatrix} \quad (33.)$$

Legyenek a gömbközeppek koordinátái a következőképpen paraméterezve $\overrightarrow{UB_1'} = [xg_1 \ yg_1 \ zg_1]$, $\overrightarrow{UB_2'} = [xg_2 \ yg_2 \ zg_2]$ és $\overrightarrow{UB_3'} = [xg_3 \ yg_3 \ zg_3]$. Ezek segítségével a három gömbre felírható összefüggésből képzett egyenletrendszer [21]:

$$\begin{aligned} (x - xg_1)^2 + (y - yg_1)^2 + (z - zg_1)^2 &= r^2 \\ (x - xg_2)^2 + (y - yg_2)^2 + (z - zg_2)^2 &= r^2 \\ (x - xg_3)^2 + (y - yg_3)^2 + (z - zg_3)^2 &= r^2 \end{aligned} \quad (34.)$$

Ha felbontom az egyenletet a következő alakhoz jutok:

$$\begin{aligned} x^2 + y^2 + z^2 - 2(xxg_1 + yyg_1 + zzg_1) + xg_1^2 + yg_1^2 + zg_1^2 &= r^2 \\ x^2 + y^2 + z^2 - 2(xxg_2 + yyg_2 + zzg_2) + xg_2^2 + yg_2^2 + zg_2^2 &= r^2 \\ x^2 + y^2 + z^2 - 2(xxg_3 + yyg_3 + zzg_3) + xg_3^2 + yg_3^2 + zg_3^2 &= r^2 \end{aligned} \quad (35.)$$

Kivonom egymásból az első és a harmadik majd a második és a harmadik egyenletet:

$$\begin{aligned} 2((xg_3 - xg_1)x + (yg_3 - yg_1)y + (zg_3 - zg_1)z) \\ = -xg_1^2 - yg_1^2 - zg_1^2 + xg_3^2 + yg_3^2 + zg_3^2 \\ 2((xg_3 - xg_2)x + (yg_3 - yg_2)y + (zg_3 - zg_2)z) \\ = -xg_2^2 - yg_2^2 - zg_2^2 + xg_3^2 + yg_3^2 + zg_3^2 \end{aligned} \quad (36.)$$

Ahhoz, hogy megoldjam ezt a három ismeretlenes egyenletrendszert elsőként kifejezem x -t és z -t y függvényében. Ehhez bevezetek új paramétereket a követhetőbb számolás érdekében:

$$a_{11}x + a_{12}y + a_{13}z = b_1 \quad (37.)$$

$$a_{21}x + a_{22}y + a_{23}z = b_2 \quad (38.)$$

A paraméterek a következő összefüggéseket jelölik:

$$a_{11} = 2(xg_3 - xg_1) \quad a_{21} = 2(xg_3 - xg_2)$$

$$a_{12} = 2(yg_3 - yg_1) \quad a_{22} = 2(yg_3 - yg_2)$$

$$a_{13} = 2(zg_3 - zg_1) \quad a_{23} = 2(zg_3 - zg_2)$$

$$b_1 = -xg_1^2 - yg_1^2 - zg_1^2 + xg_3^2 + yg_3^2 + zg_3^2$$

$$b_2 = -xg_2^2 - yg_2^2 - zg_2^2 + xg_3^2 + yg_3^2 + zg_3^2$$

Ha a (37.) és (38.) egyenleteket átrendezve kifejezem z -t:

$$z = \frac{b_1}{a_{13}} - \frac{a_{11}}{a_{13}}x - \frac{a_{12}}{a_{13}}y \quad (39.)$$

$$z = \frac{b_2}{a_{23}} - \frac{a_{21}}{a_{23}}x - \frac{a_{22}}{a_{23}}y \quad (40.)$$

A (39.) és (40.) egyenletekből z -t eliminálva kifejezem $x = f(y)$ függvényt:

$$x = f(y) = a_4y + a_5 \quad (41.)$$

Az összefüggésben szereplő rövidítések:

$$a_4 = -\frac{a_2}{a_1}$$

$$a_5 = -\frac{a_3}{a_1}$$

$$a_1 = \frac{a_{11}}{a_{13}} - \frac{a_{21}}{a_{23}}$$

$$a_2 = \frac{a_{12}}{a_{13}} - \frac{a_{22}}{a_{23}}$$

$$a_3 = \frac{b_2}{a_{23}} - \frac{b_1}{a_{13}}$$

A (39.) és (40.) egyenletekből x -t eliminálva kifejezem $z = f(y)$ függvényt:

$$z = f(y) = a_6y + a_7 \quad (42.)$$

Ebben az összefüggésben szereplő rövidítések pedig:

$$a_6 = \frac{-a_{21}a_4 - a_{22}}{a_{23}}$$

$$a_7 = \frac{b_2 - a_{21}a_5}{a_{23}}$$

Annak a segítségével, hogy ki tudtam x -t és z -t fejezni az y segítségével, ha visszahelyettesítem a (41.) és (42.) egyenletet a (34.) egyenletrendszer első egyenletébe, akkor felírható az összefüggés egy y -ra másodfokú egyenlet segítségével:

$$Ay^2 + By + C = 0 \quad (43.)$$

Ahol a paraméterek a következő összefüggéseket jelölik:

$$A = a_4^2 + a_6^2 + 1$$

$$B = 2a_4(a_5 - xg_1) - 2yg_1 + 2a_6(a_7 - zg_1)$$

$$C = a_5(a_5 - 2xg_1) + a_7(a_7 - 2zg_1) + xg_1^2 + yg_1^2 + zg_1^2 - h^2$$

A másodfokú egyenletnek két megoldása lesz y -ra:

$$y_{\pm} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (44.)$$

Emiatt, ha visszahelyettesítem a (41.) és (42.) egyenletbe a (44.) egyenlet megoldását, tehát y_+ és y_- értékeket, akkor a függvényekre is két megoldást fogok kapni az y függvényében:

$$x_{\pm} = a_4y_{\pm} + a_5 \quad (45.)$$

$$z_{\pm} = a_6y_{\pm} + a_7 \quad (46.)$$

Tehát ez alapján megállapítható, hogy általánosan az egyenlet megoldása végén két végeredményhez fogok jutni. Az egyiket szokás a negatívnak, a másikat pedig a pozitív megoldásnak nevezni. Az értékeiket nem lehet összecserélni, nem lehet őket kombinálni.

$$\{x_+ \quad y_+ \quad z_+\}^T \quad \{x_- \quad y_- \quad z_-\}^T \quad (47.)$$

A kettő megoldás között általában a robot kialakítása segít, hogy eldöntsem melyik található meg a robot munkaterületén belül. Az lesz a megfelelő a kettő közül, amelyiket a robotkialakításának köszönhetően képes elérni. Ebben az esetben mivel a z világkoordináta „felfelé” fog mutatni, így számomra a kettő esetből az lesz a megfelelő, ahol a z koordináták értékei negatívak lesznek. Ennek segítségével könnyen választhatok

a két opció között. Azonban a direkt kinematika számítása során előfordulhatnak problémák, mint például a szingularitási vagy képzeletbeli pontok számítása.

Képzeletbeli megoldások:

A három gömb egymáshoz képesti elhelyezkedése lehetséges, hogy olyan megoldást ad, ami képzeletbeli, azaz nincs rá reális esély, hogy előforduljon. Matematikailag könnyen észrevehető, hogy az y -ra felírható másodfokú egyenlet megoldóképletéhez tartozó diszkrimináns értéke befolyásolja a képzeletbeli megoldásokat. Tehát ha $b^2 - 4ac$ kisebb, mint nulla abban az esetben mindkét megoldás képzeletbeli lesz, mert nem mind a három gömb fogja egymást metszeni.

Egy speciális helyzet lesz akkor, ha a $b^2 - 4ac$ összefüggés pontosan nullával lesz egyenlő. Ekkor a két megoldás helyett csak egy fog létezni, mivel a három gömbből kettő fog tangenciálisan egy pontban fog érintkezni és a harmadik gömb pedig valamiképpen átfog haladni ezen a ponton.

Direkt kinematika delta rész - szingularitási pontok:

Szingularitás akkor fog előállni a direkt kinematika delta részében, ha a gömböknek megegyezik a z koordinátájuk, tehát ha nullával való osztást kell elvégezni. Annak érdekében, hogy ettől függetlenül ki tudjam számítani hol fognak ezek az állapotok jelentkezni, szükséges, hogy az adott koordináták meg legyenek határozva. Ennek érdekében megfogom az összes szingularitási esetet vizsgálni, ahol az említett probléma fent állhat.

Első eset ($zg_1 = zg_3$)

Az első típusú szingularitás a következő esetekben fog jelentkezni:

$$a_{13} = 2(zg_3 - zg_1) = 0 \quad (48.)$$

Ez az eset akkor fog előállni, ha az első és a harmadik gömb z koordinátái meg fognak egyezni, tehát $zg_1 = zg_3$. A (37.) egyenlet a következőképpen fog alakulni az (38.) pedig változatlan lesz:

$$a_{11}x + a_{12}y = sz_1 \quad (49.)$$

Ahol a következő rövidítéseket használom:

$$sz_1 = -xg_1^2 - yg_1^2 + xg_3^2 + yg_3^2$$

Ez alapján kifejezem az (49.) egyenletből x -t, a (38.)-ból pedig z -t :

$$x = -\frac{a_{12}}{a_{11}}y + \frac{sz_1}{a_{11}} = a_4y + a_5 \quad (50.)$$

$$z = \frac{a_{12}a_{21} - a_{22}a_{11}}{a_{11}a_{23}} + \frac{b_2a_{11} - sz_1a_{21}}{a_{11}a_{23}}y = a_6y + a_7 \quad (51.)$$

Visszahelyettesítve a (34.) összefüggésbe a másodfokú egyenlet együtthatói meg fognak egyezni az általános esetben használtakkal. A kód egyszerűbbé tételének érdekében az y együtthatói megegyeznek az általános esetekben alkalmazottakkal. Ezek egyszerre való teljesülését, megakadályozzák a kódban található feltételek. Ebben ez esetben is két megoldást fogok kapni, egy negatívot és egy pozitívot ez y függvényében és az lesz a megfelelő, amelyik bele esik a mozgástartományába a robotnak.

Második eset ($zg_2 = zg_3$)

Akkor állhat fent, ha a következő egyenlőség teljesül:

$$a_{23} = 2(zg_3 - zg_2) = 0 \quad (52.)$$

Ebben az esetben akkor jelentkezik a probléma, amikor a második és a harmadik gömb z koordinátái fognak egybe esni, azaz $zg_2 = zg_3$. Fontos észrevétel azonban, hogy a koordináták kialakításának köszönhetően ebben az esetben $xg_2 = xg_3$ is teljesül, tehát:

$$a_{21} = 2(xg_3 - xg_2) = 0 \quad (53.)$$

Ebben az esetben a (37.) egyenlet változatlan marad a (38.) pedig a következőképpen fog módosulni:

$$a_{22}y = sz_2 \quad (54.)$$

Az egyenletben sz_2 a következőt rövidíti:

$$sz_2 = -xg_2^2 - yg_2^2 + xg_3^2 + yg_3^2$$

Mivel az (54.) egyenletből az y egyértelműen kifejezhető, ezért az y értékét behelyettesítve a (37.) egyenletbe, kifejezem z -t:

$$z = \frac{-a_{11}}{a_{13}}x + \frac{b_1 a_{22} - s z_2 a_{12}}{a_{22} a_{13}} = a_6 x + a_7 \quad (55.)$$

Visszahelyettesítve a (34.) egyenletbe az x-re kapott másodfokú egyenletre a következő együtthatókat kapom:

$$\begin{aligned} A &= a_6^2 + 1 \\ B &= -2(xg_1 - a_7 \cdot a_6 + a_6 \cdot zg_1) \\ C &= y^2 - 2 \cdot y \cdot yg_1 + a_7(a_7 - 2zg_1) + xg_1^2 + yg_1^2 + zg_1^2 - h^2 \end{aligned}$$

Ebben az esetben is számolni kell azzal, hogy kettő különböző eredmény fog születni megoldásként és csak az egyik lesz megfelelő, ami eleget tesz az előzőleg megadott feltételnek.

Harmadik eset ($zg_1 = zg_2 = zg_3$)

Ebben az esetben a mindhárom gömb z koordinátája meg fog egyezni, azaz $zg_1 = zg_2 = zg_3$ és emiatt meg fog egyezni $xg_2 = xg_3$ is. Ebben az esetben a (37.) és (38.) egyenletekből mindkét oldalt ki fognak esni a z koordinátát tartalmazó tagok és a következő egyenletekhez jutok az átírás után:

$$a_{11}x + a_{12}y = sz_1 \quad (56.)$$

$$a_{22}y = sz_2 \quad (57.)$$

Az (57.) egyenletből egyértelműen kifejezhető y és ezt visszahelyettesítve az (56.)-ba x értéke is számítható lesz. Visszahelyettesítve az értékeket a (34.) egyenletrendszerbe egy z-re másodfokú egyenletet kapok, aminek egy megoldása lesz, hiszen z nem vehet fel negatív értéket. A másodfokú egyenlet együtthatói:

$$\begin{aligned} A &= 1 \\ B &= -2zg_1 \\ C &= (x - xg_1)^2 + (y - yg_1)^2 + zg_1^2 - r^2 \end{aligned}$$

Megoldásképpen kettő darab eredmény születik z-re. Ezekből az alapján választom ki a helyeset, hogy melyik az amelyik nem negatív, mert a másik abban az esetben nem lenne a robot munkaterületének része.

2.1.2. HIBRID RÉSZ DIREKT IRÁNYÚ KINEMATIKÁJA

A hibrid rész direkt kinematikai feladatrészenek a megoldási célja a szerszám végpontjának és a mozgó platform egymáshoz való pozíciójának és orientációjának meghatározása az ismert θ_i szögek alapján. Mivel az előzőekben tárgyalt direkt irányú kinematika a delta robotrész esetén a koordináta rendszer orientációját nem változtatta meg csak, a pozícióját ezért az itt kapott értékek fogják teljes mértékben meghatározni az Euler szögek értékét. Jelen esetben a könnyebb számítás érdekében, amikor a referenciakoordinátarendszerre reflektálok az alatt egy olyan koordináta rendszer érték, aminek origója egybe esik a mozgó platform M pontjával.

A megoldás menete teljes mértékben megegyezik az 1.3. fejezetben részletezett lépésekkel. Cél a Denavit-Hartenberg transzformációs mátrix meghatározása. Annak érdekében, hogy a végleges helyzetét megkapjam az effektornak, ahhoz össze kell adnom a delta rész esetén kapott x , y és z koordináták értékeit a transzformációs mátrix utolsó sorának megfelelő elemeivel.

A rotációs mátrixból az Euler szögek visszaszámításának módja független az adott robotmanipulátor típusától. Az Euler szögek, ahogy már előzőleg is említve voltak rendre ψ , θ és φ . Az (58.) egyenletben ismertetett Rotációs mátrix megegyezik az előző három szög által végzett koordináta transzformációkból képzett mátrixok szorzatával. A szorzásokat elvégezve, az adott szögek szinuszának és koszinuszának szorzata kiadja az egységvektorok merőleges vetületeinek értékét. Ezek alapján a szögek az egyenletrendszerekből alábbiak szerint kifejezhetők:

$$\psi = \arctg \frac{e_{1y}}{e_{1x}} \quad (58.)$$

$$\theta = \arctg \left[\frac{-e_{1z}}{e_{1x} \cos \psi + e_{1y} \sin \psi} \right] \quad (59.)$$

$$\varphi = \arctg \frac{e_{2z}}{e_{3z}} \quad (60.)$$

Direkt kinematika hibrid rész - szingularitási pontok:

A szögek számítása közben megjelenhetnek szingularitási pontok a nullával osztás miatt. Akkor fog ez jelentkezni, amikor $\theta = \pm \frac{\pi}{2}$, tehát amikor az $e_{1x} = e_{1y} = e_{2z} = e_{3z} = 0$,

tehát a ψ és a φ számítása során az értékek megegyeznek és egyenlőek nullával. Ebben az esetben nem ad megoldást a trigonometrikus függvény tehát a ψ forgatási szög értéke tetszőlegesen megválasztható. A ψ függvényében kétféleképpen írható fel a φ :

$$\varphi = \operatorname{arctg} \frac{-e_{2x}}{e_{2y}} - \psi \quad \text{ha } \theta = -\frac{\pi}{2} \quad (61.)$$

$$\varphi = \operatorname{arctg} \frac{e_{2x}}{e_{2y}} + \psi \quad \text{ha } \theta = \frac{\pi}{2} \quad (62.)$$

2.2. INVERZ IRÁNYÚ KINEMATIKA

Az inverz kinematikai feladat célja a csuklókoordináták kiszámítása a világkoordináták ismeretében. Tehát ismerve a robotmanipulátor effektorjának helyét és pozícióját az alapján meghatározom az adott csuklók milyen szögeket zárnak be:

$$q = f^{-1}(s) \quad (63.)$$

A delta robotok inverz kinematikája egy matematikai vektor egyenlet megoldása. A hibrid rész esetén egy háromismeretlenes egyenletrendszer kiszámítása a feladat.

2.2.1. HIBRID RÉSZ INVERZ IRÁNYÚ KINEMATIKÁJA

Az inverz kinematikai feladat megoldása a hibrid rész számításainak elvégzésével kezdődik. Ahogy már említettem a robot manipulátor hibrid része sorba kapcsolt kinematikai láncsal rendelkezik. Általánosan az inverz kinematikai feladat megoldására két módszer áll rendelkezésre: a geometria és az algebrai megoldás [22]. Ebben a kivételes esetben azonban a csuklók szögei bármelyik megoldástípustól eltérően könnyen számíthatók az Euler szögek értékeiből. Mivel három sorba kapcsolt csukló koordinátáit szeretném visszkapni három már ismert szög értékéből a feladat megoldása egy háromismeretlenes három egyenletből álló egyenletrendszerre vezethető vissza. Azért van erre lehetőségem, mert a parallel kialakítású delta rész esetén az Euler szögek változatlanok maradnak annak köszönhetően, hogy a mozgórészhez tartozó referenciakoordinátarendszer minden pillanatban párhuzamos lesz a világkoordinátarendszerrel. Emiatt le vonható a következtetés, hogy az Euler szögek értékeit minden esetben csak a hibrid rész szögelfordulásai fogják befolyásolni.

A csuklókoordináták kifejezéséhez elsőként szükséges a Homogén transzformációs mátrixot meghatározni paraméteresen. Ennek első három sora alkotja a Rotációs mátrixot, amely, ahogy már említve volt definiálja az Euler szögek értékeit. Emiatt ismert a Rotációs mátrixnak egy olyan alakja, amikben csak az Euler szögek szögfüggvényeinek szorzatai szerepelnek [10] [23]:

$${}^0R_n = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\cos\varphi - \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi - \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \quad (64.)$$

A rotációs mátrixba behelyettesítem az Euler szögek értékeit, amik már előre ismertek, majd ezt követően pedig kiválasztok ebből és a paraméteres Homogén transzformációs mátrixból három-három páronként megegyező indexű elemet és az így kapott háromismeretlenes egyenlet megoldásai lesznek a csuklókoordináták értékei.

Inverz kinematika hibrid rész - szingularitási pontok:

Annak érdekében, hogy az inverz feladatnak legyen megoldás szükségszer teljesülnie kell a következő összefüggésnek:

$$\dot{s} = \left(\frac{\partial f}{\partial q} \right) \dot{q} \quad (65.)$$

Azaz az inverz függvény felírásánál szükség van a koordináták derivált vektorának használatára, ahol a Jacobi-mátrix [24] [25]:

$$J(q) = \frac{\partial f}{\partial q} \quad (66.)$$

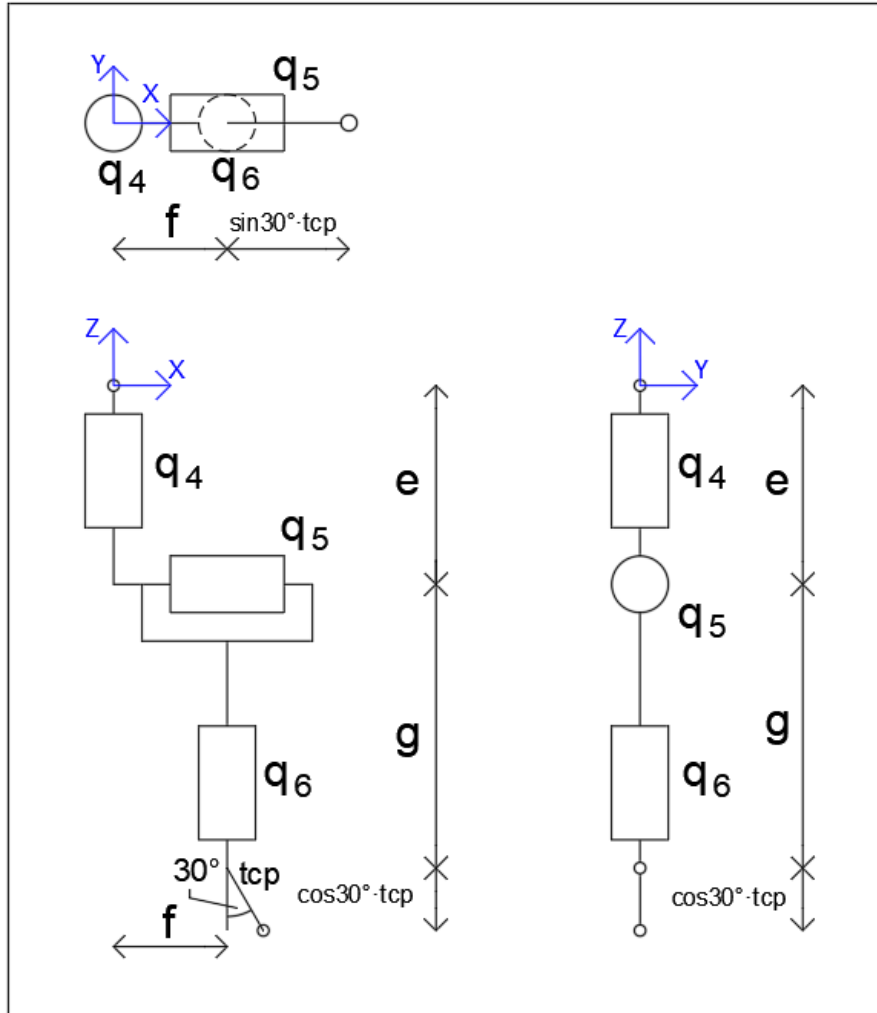
Behelyettesítve:

$$\dot{s} = J(q)\dot{q} \quad (67.)$$

$$\dot{q} = J^{-1}(q)\dot{s} \quad (68.)$$

Látható, hogy a Jacobi-mátrix inverze a csukló és világkoordináták deriváltjai között teremt kapcsolatot [26]. Ahhoz, hogy ez a számítás létrejöjjön szükséges, hogy a Jacobi-mátrixnak létezzen inverz mátrixa. Magához a mátrix meghatározásához szükséges a világkoordináták és csuklókoordináták sebesség vektorainak felírása.

Ehhez azonban szükség van a geometria megoldás elkészítéséhez. A geometria megoldás alapvető lényege, hogy a térbeli geometriájú robotkart lebontom több síkgeometriájú feladat részre, majd ezek alapján kifejezem a szögek segítségével a manipulátor világkoordinátáit [27].



9. ábra: Geometriai megoldás [Saját forrás]

A koordináták felírása:

$$\begin{aligned} x &= \cos(\theta_4) \cdot f + \cos(\theta_4 + \theta_6) \cdot tcp \cdot \sin 30^\circ \\ y &= \sin(\theta_4) \cdot f + \sin(\theta_4 + \theta_6) \cdot tcp \cdot \sin 30^\circ \\ z &= -e - \cos(\theta_5) \cdot (g + tcp \cdot \sin 30^\circ) \end{aligned} \quad (69.)$$

A sebességvektorok koordinátái

$$\dot{x} = -\sin(\theta_4) \cdot \dot{\theta}_4 \cdot f - \sin(\theta_4 + \theta_6) \cdot (\dot{\theta}_4 + \dot{\theta}_6) \cdot tcp \cdot \sin 30^\circ \quad (70.)$$

$$\begin{aligned}\dot{y} &= \cos(\theta_4) \cdot \dot{\theta}_4 \cdot f + \cos(\theta_4 + \theta_6) \cdot (\dot{\theta}_4 + \dot{\theta}_6) \cdot tcp \cdot \sin 30^\circ \\ \dot{z} &= -\sin(\theta_5) \cdot \dot{\theta}_5 \cdot (g + tcp \cdot \cos 30^\circ)\end{aligned}$$

Átírva az egyenleteket mátrixos alakba a következő kifejezéshez jutok:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -\sin(\theta_4) \cdot f - \sin(\theta_4 + \theta_6) \cdot \frac{tcp}{2} & 0 & -\sin(\theta_4 + \theta_6) \cdot \frac{tcp}{2} \\ \cos(\theta_4) \cdot f + \cos(\theta_4 + \theta_6) \cdot \frac{tcp}{2} & 0 & \cos(\theta_4 + \theta_6) \cdot \frac{tcp}{2} \\ 0 & -\sin(\theta_5) \cdot \dot{\theta}_5 \cdot (g + tcp \cdot \frac{\sqrt{3}}{2}) & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix} \quad (71.)$$

A Jacobi-mátriya ezek alapján

$$J(q) = \begin{bmatrix} -\sin(\theta_4) \cdot f - \sin(\theta_4 + \theta_6) \cdot \frac{tcp}{2} & 0 & -\sin(\theta_4 + \theta_6) \cdot \frac{tcp}{2} \\ \cos(\theta_4) \cdot f + \cos(\theta_4 + \theta_6) \cdot \frac{tcp}{2} & 0 & \cos(\theta_4 + \theta_6) \cdot \frac{tcp}{2} \\ 0 & -\sin(\theta_5) \cdot \dot{\theta}_5 \cdot (g + tcp \cdot \frac{\sqrt{3}}{2}) & 0 \end{bmatrix} \quad (72.)$$

A robot akkor fog szingularitási pontba kerülni, amikor a Jacobi mátrix determinánsa meg fog egyezni nullával [28]. Hiszen az inverz mátrix kiszámítása során szükség van az inverz mátrix előállításánál minden esetben le kell osztani az elemek értékét az eredeti mátrix determinánsának értékével. Így abban az esetben, ha ez nulla, nullával való osztás következne be, amit a gép nem tudna kiértékelni.

2.2.2. DELTA RÉSZ INVERZ IRÁNYÚ KINEMATIKÁJA

A robot az inverz kinematikai feladatát vektor módszerrel lehet levezetni. Mivel a rendszer zárt kinematikai láncot alkot, ez a módszer egy jól átlátható és követhető megoldást biztosít [5] [29]. Jelen esetben a robot három karral rendelkezik ezért eredményként a számítás végén három θ értéket kell visszkapnom. A $\overrightarrow{B_l D_l}$ vektorok koordinátáit a (19.) és (20.) egyenletek alapján számítom:

$$\overrightarrow{B_l D_l} = (D_{ix} - B_{ix})\hat{i} + (D_{iy} - B_{iy})\hat{j} + (D_{iz} - B_{iz})\hat{k} \quad (73.)$$

Első kinematikai lánc:

A hibrid rész inverz kinematikájának köszönhetően a rendszer világkoordináta-rendszerében ismert a mozgó platform középpontjának pozíciója $\overrightarrow{UM} = [P_x; P_y; P_z]^T$. Ahhoz, hogy behelyettesítsem a megfelelő értékeket a (20.) összefüggésbe szükséges a következő vektor kiszámítására:

$$\overrightarrow{UD_1} = \overrightarrow{UM} + \overrightarrow{MD_1} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} P_x - d \\ P_y \\ P_z \end{bmatrix} \quad (74.)$$

Ha visszahelyettesíték a (20.) egyenletbe, akkor a következő összefüggéshez jutok:

$$\overrightarrow{B_1D_1} = (Q_{11} + lC_1)\hat{i} + (P_y)\hat{j} + (Q_{12} + lS_1)\hat{k} \quad (75.)$$

ahol az egyszerűsítés érdekében $Q_{11} = P_x - d + a$ és $Q_{12} = P_z + 5 + lS_1$. $\overrightarrow{B_1D_1}$ vektor hossz négyzetének számítása:

$$\begin{aligned} |\overrightarrow{B_1D_1}|^2 &= h^2 = (Q_{11} + lC_1)^2 + (P_y)^2 + (Q_{12} + lS_1)^2 \\ &= Q_{11}^2 + P_y^2 + Q_{12}^2 + l^2 + 2Q_{11}lC_1 + 2Q_{12}lS_1 \end{aligned} \quad (76.)$$

A (76.) egyenletből kifejezem a szinuszt és koszinuszt nem tartalmazó elemeket. az egyszerűsítés érdekében, majd pedig megkapom a (77.) egyenletet belőle:

$$\begin{aligned} G_1 &= \frac{h^2 - Q_{11}^2 - P_y^2 - Q_{12}^2 - l^2}{2l} \\ Q_{12}\sin(\theta_1) + Q_{11}\cos(\theta_1) &= G_1 \end{aligned} \quad (77.)$$

A (77.) egyenletet megoldásához bevezettem egy D változót, amit $D = \sqrt{A^2 + B^2}$ összefüggéssel definiáltam (ami jelen esetben $D = \sqrt{Q_{12}^2 + Q_{11}^2}$ lenne), majd mindkét oldalt elosztottam D -vel:

$$\frac{A}{D}\sin(\theta_i) + \frac{B}{D}\cos(\theta_i) = \frac{C}{D} \quad (78.)$$

Mivel az A , B és D oldalak egy derékszögű háromszög oldalai, ezért:

$$\cos(\alpha) = \frac{A}{D}, \quad \sin(\alpha) = \frac{B}{D} \quad (79.)$$

A (78.) és a (79.) egyenlet alapján:

$$\begin{aligned}
\sin(\theta_i)\cos(\alpha) + \cos(\theta_i)\sin(\alpha) &= \frac{C}{D} \\
\sin(\theta_i + \alpha) &= \frac{C}{D} \\
\theta_i + \alpha &= \sin^{-1}\left(\frac{C}{D}\right)
\end{aligned} \tag{80.}$$

Ebből a megoldásból két értéket kapok vissza a szögfüggvények miatt. Az alapján fogom tudni kiválasztani a számomra megfelelő szöget, hogy megvizsgálom melyik fog bele esni a csukló maximális és minimális elfordulási értékébe.

Második kinematikai lánc:

Az $\overrightarrow{UM}$ vektor koordinátái változatlanok lesznek, mivel ugyanabból a mozgóplatform pozícióból szeretném visszaszámolni a motorok elfordulásának értékeit. Ez alapján az $\overrightarrow{UD_2}$:

$$\overrightarrow{UD_2} = \overrightarrow{UM} + \overrightarrow{MD_1} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} 0.5d \\ -0.866d \\ 0 \end{bmatrix} = \begin{bmatrix} P_x + d \\ P_y - 0.866d \\ P_z \end{bmatrix} \tag{81.}$$

A (20.) egyenletben definiált módon összeadom a megfelelő vektorokat, majd pedig azokat egymásból kivonva megkapom a $\overrightarrow{B_2D_2}$ vektort:

$$\overrightarrow{B_2D_2} = (Q_{21} - 0.5lC_2)\hat{i} + (Q_{22} + 0.866lC_2)\hat{j} + (Q_{23} + lS_2)\hat{k} \tag{82.}$$

ahol $Q_{21} = P_x + 0.5d - 0.5a$, $Q_{22} = P_y - 0.866d + 0.866a$ és $Q_{23} = P_z + 5$.

Ezt követően kiszámítom a $\overrightarrow{B_2D_2}$ vektor abszolútértékének a négyzetét, annak érdekében, hogy az előzőekhez hasonlóan ne koordinátákkal, hanem hossz értékekkel számoljak:

$$\begin{aligned}
|\overrightarrow{B_2D_2}|^2 &= h^2 = (Q_{21} - 0.5lC_2)^2 + (Q_{22} + 0.866lC_2)^2 + (Q_{23} + lS_2)^2 \\
&= Q_{21}^2 + Q_{22}^2 + Q_{23}^2 + l^2 - Q_{21}lC_2 + 1.732Q_{22}lC_2 + 2Q_{23}lS_2
\end{aligned} \tag{83.}$$

A (83.) egyenletben lévő tagokat csoportosítva a szerint, hogy tartalmazznak-e szinuszos vagy koszinuszos összefüggéseket a következő együtthatókhoz jutok:

$$\begin{aligned}
G_{21} &= \frac{h^2 - Q_{21}^2 - Q_{22}^2 - Q_{23}^2 - l^2}{l} \\
G_{22} &= -Q_{21} + 1.732Q_{22}
\end{aligned} \tag{84.}$$

$$G_{23} = 2Q_{23}$$

$$G_{23}\sin(\theta_2) + G_{22}\cos(\theta_2) = G_{21}$$

Ebben az esetben is alkalmazható a (78.), (79.) és (80.) egyenletekben levezett megoldási módszer. $A(\theta_2)$ kiszámítása során itt és két eredményt fogok kapni.

Harmadik kinematikai lánc:

Az $\overrightarrow{UM}$ vektor koordinátái itt szintén változatlanok maradnak. Az első és második kinematikai lánc esetén alkalmazott (20.) egyenletben definiált megoldási módszert követve az elvégzett összeadások és kivonás után a (85.) egyenletet kapom meg:

$$\overrightarrow{B_3D_3} = (Q_{31} - 0.5lC_2)\hat{i} + (Q_{32} - 0.866lC_3)\hat{j} + (Q_{33} + lS_3)\hat{k} \quad (85.)$$

ahol $Q_{31} = P_x + 0.5d - 0.5a$, $Q_{32} = P_y + 0.866d - 0.866a$ és $Q_{33} = P_z + 5$.

A $\overrightarrow{B_3D_3}$ vektor hosszúságának négyzete kifejezve a koordináták segítségével:

$$\begin{aligned} |\overrightarrow{B_3D_3}|^2 = h^2 &= (Q_{31} - 0.5lC_3)^2 + (Q_{32} - 0.866lC_3)^2 + (Q_{33} + lS_3)^2 \\ &= Q_{31}^2 + Q_{32}^2 + P_z^2 + l^2 + Q_{31}lC_3 - 1.732Q_{32}lC_3 - 2P_zlS_3 \end{aligned} \quad (86.)$$

A szinuszt és koszinuszt tartalmazó tagok csoportosításához szükséges együtthatók a következők, amelyek segítségével felírható a (24.) összefüggés, amiből kiszámítható az aktív csuklók szög elfordulása:

$$\begin{aligned} G_{31} &= \frac{h^2 - Q_{31}^2 - Q_{32}^2 - Q_{33}^2 - l^2}{l} \\ G_{32} &= -Q_{31} - 1.732Q_{32} \\ G_{33} &= 2Q_{33} \\ G_{33}\sin(\theta_3) + G_{32}\cos(\theta_3) &= G_{31} \end{aligned} \quad (87.)$$

A θ_3 értékre kiszámítására az első és a második kar esetén alkalmazott módszer segítségével kétféle megoldás lesz. Ezekből az lesz megfelelő, amely a motor által megengedett szögtartományok között mozog.

3. MÉRÉSI EREDMÉNYEK

A fejezetben a mérési eredményeim kerülnek bemutatásra. Elsőként a kinematikai feladat direkt és inverz irányú számítására mutatok be eseteket. Ezt követően pedig a szingularitás detektáló programkódomhoz szemléltetek példákat.

3.1 KINEMATIKAI SZÁMÍTÁSI ÉS MÉRÉSI EREDMÉNYEK ÖSSZEVETÉSE

A FANUC robot képes arra, hogy az iPendant Touch kézi vezérlőjének képernyőjén kijelezze az adott pillanatban felvett világ- és csuklókoordinátáinak értékét. A bemutatott mérések esetén minden esetben látható lesz a iPendant Touch képernyőjéről készített képernyőképek és a direkt, illetve inverz irányból közelített MATLAB számítások eredményei.

3.1.1 ELSŐ POZÍCIÓ

Az első pozíció esetén a következő értékek kerültek beállításra:

Busy	Stop	Hold	Fault	SRVO-003 Deadman switch released	10%
Run	I/O	Prod	Stop	MOTION LINE 2 T1 PAUSED WORLD	
POSITION			POSITION		
World Tool: 1			Joint Tool: 1		
Configuration: M, , 0, 0					
x: -12.260 y: -26.789 z: -394.081			J1: 32.234 J2: 14.773 J3: 2.874		
w: 178.979 p: -5.936 r: 91.920			J4: 4.089 J5: -24.083 J6: 87.510		

10. ábra: Első pozícióhoz tartozó mérési eredmények [Saját forrás]

Látható, hogy a képernyő fotó bal oldalán helyezkednek el a világ a jobb oldalán pedig a csuklókoordináták.

A MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

```
% BEMENETI ÉRTÉKEK
```

```
disp('Szögek')
```

```
theta1 =32.234
```

```
theta2 =14.773
```

```
theta3 =2.874
```

```
theta4 =4.089
```

```
theta5 =-24.083
```

```
theta6 =87.510
```

```
xyz =
```

```
-9.3511
```

```
-31.0070
```

```
-387.4450
```

```
foratasi_yaw =
```

```
91.9211
```

```
billentesi_pitch =
```

```
-5.9362
```

```
csavarasi_roll =
```

```
178.9787
```

11. ábra: Első pozíció direkt irányú közelítése MATLAB-ban [Saját forrás]

A MATLAB kódban kapott eredmények inverz irányú közelítés esetén:

```
BEMENETI ÉRTÉKEK
```

```
disp('Világkoordináták')
```

```
x=-12.260
```

```
y=-26.789
```

```
z=-394.081
```

```
yaw=91.92
```

```
pitch=-5.936
```

```
roll =178.979
```

```
theta5 =
```

```
-24.083209622250660701211085750069
```

```
theta6 =
```

```
87.510699630682073360862272111549
```

```
theta4 =
```

```
4.6624
```

```
theta1 =
```

```
28.365924293049716033260215549288
```

```
theta2 =
```

```
13.907618608152638989777308839485
```

```
theta3 =
```

```
-4.9554272274219066573254597766335
```

12. ábra: Első pozíció inverz irányú közelítése MATLAB-ban [Saját forrás]

Látható, hogy kód által számított értékek és a robotból lekért pozíciók nagy mértékben megegyeznek. A következő mérések során a helytakarékossági okokból a MATLAB kódban megkapott értékeket táblázatos formátumban fogom megjeleníteni.

3.1.2 MÁSODIK POZÍCIÓ

Az második pozíció esetén az alábbi értékek kerültek beállításra:

Busy	Stop	Hold	Fault	SRVO-003 Deadman switch released	10%
Run	I/O	Prod	Free	MOTION LINE 2 T1 PAUSED WORLD	
POSITION			POSITION		
World Tool: 1			Joint Tool: 1		
Configuration: M, , 0, 0					
x: 42.521 y: 33.963 z: -423.391			J1: -3.454 J2: -18.583 J3: 19.822		
w: 173.621 p: -22.347 r: -4.670			J4: -59.233 J5: -9.552 J6: 51.741		

13. ábra: Második pozícióhoz tartozó mérési eredmények [Saját forrás]

A MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

1. táblázat: Második pozíció direkt irányú közelítése MATLAB-ban [Saját forrás]

$J_1 = -3.454$	$J_2 = -18.583$	$J_3 = 19.822$	$J_4 = -59.233$	$J_5 = -9.552$	$J_6 = 51.741$
$x = 28.301$	$y = 30.672$	$z = -414.436$	$\psi = -4.67$	$\theta = -22.348$	$\varphi = 173.621$

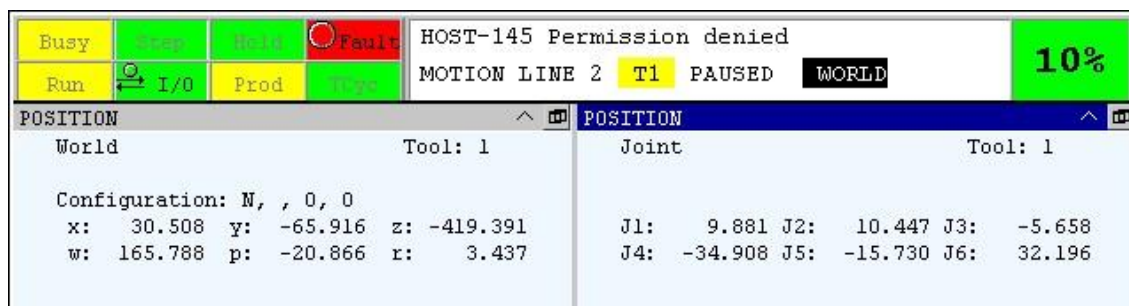
A MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

2. táblázat: Második pozíció inverz irányú közelítése MATLAB-ban [Saját forrás]

$x = 42.521$	$y = 33.963$	$z = -423.391$	$\psi = -4.67$	$\theta = -22.347$	$\varphi = 173.621$
$J_1 = -9.378$	$J_2 = -23.132$	$J_3 = 12.918$	$J_4 = -46.257$	$J_5 = -9.553$	$J_6 = 51.741$

3.1.3 HARMADIK POZÍCIÓ

A harmadik pozíciónál az alábbi értékek kerültek beállításra:



14. ábra: Harmadik pozícióhoz tartozó mérési eredmények [Saját forrás]

A MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

3. táblázat: Harmadik pozíció direkt irányú közelítése MATLAB-ban [Saját forrás]

$J_1 = 9.881$	$J_2 = 10.447$	$J_3 = -5.658$	$J_4 = -34.908$	$J_5 = -15.730$	$J_6 = 32.196$
$x = 16.723$	$y = -63.168$	$z = -411.905$	$\psi = 3.437$	$\theta = -20.866$	$\varphi = 165.788$

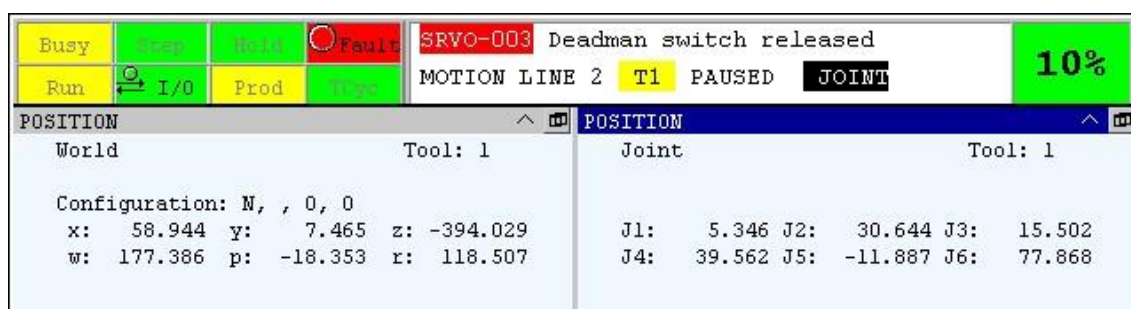
A MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

4. táblázat: Harmadik pozíció inverz irányú közelítése MATLAB-ban [Saját forrás]

$x = 30.508$	$y = -65.916$	$z = -419.391$	$\psi = 3.437$	$\theta = -20.866$	$\varphi = 165.788$
$J_1 = 2.068$	$J_2 = 10.534$	$J_3 = -18.717$	$J_4 = -21.002$	$J_5 = -15.729$	$J_6 = 32.196$

3.1.4 NEGYEDIK POZÍCIÓ

A negyedik pozíciónál az alábbi értékeket állítottam be:



15. ábra: Negyedik pozícióhoz tartozó mérési eredmények [Saját forrás]

MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

5. táblázat: Negyedik pozíció direkt irányú közelítése MATLAB-ban [Saját forrás]

$J_1 = 5.346$	$J_2 = 30.644$	$J_3 = 15.502$	$J_4 = 39.562$	$J_5 = -11.887$	$J_6 = 77.868$
$x = 71.514$	$y = -2.366$	$z = -389.075$	$\psi = 118.506$	$\theta = -18.353$	$\varphi = 177.386$

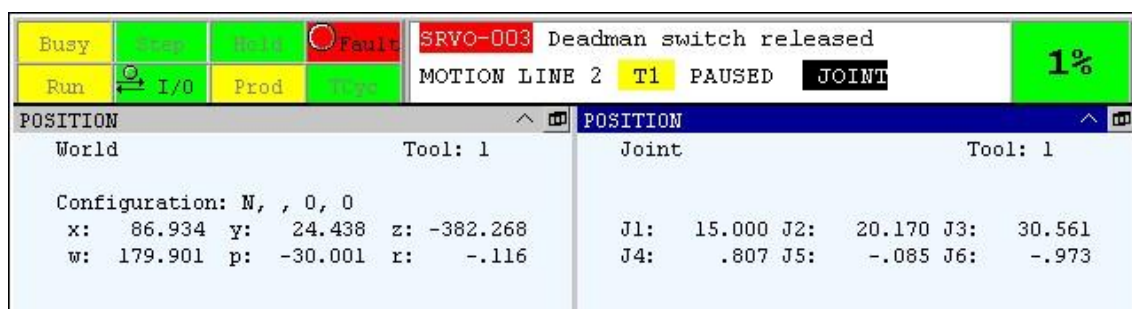
A MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

6. táblázat: Negyedik pozíció inverz irányú közelítése MATLAB-ban [Saját forrás]

$x = 58.944$	$y = 7.465$	$z = -394.029$	$\psi = 118.507$	$\theta = -18.353$	$\varphi = 177.386$
$J_1 = 3.693$	$J_2 = 27.342$	$J_3 = 12.460$	$J_4 = 45.994$	$J_5 = -11.887$	$J_6 = 77.869$

3.1.5 ÖTÖDIK POZÍCIÓ

Az ötödik pozíciónál az alábbi értékeket állítottam be az eszközön:



The screenshot shows a CNC control interface with a status bar at the top containing buttons for 'Busy', 'Stop', 'Hold', 'Fault', 'Run', 'I/O', 'Prod', and 'Stop'. The status bar also displays 'SRVO-003 Deadman switch released', 'MOTION LINE 2 T1 PAUSED', and 'JOINT'. Below the status bar, there are two panels for 'POSITION'. The left panel shows 'World' coordinates: Configuration: N, 0, 0; x: 86.934, y: 24.438, z: -382.268; w: 179.901, p: -30.001, r: -.116. The right panel shows 'Joint' coordinates: J1: 15.000, J2: 20.170, J3: 30.561, J4: .807, J5: -.085, J6: -.973.

16. ábra: Ötödik pozícióhoz tartozó mérési eredmények [Saját forrás]

MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

7. táblázat: Ötödik pozíció direkt irányú közelítése MATLAB-ban [Saját forrás]

$J_1 = 15.000$	$J_2 = 20.170$	$J_3 = 30.561$	$J_4 = 0.807$	$J_5 = -0.085$	$J_6 = -0.973$
$x = 69.354$	$y = 19.380$	$z = -381.701$	$\psi = -0.117$	$\theta = -30.001$	$\varphi = 179.902$

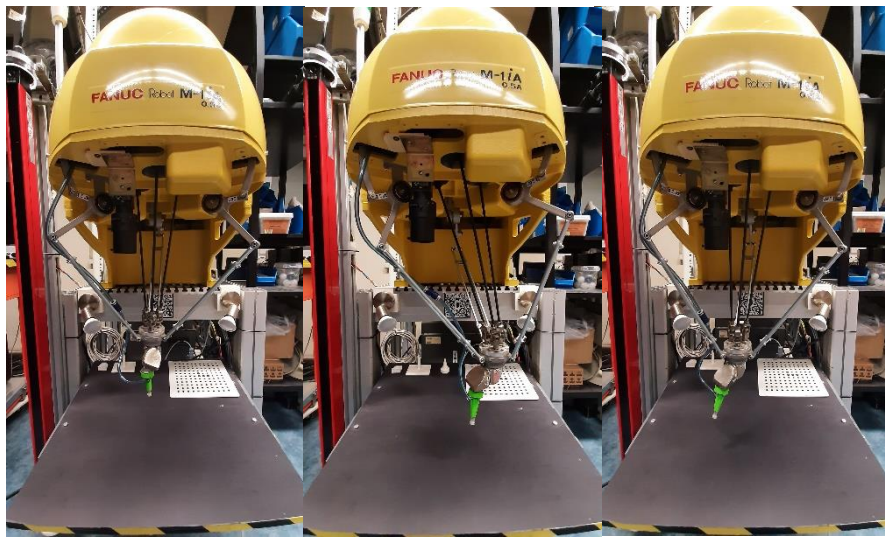
A MATLAB kódban kapott eredmények direkt irányú közelítés esetén:

8. táblázat: Ötödik pozíció inverz irányú közelítése MATLAB-ban [Saját forrás]

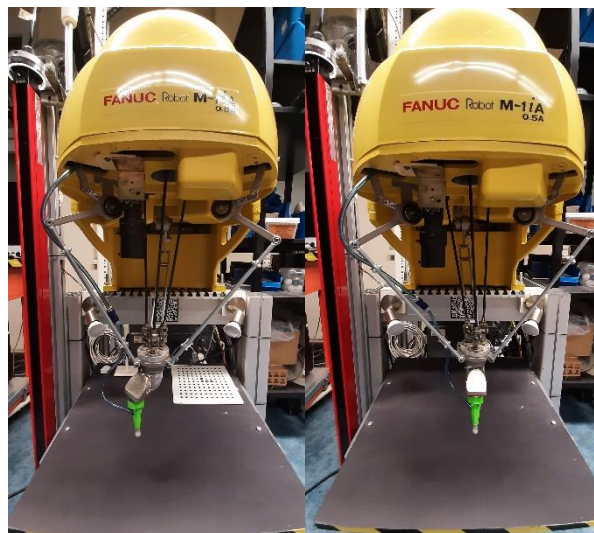
$x = 86.934$	$y = 24.438$	$z =$ -382.261	$\psi = -0.116$	$\theta =$ -30.001	$\varphi =$ 179.901
$J_1 = 8.259$	$J_2 = 22.550$	$J_3 = 31.190$	$J_4 = 11.135$	$J_5 = -0.086$	$J_6 = -0.693$

3.1.6 KÉPEK A ROBOTPOZÍCIÓKRÓL

Az alábbi (17.) és (18.) ábrán láthatóak egymás mellett a robot a különféle eddig ismertetett pozíciók mindegyikében. Az adott méréshez tartozó pozíciót a mérés sorszámaival jelölöm



17. ábra: Az 1., 2. és 3. pozíció szemléltetése [Saját forrás]



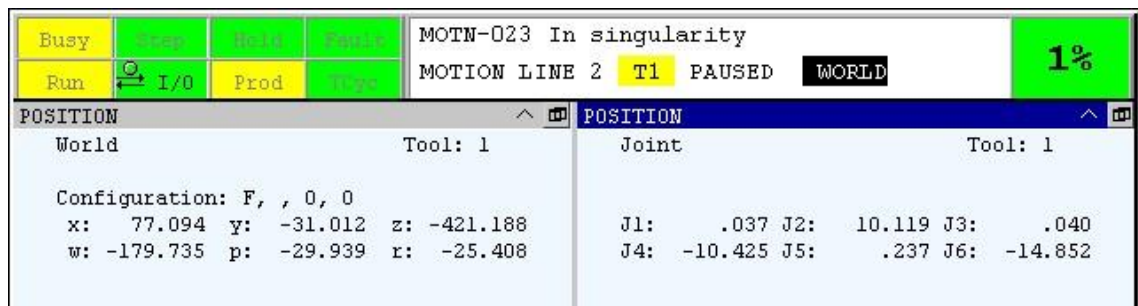
18. ábra: A 4. és 5. pozíció szemléltetése [Saját forrás]

3.2 SZINGULARITÁS DETEKCIO TESZTELÉSE

Az előző mérések szükségesek voltak ahhoz, hogy bemutassák, hogy a robot kinematikáját le tudja képezni az általam készített kód. Ezt követően ebben a fejezetben bemutatásra fog kerülni, hogy az általam készített program kód képes a szingularitás felismerésére. A következő részben beillesztett képekről az iPendant Touch képernyőjéről ki lehet olvasni, hogy a robot valóban szingularitási pozícióban van és látható lesz, hogy a programom is jelezni fogja ezt.

3.2.1 SZINGULARITÁS: DIREKT KINEMATIKA DELTA RÉSZ $ZG_1=ZG_3$

Az iPendant Touch képernyőjéről készült képernyőkép:



19. ábra: Szingularitás a robot rendszerében $z_{g1}=z_{g3}$ [Saját forrás]

Az általam készített programkód fotója:

% BEMENETI ÉRTÉKEK

disp('Szögek')

```
theta1 = 0
theta2 = 10.119
theta3 = 0

theta4 = -10.425
theta5 = 0.237
theta6 = -14.852
```

Singularitási pont: $z_{g1}=z_{g3}$

xyzdelta =

```
10.8512
-18.7948
-335.1928
```

xyz =

```
56.5125
-28.9739
-416.1907
```

forгатasi_yaw =

```
-25.4091
```

billentesi_pitch =

```
-29.9390
```

csavarasi_roll =

```
100.0000
```

20. ábra: Szingularitás detektálása: $z_{g1}=z_{g3}$ [Saját forrás]

3.2.2 SZINGULARITÁS: DIREKT KINEMATIKA DELTA RÉSZ $ZG_2=ZG_3$

A robot kezelőfelületéről leolvasható értékek:

Busy	Stop	Hold	Pause	MOTN-023 In singularity			1%
Run	I/O	Prod	TOYS	MOTION LINE 2 T1 PAUSED WORLD			
POSITION				POSITION			
World		Tool: 1		Joint		Tool: 1	
Configuration: F, , 0, 0							
x: 118.090		y: -12.266		z: -419.161			
w: -179.735		p: -29.939		r: -25.408			
				J1: -13.170		J2: 10.118 J3: 10.090	
				J4: -10.424		J5: .238 J6: -14.851	

21. ábra: Szingularitás a robot rendszerében $zg_2=zg_3$ [Saját forrás]

Az általam készített programkódról készült kép:

% BEMENETI ÉRTÉKEK

disp('Szögek')

theta1 = -13.170

theta2 = 10

theta3 = 10

theta4 = -10.425

theta5 = 0.237

theta6 = -14.852

Szingularitási pont: $zg_2=zg_3$

xyzdelta =

50.6902

0

-333.0175

xyz =

96.3515

-10.1791

-414.0154

forгатasi_yaw =

-25.4091

billentesi_pitch =

-29.9390

csavarasi_roll =

180.2644

22. ábra: Szingularitás detektálása: $zg_2=zg_3$ [Saját forrás]

3.2.3 SZINGULARITÁS: DIREKT KINEMATIKA DELTA RÉSZ $ZG_1=ZG_2=ZG_3$

A robot adott pozíciójában leolvasható értékek:

Busy	Stop	Hold	Pause	MOTN-023 In singularity			1%
Run	 I/O	Prod	Clear	MOTION LINE 2 T1 PAUSED WORLD			
POSITION				POSITION			
World		Tool: 1		Joint		Tool: 1	
Configuration: F, , 0, 0							
x: 66.253		y: -12.237		z: -428.526		J1: .037 J2: .051 J3: .041	
w: -179.736		p: -29.939		r: -25.409		J4: -10.425 J5: .237 J6: -14.852	

23. ábra: Szingularitás a robot rendszerében $z_{g1}=z_{g2}=z_{g3}$ [Saját forrás]

Az általam készített programkódról készült kép:

BEMENETI ÉRTÉKEK

disp('Szögek')

theta1 =0

theta2 =0

theta3 =0

theta4 =-10.425

theta5 =0.237

theta6 =-14.852

Szingularitási pont: $z_{g1}=z_{g2}=z_{g3}$

xyzdelta =

-0.0000

0

-341.1858

xyz =

45.6613

-10.1791

-422.1838

forgatasi_yaw =

-25.4091

billentesi_pitch =

-29.9390

csavarasi_roll =

180.2644

24. ábra: Szingularitás detektálása: $z_{g1}=z_{g2}=z_{g3}$ [Saját forrás]

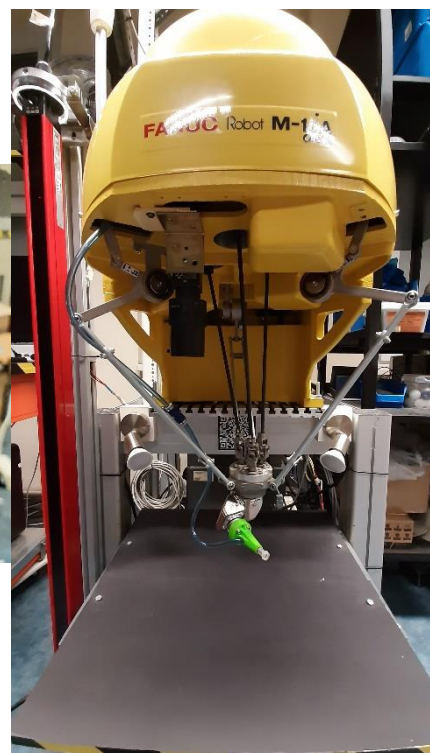
3.2.4 SZINGULARITÁS: DIREKT KINEMATIKA HIBRID RÉSZ

Ahogy az az 1.2.3.-as fejezet során már kifejtésre került ez a jellegű szingularitás abban az esetben áll fent, amennyiben $\theta = \pm \frac{\pi}{2}$. Azaz ez szögbe átkonvertálva azt fogja jelenteni, hogy $\theta = \pm 90^\circ$. A kísérlet többszöri elvégzése során észrevettem, hogy a FANUC robot nem képes arra, hogy ebbe az irányba álljon a geometriája, maximálisan az alábbi érték elérésére képes:

Busy	Stop	Hold	SRVO-003	Deadman switch released	1%
Run	I/O	Prod	MOTION LINE 2	T1 PAUSED	WORLD
POSITION			POSITION		
World			Joint		
Tool: 1			Tool: 1		
Configuration: F, , 0, 0					
x:	97.292	y: 38.592	z:	-310.107	
w:	125.182	p: -89.533	r:	89.402	
			J1:	22.555	J2: 22.765
			J3:	34.776	J4: -55.637
			J5:	59.732	J6: 90.442

25. ábra: θ szög értékének vizsgálata [Saját forrás]

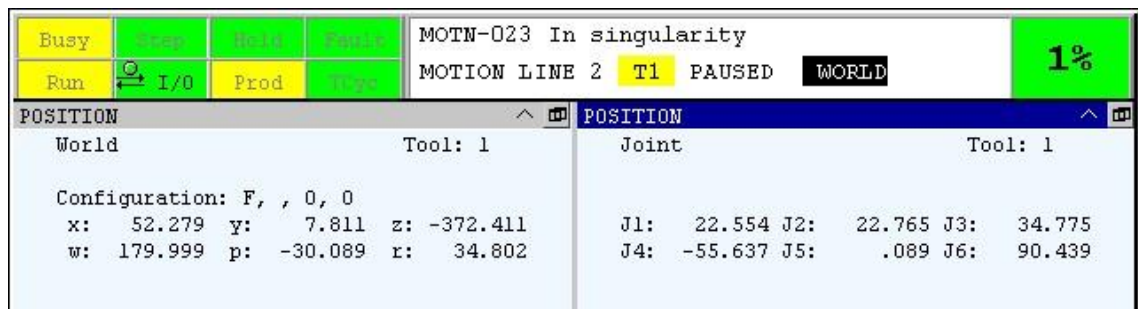
Látható, hogy maximálisan $\pm 89.533^\circ$ elérésére képes a robot manipulátor, emiatt ez a jellegű probléma gyárilag le van kezelve ez eszközben és ilyen jellegű szingularitás nem jelentkezhet nála. A következő képek szemléltetik a robot vizsgált pillanatbeli pozícióját:



26. ábra: θ szög maximális kitérésének tesztelése [Saját forrás]

3.2.5 SZINGULARITÁS: INVERZ KINEMATIKA HIBRID RÉSZ

Ebben az esetben a szingularitást az okozza, ha a Jacobi-mátrix determinánsa nulla. Az adott pozícióban ebből fakadólag szingularitást detektáltak:



27. ábra: Szingularitás a robot rendszerében a Jacobi mátrix miatt [Saját forrás]

Az általam készített programkód válasza az adott koordináták bevitelére:

% BEMENETI ÉRTÉKEK

disp('Világkoordináták')

x=52.279
y=27.811
z=-372.411
yaw=34.802
pitch=-30.089
roll=179.999

```
theta5 =
-0.089004209601454528359581820266665

theta6 =
89.442994417991042933684334603

theta4 =
-54.3818

Singularitási pont: A Jacobi mátrix detereminánsa nulla - Hibrid rész

thetal =
16.120305707489534182242143311674

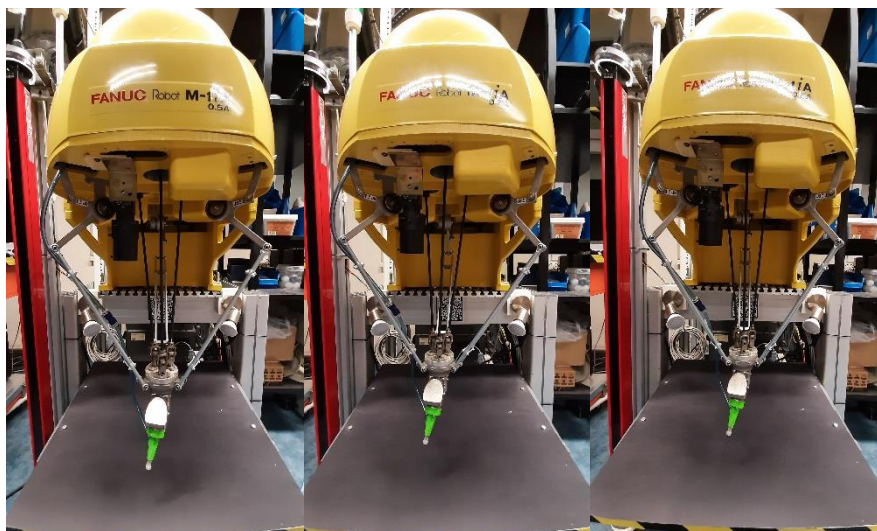
theta2 =
12.146842010706778067390626365393

theta3 =
43.26804811697073873991218339808
```

28. ábra: Szingularitás a robot rendszerében a Jacobi-mátrix miatt [Saját forrás]

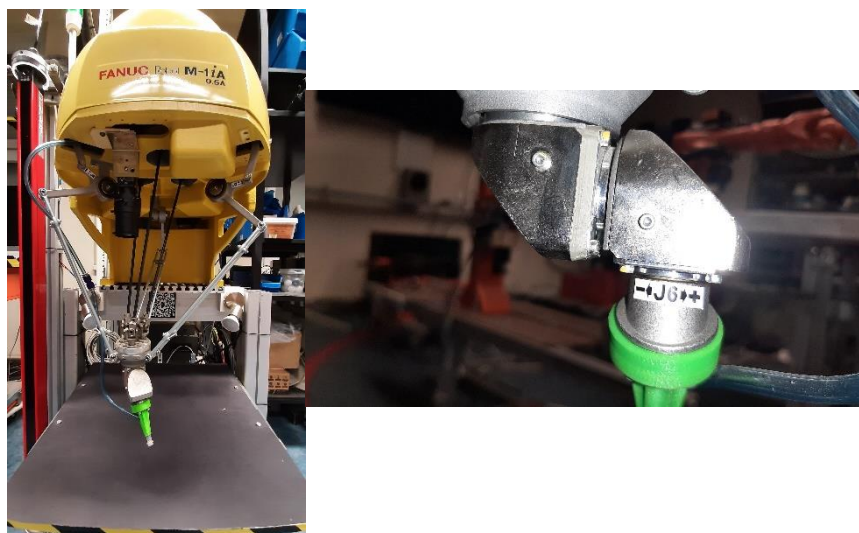
3.2.6 KÉPEK A SZINGULARITÁSI POZÍCIÓKRÓL

A direkt rész esetén vizsgált szingularitások pozíciók:



29. ábra: Direkt kinematika delta részén vizsgált pozíciók [Saját forrás]

Az inverz rész esetén vizsgált szingularitások pozíció:



30. ábra: Inverz kinematika hibrid részén vizsgált pozíciók [Saját forrás]

4. ÖSSZEFOGLALÓ

A feladatmegvalósítást sikeresnek tekintem. A dokumentumban bemutattam a kutatás célját és ismertettem annak felépítését. A mérési eredmények esetén látható, hogy a kód működése megfelel az elvártaknak és a valósághoz közeli eredményeket kaptam vissza. Megfigyelhető, hogy a kapott eredmények legtöbb esetben milliméteres vagy akár századmilliméteres eltérést mutatnak a robot saját rendszere által kalkulált és kijelzett adatoktól és ugyan ez figyelhető meg a kiszámított szög értékek esetében. Mindenképpen szeretném beazonosítani a jövőben, hogy honnan és milyen okból eredhetnek ezek a kis mértékű pontatlanságok és szeretném úgy átalakítani a kódomat., hogy kiküszöböljem ezeket. A tervem az, hogy még több mérés elvégzése után, meg szeretném becsülni, hogy a detektált eltérések állandó jelleggel jelen vannak, vagy felfedezhető-e bennük korreláció. Abban az esetben, ha a mért adatok korrelálnak, akkor felfedezhető a mért értékek és az általam számított adatok között egy lineáris kapcsolat. A korreláció számértéke jellemezni fogja ennek a kapcsolatnak az úgynevezett erősségét.

Véleményem szerint a kutatásom témája sok lehetőséget rejt magában és egy nagyon innovatív ötletnek tekinthető. A jövőben sok lehetőség áll a rendelkezésemre, hogy beléphessek a kifejlesztett kódnak köszönhetően nagyobb, karokon átívelő projektek létrejöttébe. A kutatás továbbfejlesztési és egyben alkalmazási lehetősége a FANUC robot kollaboratív robottá tétele. A kollaboratív robotok és azipari robotok közötti legfontosabb különbség, hogy a kollaboratív robotok nem csak nem veszélyesek az emberekre, ha azok bekerülnek a munkaterületükre, hanem együtt tudnak dolgozni velük, azaz képesek a közös munkavégzésre [30]. Ezen okokból manapság egyre nagyobb népszerűségnek örvendenek. Amennyiben a FANUC robot mozgását valamilyen jellegű mozgás leképezésére alkalmas eszközzel, szenzorokkal vagy radarral rögzítjük és lekérdezzük ennek a minden pillanatban vett helyzetét és helyét, akkor az általam kialakított kód képes lenne, hogy minden pillanatban visszaszámítsa annak az inverz kinematikáját és az adott szögértékeket. Abban az esetben pedig, ha ismertek ezek az adatok egy PLC (Programmable Logic Controller) segítségével elkészíthető a rendszer szabályozása. Tehát az eddig vezérléssel működő ipari robotot képesek leszünk átmódosítani egy kollaboratív eszközre, amely valós idejű szabályozással rendelkezik.

A program kód a dokumentumba is láthatóan eleget tett annak a feltételnek, hogy képes az adott típusú szingularitás detektálására és annak azonosítására, hogy milyen esetben és a robot melyik részének a hibájából fog az bekövetkezni. Ezenfelül eleget tett annak a kritériumnak is miszerint attól még hogy adott szingularitási pozícióba kerül az eszköz képes arra, hogy kiszámítsa világ- és csuklókoordinátáit típustól függetlenül. Ez azonban könnyen tovább fejleszthető és egy egyedi lehetőséget rejt magában, hiszen a programkód minimális módosításával elő tudok állítani egy olyan kódot, amely nem csak detektálja az adott pontbeli szingularitást, hanem prediktálja is. Ennek a funkciónak a kifejlesztésével képes lennék a szingularitási pontok egyfajta előrejelzésére és megtudnám akadályozni, hogy a robotot olyan koordinátákhoz vezéreljem, ami hibás működést okozhat. Illetve a kollaboratív robotátalakításhoz kapcsolódó kutatás elengedhetetlen részét képezné a szingularitások előrejelzése hiszen, ha a robot véletlenül bekerülne egy ilyen pozícióba, onnan kezdve a viselkedése beszámíthatatlanná válna és a szabályozás se biztos, hogy meg tudná akadályozni, hogy kárt tegyen a körülötte lévő emberekben.

Látható, hogy a kódom eddig is nagyrészt paraméteres értékekkel működik és a feladat elején elégséges azoknak a beolvasása a számításhoz. Ezek mellett az általam használt hibrid delta-parallel elég ritka és különleges kialakítás, amiből nem sok található meg a világon. Céлом lenne, hogy ezt a kódot át tudjam úgy alakítani, hogy kompatibilis legyen a lehető legtöbb ilyen kinematikával rendelkező robotra. Azt szeretném elérni, hogy egy olyan általánosan működő kódot hozzak létre, ami kisebb módosításokkal a lehető legtöbb konfigurációra működőképes legyen.

5. IRODALOM JEGYZÉK

- [1] V. D. Zoran Pandilov, „Comparison of the characteristics between serial and parallel robots,” *ACTA Technica Corviniensis*, 2014.
- [2] H. W. Walker, „Degrees of Freedom,” *Journal of Educational Psychology*, pp. 253-269, 1940.
- [3] Z.-M. Li, „Functional Degrees of Freedom,” *Motor Control*, pp. 301-310, 2006.
- [4] W. L. L. J. H. Z. LI Yanwen, „Applicability and Generality of the Modified Grübler-Kutzbach Criterion,” *CHINESE JOURNAL OF MECHANICAL ENGINEERING*, 2013.
- [5] S. B. Niku, *Introduction to Robotics*, California: John Wiley & Sons Ltd, 2020.
- [6] R. C. Peter Vischer, „Kinematic calibration of the parallel Delta robot,” *Robotica*, %1. kötet16, p. 208, 1998.
- [7] R. C. M. Bouri, „The Linear Delta: Developments and Applications,” 2010.
- [8] V. S. F. A. M. D. F. F. F. B. S. S. S. Johannes Lachner, „The influence of coordinates in robotic manipulability analysis,” *Science Direct*, 2020.. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0094114X19323286?casa_token=OVqJbJubHTkAAAAA:t_VoM7I8qOBSQyVuuMfXBUjpLUwdnr_UIn1WFZK1ZUzFfKuGQ0HayYOP4Uz4OlAGbbwRLNmXEQ. [Hozzáférés dátuma: 12. 11. 2023.].
- [9] „Forward and inverse kinematics,” *Compas Fab*, [Online]. Available: https://gramaziokohler.github.io/compas_fab/0.14.0/examples/03_backends_ros/03_forward_and_inverse_kinematics.html. [Hozzáférés dátuma: 13. 11. 2023.].
- [10] M. Gyula, *Robotika*, Szeged: Typotex Kiadó, 2011.
- [11] A. S. Mochammad Rizky Diprasetya, „Homogeneous Transformation Matrix Based Neural Network for Model Based Reinforcement Learning on Robot Manipulator,” 2022..
- [12] S. B. Jolly Shah, „End-Effector Position Analysis Using Forward Kinematics for 5 DOF Pravak Robot Arm,” *International Journal of Robotics and Automation (IJRA)*, pp. 112-116, 2013.
- [13] R. G. C. A. D. U. S. K. S. Abdullah Aamir Hayat, „Identification of Denavit-Hartenberg Parameters of an Industrial Robot,” *Conference on Advances In Robotics*, p. 1–6, 2013.
- [14] C. B. M. L. B. D. I. Daniel Constantin, „Forward Kinematic Analysis of an Industrial Robot,” in *New Developments in Mechanics and Mechanical Engineering*, 2015.

- [15] A. F. C. C. A. T. C. A. L. T. a. L. F. O. M. Ruthber Rodríguez Serrezuela, „KINEMATIC MODELLING OF A ROBOTIC ARM MANIPULATOR,” *ARPJ Journal of Engineering and Applied Sciences*, 2017.
- [16] P. S. Donelan, „Singularity - theoretic methods in robot kinematics,” *Robotica*, %1. kötet25, pp. 641-659, 2007.
- [17] P. Robert L. Williams II, Ohio University, 10. 2016. [Online]. Available: <https://www.ohio.edu/mechanical-faculty/williams/html/PDF/DeltaKin.pdf>. [Hozzáférés dátuma: 07. 09. 2023.].
- [18] I. Bonev, „Delta Parallel Robot — the Story of Success,” 2001.
- [19] G. D. V. M. Z. B. M. János Somló, „The „Phantom” Delta Robot A New Device for Parallel Robot Investigations,” *Acta Polytechnica Hungarica*, pp. 153-156, 2018.
- [20] „Delta robot kinematics,” Trossen Robotics Community, [Online]. Available: <https://hypertriangle.com/~alex/delta-robot-tutorial/>. [Hozzáférés dátuma: 10. 11. 2023.].
- [21] L. S. M.A.Laribia, „Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace,” *Mechanism and Machine Theory*, pp. 859-870, 2007.
- [22] S. K. a. Z. Bingul, „Robot Kinematics: Forward and Inverse Kinematics,” *Industrial Robotics Theory Modelling Control*, p. 964, 2006..
- [23] S. Y. K. P.K.Jamwal, Forward kinematics modelling of a parallel ankle rehabilitation robot using modified fuzzy inference, Elsevier, 2010.
- [24] K. Y. Yoji Umetani, „Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix,” *IEEE Transactions on Robotics and Automation*, 1989.
- [25] S. S. Gaurav Tevatia, „Inverse Kinematics for Humanoid Robots,” *IEEE International Conference on Robotics and Automation* , 2000.
- [26] B. Siciliano, „The Tricept robot: Inverse kinematics, manipulability analysis and closed-loop direct kinematics algorithm,” *Robotica*, p. 437–445, 1999.
- [27] R. V. V. P. Florian Ion Tiberiu Petrescu, „Direct and inverse kinematics to the anthropomorphic robots,” *Engevista*, pp. 109-124, 2016.
- [28] E. C. G. G. A. B. M Lo’pez, „Deltarobot: inverse, direct, and intermediate Jacobians,” *Mechanical Engineering Science*, pp. 103-109, 2006.
- [29] S. A. M. E. A. Eltaye, „Kinematic Analysis of Delta Parallel Robot: Simulation Study,” *ENGINEERING RESEARCH JOURNAL (ERJ)* , pp. 28-29, 2021.
- [30] M. A. A. M. A. Abdelfetah Hentout, „Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017,” *ADVANCED ROBOTICS*, pp. 764-799, 2019.

MELLÉKLETEK

1. MELLÉKLET: DIREKT KINEMATIKAI FELADAT

```
disp('Direkt kinematika')

% PARAMÉTEREK
a = 110; %mm
d = 40; %mm
l = 100; %mm
h = 270; %mm
e = 39;
f = 39.5;
g = 29;
tcp = 15;

% BEMENETI ÉRTÉKEK
disp('Szögek')
theta1 = -13.170
theta2 = 10
theta3 = 10
theta4 = -10.425
theta5 = 0.237
theta6 = -14.852

% DELTA RÉSZ
%Vektorok
%1. csukló
MD1 = [-d;
        0;
        0;];
UA1 = [-a;
        0;
        -5;];
A1B1 = [-1*cosd(-theta1+65);
        0;
        -1*sind(-theta1+65)];
UB1 = UA1 + A1B1;
UB1_v = UB1-MD1;
%2. csukló
Rot_plus120 = [cosd(120) -sind(120) 0; % Rotációs mátrix (+120
fokos forgatás Z tengely körül)
               sind(120) cosd(120) 0;
               0 0 1;];
MD2 = Rot_plus120 * MD1;
UA2 = Rot_plus120 * UA1;
A2B2 = Rot_plus120 * [-1*cosd(-theta2+65);
                     0;
                     -1*sind(-theta2+65)];
UB2 = UA2 + A2B2;
UB2_v = UB2-MD2;
```

```

%3.csukló
Rot_min120 = [cosd(-120) -sind(-120) 0; % Rotációs mátrix (-120
fokos forgatás Z tengely körül)
               sind(-120) cosd(-120) 0;
               0          0          1];
MD3 = Rot_min120 * MD1;
UA3 = Rot_min120 * UA1;
A3B3 = Rot_min120 * [-1*cosd(-theta3+65);
                     0;
                     -1*sind(-theta3+65)];

UB3 = UA3 + A3B3;
UB3_v = UB3-MD3;
% Virtuális gömb középpontok
% Az Ub és MD vektor az inverz kinematikánál már ki lettek
számítva
xg1 = UB1_v(1,1); % Gömb középpontok paraméterei
yg1 = UB1_v(2,1); % UB1_comma koordinátái
zg1 = UB1_v(3,1);
xg2 = UB2_v(1,1); % UB2_comma koordinátái
yg2 = UB2_v(2,1);
zg2 = UB2_v(3,1);
xg3 = UB3_v(1,1); % UB3_comma koordinátái
yg3 = UB3_v(2,1);
zg3 = UB3_v(3,1);

% A három gömbből álló egyenletrendszer
%(x-xg1)^2+(y-yg1)^2+(z-zg1)^2=h^2 Gömb UB1_comma középponttal
%(x-xg2)^2+(y-yg2)^2+(z-zg2)^2=h^2 Gömb UB2_comma középponttal
%(x-xg3)^2+(y-yg3)^2+(z-zg3)^2=h^2 Gömb UB3_comma középponttal
a11 = 2*(xg3 - xg1); % Paraméterek az egyenletrendszer felbontása
után
a12 = 2*(yg3 - yg1);
a13 = 2*(zg3 - zg1);
a21 = 2*(xg3 - xg2);
a22 = 2*(yg3 - yg2);
a23 = 2*(zg3 - zg2);
b1 = -(xg1.^2)-(yg1.^2)-(zg1.^2)+(xg3.^2)+(yg3.^2)+(zg3.^2);
b2 = -(xg2.^2)-(yg2.^2)-(zg2.^2)+(xg3.^2)+(yg3.^2)+(zg3.^2);

if not ((a13 == 0)&(a23 == 0))
    if not ((a13 == 0)|(a23 == 0))
        % x = f(y)
        a1 = (a11/a13) - (a21/a23);
        a2 = (a12/a13) - (a22/a23);
        a3 = (b2/a23) - (b1/a13);

        a4 = - a2/a1;
        a5 = - a3/a1;
        % z = f(y)
        a6 = (- a21*a4 - a22)/a23;
        a7 = (b2 - a21*a5)/a23;
        % Másodfokú egyenlet (y) paraméterei
        a = a4^2 + 1 + a6^2;
        b = 2*a4*(a5 - xg1) - 2*yg1 + 2*a6*(a7-zg1);
        c = a5*(a5 - 2*xg1) + a7*(a7 - 2*zg1) + xg1^2 + yg1^2 +
zg1^2 - h^2;

```

```

%x,y és z értékek számítása
    yplus = (-b + sqrt(b^2-4*a*c))/(2*a);
    yminus = (- b - sqrt(b^2-(4*a*c)))/(2*a);
    xplus = a4*yplus + a5;
    xminus= a4*yminus + a5;
    zplus = a6*yplus + a7;
    zminus = a6*yminus + a7;
    if (zplus<0)
        xyzdelta = [ xplus;
                      yplus;
                      zplus;]
    elseif (zminus<0)
        xyzdelta = [ xminus;
                      yminus;
                      zminus;]
    end
elseif (a13 == 0)
    disp('Singularitási pont: zg1=zg3')
    sz1=- xg1^2 - yg1^2 + xg3^2 + yg3^2;
    % x = f(y)
    a4=-a12/a11;
    a5=sz1/a11;
    % z = f(y)
    a6=(a12*a21-a22*a11)/(a11*a23);
    a7=(b2*a11-sz1*a21)/(a11*a23);
    % Másodfokú egyenlet (y) paraméterei
    a = a4^2 + 1 + a6^2;
    b = 2*a4*(a5 - xg1) - 2*yg1 + 2*a6*(a7-zg1);
    c = a5*(a5 - 2*xg1) + a7*(a7 - 2*zg1) + xg1^2 + yg1^2 +
zg1^2 - h^2;
    %x,y és z értékek számítása
    yplus = (-b + sqrt(b^2-4*a*c))/(2*a);
    yminus = (- b - sqrt(b^2-(4*a*c)))/(2*a);
    xplus = a4*yplus + a5;
    xminus= a4*yminus + a5;
    zplus = a6*yplus + a7;
    zminus = a6*yminus + a7;
    if (zplus<0)
        xyzdelta = [ xplus;
                      yplus;
                      zplus;]
    elseif (zminus<0)
        xyzdelta= [ xminus;
                     yminus;
                     zminus;]
    end
end
if (a23 == 0)
    disp('Singularitási pont: zg2=zg3')
    sz2=- xg2^2 - yg2^2 + xg3^2 + yg3^2;
    % y értéke
    yplus=sz2/a22;
    yminus=sz2/a22;
    % z = f(x)
    a6=-(a11)/(a13);
    a7=(a22*b1-a12*sz2)/(a22*a13);

```

```

% Másododfokú egyenlet (x) paraméterei
a=1+a6^2;
b=-2*(xg1-a7*a6+a6*zg1);
c=yplus^2-2*yplus*yg1 + a7*(a7 - 2*zg1) + xg1^2 + yg1^2 +
zg1^2 - h^2;
xplus = (-b + sqrt(b^2-4*a*c))/(2*a);
xminus = (- b - sqrt(b^2-(4*a*c)))/(2*a);
zplus=a7+a6*xplus;
zminus=a7+a6*xminus;
if (zplus<0)
    xyzdelta = [ xplus;
                  yplus;
                  zplus;]
elseif (zminus<0)
    xyzdelta = [ xminus;
                  yminus;
                  zminus;]
end
end
end
if ((a13 == 0)&(a23 == 0))
    disp('Singularitási pont: zg1=zg2=zg3')
    sz1=- xg1^2 - yg1^2 + xg3^2 + yg3^2;
    sz2=- xg2^2 - yg2^2 + xg3^2 + yg3^2;
% y értéke
yplus=sz2/a22;
yminus=sz2/a22;
% x értéke
xplus=(sz1-a12*yplus)/a11;
xminus=(sz1-a12*yminus)/a11;
% Másododfokú egyenlet (z) paraméterei
a = 1;
b = -2*zg1;
c =(xplus-xg1)^2+(yplus-yg1)^2+zg1^2-h^2;
zplus = (- b + sqrt(b^2-(4*a*c)))/(2*a);
zminus= (- b - sqrt(b^2-(4*a*c)))/(2*a);
if (zplus<0)
    xyzdelta = [ xplus;
                  yplus;
                  zplus;]
elseif (zminus<0)
    xyzdelta= [ xminus;
                 yminus;
                 zminus;]
end
end
if ((b^2 - 4*a*c) < 0)
    disp('Képzletbeli megoldás: Nem mind a három gömb
érintkezik - Y nem létezik')
elseif ((b^2 - 4*a*c) == 0)
    disp('Képzletbeli megoldás: Két gömb tangenciálisan egy
pontban érintkezik (a harmadik gömb is átmegy ezen a ponton)')
end
end

```

```

% HIBRID RÉSZ
% Koordinátarendszer beállítása
Dx = [1 0 0 0;
      0 cosd(180) -sind(180) 0;
      0 sind(180) cosd(180) 0;
      0 0 0 1];
Dz = [cosd(90) -sind(90) 0 0;
      sind(90) cosd(90) 0 0;
      0 0 1 0;
      0 0 0 1];

% Negyedik csukló
D45 = [cosd(-theta4) -sind(-theta4)*cosd(90) sind(-
theta4)*sind(90) 0;
      sind(-theta4) cosd(-theta4)*cosd(90) -cosd(-
theta4)*sind(90) 0;
      0 sind(90) cosd(90)
e;
      0 0 0
1];

% Ötödik csukló
D56 = [cosd(theta5) -sind(theta5)*cosd(-90) sind(theta5)*sind(-
90) 0;
      sind(theta5) cosd(theta5)*cosd(-90) -cosd(theta5)*sind(-
90) 0;
      0 sind(-90) cosd(-90)
f;
      0 0 0
1];

% Hatodik csukló
D67 = [cosd(-theta6) -sind(-theta6)*cosd(30) sind(-
theta6)*sind(30) 0;
      sind(-theta6) cosd(-theta6)*cosd(30) -cosd(-
theta6)*sind(30) 0;
      0 sind(30) cosd(30)
g;
      0 0 0
1];

% Szerszám végpontjának meghatározása
Dt = [1 0 0 0;
      0 1 0 0;
      0 0 1 tcp;
      0 0 0 1];
Dz2=[cosd(-90) -sind(-90) 0 0;
      sind(-90) cosd(-90) 0 0;
      0 0 1 0;
      0 0 0 1];

% DH féle homogén transzformációs mátrix
T48 = Dx*Dz*D45*D56*D67*Dt*Dz2;

% SZERSZÁM POZÍCIÓJA
xyz = [ T48(1,4)+xyzdelta(1,1);
        T48(2,4)+xyzdelta(2,1);
        T48(3,4)+xyzdelta(3,1);]

% SZERSZÁM ORIENTÁCIÓJA
forgatasi_yaw = atand(T48(2,1)/T48(1,1))
billentesi_pitch = atand(-
T48(3,1)/(T48(1,1)*cosd(forgatasi_yaw)+T48(2,1)*sind(forgatasi_yaw)))

```

2. MELLÉKLET: INVERZ KINEMATIKAI FELADAT

```

disp('Inverz kinematika')

% PARAMÉTEREK
a = 110; %mm
d = 40; %mm
l = 100; %mm
h = 270; %mm
e = 39; %mm
f = 39.5; %mm
g = 29; %mm
tcp = 15; %mm
thetamin = - 25; % Theta szög alsó határértéke
thetamax = 80 ; % Theta szög felső határértéke

% BEMENETI ÉRTÉKEK
disp('Világkoordináták')
x=
y=
z=
yaw=
pitch=
roll =
syms t1 t2 t3 t4 t5 t6

% HIBRID RÉSZ
% Koordinátarendszer beállítása
Dx = [1 0 0 0;
      0 cosd(180) -sind(180) 0;
      0 sind(180) cosd(180) 0;
      0 0 0 1];
Dz = [cosd(90) -sind(90) 0 0;
      sind(90) cosd(90) 0 0;
      0 0 1 0;
      0 0 0 1];

% Negyedik csukló
D45 = [cosd(t4) -sind(t4)*cosd(90) sind(t4)*sind(90) 0;
      sind(t4) cosd(t4)*cosd(90) -cosd(t4)*sind(90) 0;
      0 sind(90) cosd(90) 0;
      0 0 0 1];
e;

1;];

% Ötödik csukló
D56 = [cosd(t5) -sind(t5)*cosd(-90) sind(t5)*sind(-90) 0;
      sind(t5) cosd(t5)*cosd(-90) -cosd(t5)*sind(-90) 0;
      0 sind(-90) cosd(-90) 0;
      0 0 0 1];
f;

1;];

```

```

% Hatodik csukló
D67 = [cosd(t6) -sind(t6)*cosd(30) sind(t6)*sind(30) 0;
       sind(t6) cosd(t6)*cosd(30) -cosd(t6)*sind(30) 0;
       0 sind(30) cosd(30) 0];
g;
1;];
% Szerszám végpontjának meghatározása
Dt = [1 0 0 0;
      0 1 0 0;
      0 0 1 tcp;
      0 0 0 1];
Dz2=[cosd(-90) -sind(-90) 0 0;
      sind(-90) cosd(-90) 0 0;
      0 0 1 0;
      0 0 0 1];
% DH féle homogén transzformációs mátrix
T48 = Dx*Dz*D45*D56*D67*Dt*Dz2;
% Euler szögek
eqy = -sind(pitch) == T48(3,1);
eqp = cosd(pitch)*sind(roll-180) == T48(3,2);
theta456=solve([eqy,eqp],[t5,t6]);
theta5=-vpa(theta456.t5(1,1))
theta6=vpa(theta456.t6)
if (theta5>-0.1)||(theta5<0.1)
    theta5=0;
end
% Visszahelyettesítés
% Negyedik csukló
D45 = [cosd(-t4) -sind(-t4)*cosd(90) sind(-t4+15)*sind(90)
       sind(-t4) cosd(-t4)*cosd(90) -cosd(-t4+15)*sind(90)
       0 sind(90) cosd(90)
       0 0 0];
e;
1;];
% Ötödik csukló
D56 = [cosd(theta5) -sind(theta5)*cosd(-90)
       sind(theta5)*sind(-90) 0;
       sind(theta5) cosd(theta5)*cosd(-90) -
       cosd(theta5)*sind(-90) 0;
       0 sind(-90) cosd(-90)
       0 0 0];
f;
1;];
% Hatodik csukló
D67 = [cosd(-theta6) -sind(-theta6)*cosd(30) sind(-
theta6)*sind(30) 0;
       sind(-theta6) cosd(-theta6)*cosd(30) -cosd(-
theta6)*sind(30) 0;
       0 sind(30) cosd(30)
       0 0 0];
g;
1;];
% DH féle homogén transzformációs mátrix
T48 = Dx*Dz*D45*D56*D67*Dt*Dz2;

```

```

% Theta 4 meghatározása
eqr2 = cosd(yaw)*cosd(pitch) == T48(1,1);
theta42_values=real(vpa(solve(eqr2,t4)));
if (theta42_values(1,1)>-120)&(theta42_values(1,1)<120)
    theta4=theta42_values(1,1)
else
    theta4=theta42_values(2,1)
end
%theta4= %Manuális beállításhoz, amenyiben nem megfelelő a
szög
% Visszahelyettesítés
% Negyedik csukló
D45 = [cosd(-theta4) -sind(-theta4)*cosd(90)   sind(-
theta4)*sind(90)      0;
        sind(-theta4)  cosd(-theta4)*cosd(90) -cosd(-
theta4)*sind(90)      0;
        0               sind(90)               cosd(90)
e;
        0               0               0
1;];

% Ötödik csukló
D56 = [cosd(theta5) -sind(theta5)*cosd(-90)
sind(theta5)*sind(-90)  0;
        sind(theta5)  cosd(theta5)*cosd(-90) -
cosd(theta5)*sind(-90)  0;
        0               sind(-90)               cosd(-90)
f;
        0               0               0
1;];

% Hatodik csukló
D67 = [cosd(-theta6) -sind(-theta6)*cosd(30)   sind(-
theta6)*sind(30)      0;
        sind(-theta6)  cosd(-theta6)*cosd(30) -cosd(-
theta6)*sind(30)      0;
        0               sind(30)               cosd(30)
g;
        0               0               0
1;];

% DH féle homogén transzformációs mátrix
T48 = Dx*Dz*D45*D56*D67*Dt*Dz2;
% Szingularitás
J = [-f*sind(theta4)-(tcp/2)*sind(theta4+theta6)      0
      -(tcp/2)*sind(theta4+theta6);
      f*cosd(theta4)+(tcp/2)*cosd(theta4+theta6)      0
      (tcp/2)*cosd(theta4+theta6);
      0
      sind(theta5)*(g+cosd(30)*tcp)      0;];
if (det(J)==0)
    disp('Szingularitási pont: A Jacobi mátrix detereminánsa
nulla - Hibrid rész')
end
% Pi paraméterek kiszámítása
Px = x-T48(1,4);
Py = y-T48(2,4);
Pz = z-T48(3,4);

```

```

% DELTA RÉSZ
UM = [Px; % UM vektor mindhárom esetben megegyezik
      Py;
      Pz;];
% Első csukló
MD1 = [-d;
        0;
        0;];
UA1 = [-a;
        0;
        -5;];
A1B1 = [-1*cosd(t1);
        0;
        -1*sind(t1);];
UD1 = UM + MD1;
UB1 = UA1 + A1B1;
B1D1 = UD1 - UB1; % BD vektor kifejezése, hogy ne kelljen a kar és
a láb közötti szöget figyelembe venni
Q11 = Px - d + a; % Q kifejezése annak érdekében, hogy theta
könnyen meghatározható legyen
Q12 = Pz + 5;
G1 = (h^2 - Q11^2 - Q12^2 - Py^2 - 1^2)/(2*1); % G kifejezése
annak érdekében, hogy theta könnyen meghatározható legyen
%Q12*sind(theta1) + Q11*cosd(theta1) = G1;
D1 = sqrt((Q12^2)+(Q11^2));
alfa1 = acosd(Q12/D1);
eqn1 = (G1)/D1 == sind(alfa1 + t1);
theta1_values = 65-vpa(solve(eqn1,t1));
% Második csukló
Rot_plus120 = [cosd(120) -sind(120) 0; % Rotációs mátrix (+120
fokos forgatás Z tengely körül)
               sind(120) cosd(120) 0;
               0 0 1];
MD2 = Rot_plus120 * MD1;
UA2 = Rot_plus120 * UA1;
A2B2 = Rot_plus120 * [-1*cosd(t2);
                      0;
                      -1*sind(t2);];
UD2 = UM + MD2;
UB2 = UA2 + A2B2;
B2D2 = UD2 - UB2; % BD vektor kifejezése, hogy ne kelljen a kar és
a láb közötti szöget figyelembe venni
Q21 = Px + 0.5*d - 0.5*a; % Q kifejezése annak érdekében, hogy
hogy theta könnyen meghatározható legyen
Q22 = Py - 0.866*d + 0.866*a;
Q23 = Pz + 5;
G21 = (h^2 - Q21^2 - Q22^2 - Q23^2 - 1^2)/(1); % G kifejezése
annak érdekében, hogy theta könnyen meghatározható legyen
G22 = -Q21 + 1.732*Q22;
G23 = 2*Q23;
%G23*sind(theta2) + G22*cosd(theta2) = G21;
D2 = sqrt((G23^2)+(G22^2));
alfa2 = acosd(G23/D2);
eqn2 = G21/D2 == sind(alfa2 + t2);
theta2_values = 65-vpa(solve(eqn2,t2));

```

```

% Harmadik csukló
Rot_min120 = [cosd(-120) -sind(-120) 0; % Rotációs mátrix (-120
fokos forgatás Z tengely körül)
              sind(-120) cosd(-120) 0;
              0          0          1;];
MD3 = Rot_min120 * MD1;
UA3 = Rot_min120 * UA1;
A3B3 = Rot_min120 * [-1*cosd(t3);
                    0;
                    -1*sind(t3)];

UD3 = UM + MD3;
UB3 = UA3 + A3B3;
B3D3 = UD3 - UB3; % BD vektor kifejezése, hogy ne kelljen a kar és
a láb közötti szöget figyelembe venni
Q31 = Px + 0.5*d - 0.5*a; % Q kifejezése annak érdekében, hogy
hogy theta könnyen meghatározható legyen
Q32 = Py + 0.866*d - 0.866*a;
Q33 = Pz + 5;
G31 = (h^2 - Q31^2 - Q32^2 - Q33^2 - 1^2)/(1); % G kifejezése
annak érdekében, hogy theta könnyen meghatározható legyen
G32 = - Q31 - 1.732*Q32;
G33 = 2*Q33 ;
%G33*sind(theta2) + G32*cosd(theta2) = G31;
D3 = sqrt((G33^2)+(G32^2));
alfa3 = acosd(G33/D3);
eqn3 = G31/D3 == sind(alfa3 + t3);
theta3_values = 65-vpa(solve(eqn3,t3));
if (theta1_values(1,1)>thetamin)&(theta1_values(1,1)<thetamax)
    theta1=theta1_values(1,1)
else
    theta1=theta1_values(2,1)
end
if (theta2_values(1,1)>thetamin)&(theta2_values(1,1)<thetamax)
    theta2=theta2_values(1,1)
else
    theta2=theta2_values(2,1)
end
if (theta3_values(1,1)>thetamin)&&(theta3_values(1,1)<thetamax)
    theta3=theta3_values(1,1)
else
    theta3=theta3_values(2,1)
end

```