

FELADATKIÍRÁS

Az elektronikusan beadott változatban ez az oldal törlendő. A nyomtatott változatban ennek az oldalnak a helyére a diplomaterv portálról letöltött, jóváhagyott feladatkiírást kell befűzni.



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Automatizálási és Alkalmazott Informatikai Tanszék

Komáromi Sándor

ROS ALAPÚ PÓKSZERŰ ROBOT FEJLESZTÉSE

KONZULENS

Nagy Ákos

BUDAPEST, 2022

Tartalomjegyzék

Összefoglaló	6
Abstract.....	7
1 Bevezetés	8
1.1 Pókszerű járó robot	8
1.2 Új mechanikai modell.....	10
1.3 Célkitűzések.....	11
2 Nyomtatás, tesztelés, összeszerelés	13
2.1 Mechanikai paraméterek	13
2.2 Nyomtatás	15
2.2.1 Csavarok tesztelése	15
2.2.2 Összeszerelési újra tervezések	17
2.3 Összeszerelés és kalibrálás	17
3 Omnidirekcionális járási algoritmus	18
3.1 Láb elérési tartománya.....	19
3.1.1 Az elérési tartomány	19
3.1.2 Láb pályája.....	20
3.1.3 Elérési tartomány középpontja.....	20
3.1.4 Az elérési tartomány továbbfejlesztése.....	21
3.2 Maximális lépéstávolság.....	21
3.2.1 Láb pályájának metszete a lépési tartománnyal.....	21
3.2.2 Lépéstávolság.....	22
3.3 Lépéstávolság kiszámolása a teljes távolság függvényében.....	22
3.4 Lépéssorrend kiválasztása mozgásirány függvényében.....	23
3.4.1 Kritikus szög.....	23
3.5 Láb új pozíciójának kiválasztása	23
3.6 A mozgási algoritmus hibái	23
4 Szimuláció.....	24
4.1 Modell betöltése a szimulációba.....	24
4.2 Algoritmus tervezése szimulációban	24
5 Áttérés fizikai robotra	25
6 Fejlesztési lehetőségek	26

6.1 Kanyarodás megvalósítása.....	26
7 Irodalomjegyzék.....	27

HALLGATÓI NYILATKOZAT

Alulírott **Komáromi Sándor**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző, cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2022. 05. 19

.....
Komáromi Sándor

Összefoglaló

A szakdolgozat, vagy diplomaterv elkészítése minden egyetemi hallgató életében egy fontos mérföldkő. Lehetőséget ad arra, hogy az egyetemi évei során megtanultakat kamatoztassa és eredményeit szélesebb közönség előtt bemutassa, s mérnöki rátermettségét bizonyítsa. Fontos azonban, hogy a dolgozat elkészítésének folyamata számos csapdát is rejt magában. Rossz időgazdálkodás, hiányos szövegszerkesztési ismeretek, illetve a dolgozat készítéséhez nélkülözhetetlen „műfaji” szabályok ismeretének hiánya könnyen oda vezethetnek, hogy egy egyébként jelentős időbefektetéssel készült kiemelkedő szoftver is csak gyengébb minősítést kapjon a gyenge minőségű dolgozat miatt.

E dokumentum – amellet, hogy egy általános szerkesztési keretet ad a dolgozatodnak – összefoglalja a szakdolgozat/diplomaterv írás írott és íratlan szabályait. Összeszedjük a Word kezelésének legfontosabb részeit (címsorok, ábrák, irodalomjegyzék stb.), a dolgozat felépítésének általános tartalmi és szerkezeti irányelveit. Bár mindenkire igazítható sablon természetesen nem létezik, megadjuk azokat az általános arányokat, oldalszámokat, amelyek betartásával jó eséllyel készíthetsz egy színvonalas dolgozatot. A részletes és pontokba szedett elvárás-lista nem csupán a dolgozat írásakor, de akár más dolgozatok értékelésekor is kiváló támpontként szolgálhat.

Az itt átadott ismeretek és szemléletmód nem csupán az aktuális feladatod leküzdésében segíthet, de hosszútávon is számos praktikus fogással bővítheti a szövegszerkesztési és dokumentumkészítési eszköztáradat.

Abstract

Ide jön a ½-1 oldalas angol nyelvű összefoglaló, amelynek szövege a Diplomaterv Portálra külön is feltöltésre kerül. Ez a magyar nyelvű összefoglaló angolra fordított változata.

1 Bevezetés

Manapság egyre inkább előtérbe kerül azon robotok használata, melyek felderítési, felfedezési feladatokat látnak el. Egy ilyen robotnak a talaj minőségétől függetlenül is képesnek kell lennie a problémamentes mozgásra. Ebből kiindulva a teljességre való törekvés nélkül elmondhatjuk, hogy a legtöbb ilyen robot nem kerekek segítségével mozog, hanem lábakon jár, melyek alkalmasak a talaj egyenetlenségének kikompenzálására. Egy felderítő robotnak alkalmasnak kell lennie több irányú mozgásra, hiszen elképzelhető olyan eset, hogy a robot mozgási iránya különbözik annak nézési irányától. Egy felderítésre alkalmas robot esetében nagyon fontos a külvilágról szóló információk gyűjtése, a robotnak fel kell tudnia ismerni a körülötte lévő tér határait, illetve a saját mozgását ebben a térben.

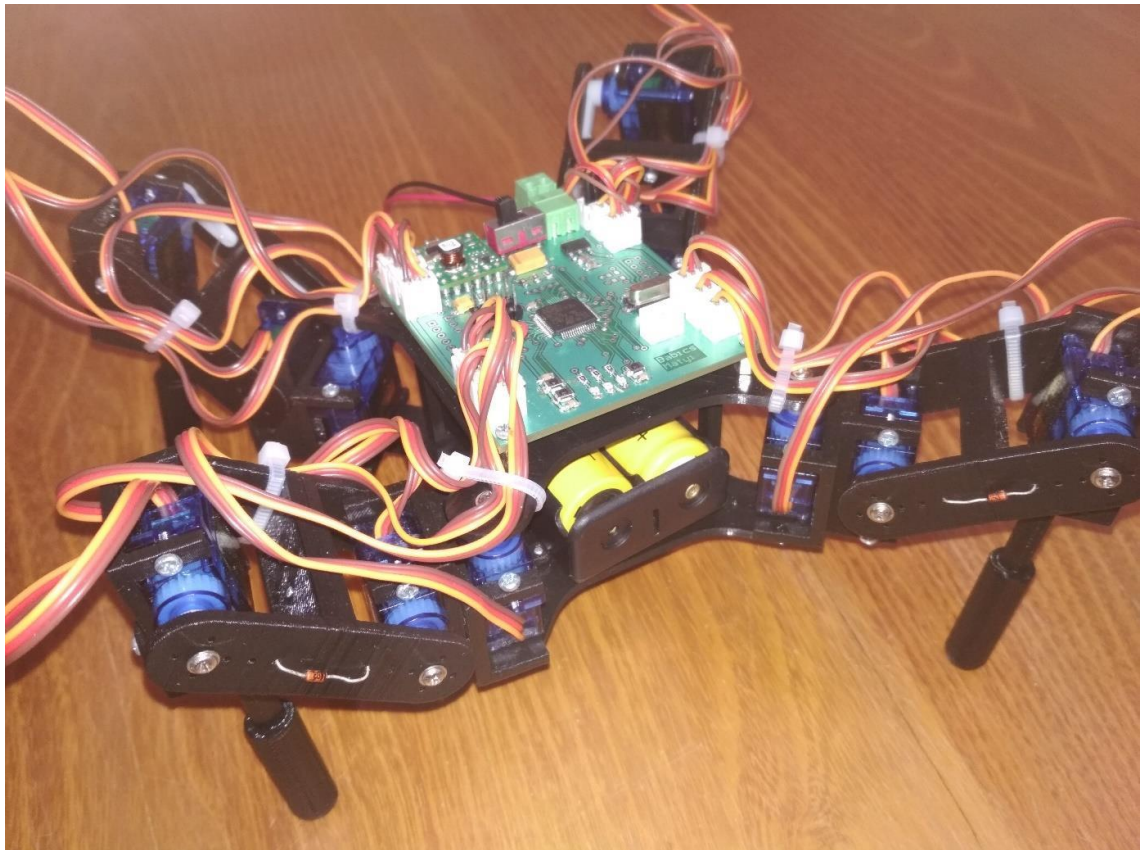
Manapság egyre inkább előtérbe kerül a mobil robotok fejlesztése, egyre több helyen jelennek meg, elsődlegesen az iparban, de egyéb más alkalmazásterületük is feltörekvően van. A mobil robotok egy speciális területe a járó robotok, ezeknek felépítésük bonyolultabb a hagyományos kerék meghajtású robotokkal szemben, viszont alkalmasabbak a talaj egyenetlenségeinek kompenzálására mozgás közben. Ezen robotok elsődleges feladata általában környezetük feltérképezése és bennük a problémamentes navigáció megvalósítása. Ezen robotok mechanikai kialakításából következik, hogy alkalmasak többirányú mozgás megvalósítására, például ugyanúgy tud előre és oldalra is lépkedni.

A szakdolgozatom során két másik hallgató munkáját folytattam. Elsődleges feladatam az önálló laboratóriumom során tervezett robot 3D nyomtatással készített alkatrészek legyártása, összeszerelése. Továbbá feladatam egy olyan új járási algoritmus készítése, mely megvalósítja a fent említett mozgási sémát, miközben a robot orientációjának változtatásáért is felel. Valamint feladatam, az új mechanikával rendelkező robot vissza integrálása Robot Operating System (ROS) [1] interfész alá.

1.1 Pókszerű járó robot

A robot készítése Babits Mátyás munkájával kezdődött. [2] Aki egy négylábú robotot alkotott meg, mely első jellegre pókszerű volt. A robot négy lába egyenként három csuklóból állt, olyan elrendezésben, mint a pókok, rákok, vagy hangyák lábai. A robot

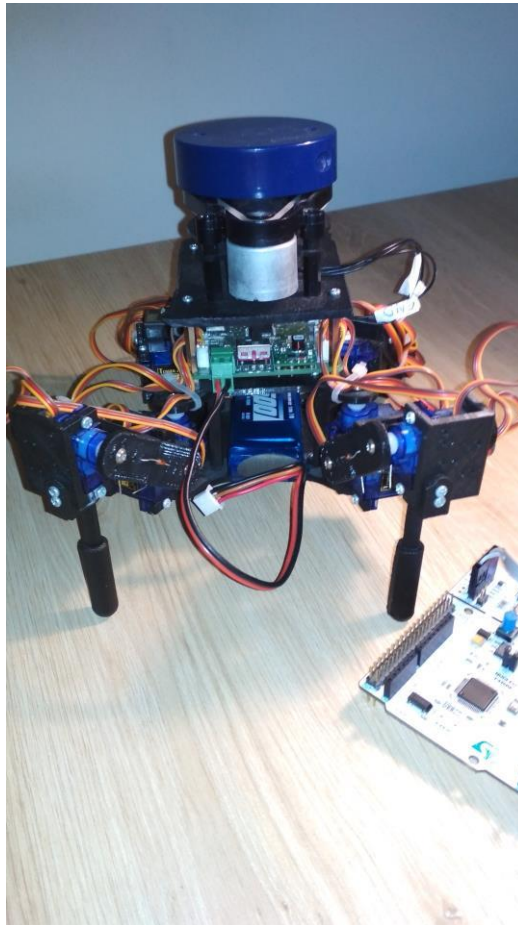
alkatrészei 3D nyomtatással készültek. Lábanként három hajlási pont biztosítja, az elegendő szabadsági fokot ahhoz, hogy egy láb minden pozíciót fel tudjon venni, amelyet a járási algoritmus megkövetel. A Mátyás által készített robot rendelkezett egy olyan beágyazott programmal, mely be tudta mutatni a robot járási képességeit. Kezelte a szervomotorokat, valamint a lábakra kiadott végpontokból kiszámolta inverz geometriai algoritmus segítségével a motorok szögét. A robot mozgása limitált volt, tudott előre menni, valamint jobbra és balra forogni.



1. ábra: Babits Mátyás által készített pókszerű robot.

Mátyás után Massár Lóránt Mátyás fejlesztette tovább a robotot. [3] akinek elsődleges fókusz a robot okosítása volt, szenzorokkal szerelte fel, melyek térképezési és kommunikációs feladatot láttak el. Az okosítás jegyében Lóránt felszerelte a robotot egy Light Detection and Ranging (LIDAR) egységgel, illetve egy Internal Measurement Unit (IMU) egységgel. Előbbi egy olyan lézer alapú távolságérzékelő, mely 360°-ban meghatározza a robottól a külvilág távolságát, míg utóbbi a robot térbeli mozgásáról és elhelyezkedéséről biztosít pontos információkat. Lóránt továbbá felszerelte a robottal való kommunikációt egy ESP-01 wifi modullal a távoli vezérlés megvalósítása érdekében. Va-

lamint megvalósította a robot irányítását ROS platformon keresztül. ROSSerial node segítségével kiépítette a robottal való kommunikációt. RVIZ és hector_slam segítségével környezet feltérképező algoritmust hozott létre.

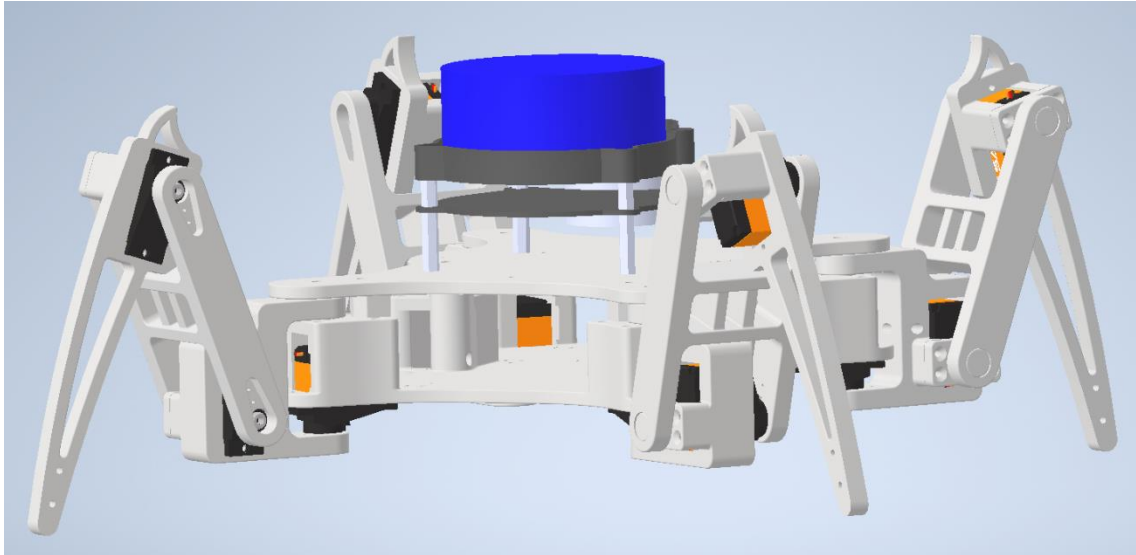


2. ábra: Massár Lóránt Mátyás által készített pókszerű robot.

1.2 Új mechanikai modell

Az önálló laboratóriumom kezdetekor, mikor kézhez kaptam a Lóránt által készített robotot, elsődlegesen a robot mechanikai hibái voltak szembetűnőek. A robot több helyen el volt törve és a szervomotorjai nehezen bírták a LIDAR által megnövelt terhet. Ezért mechanikailag teljesen újra terveztem a robotot, erősebb illesztéseket használtam a csuklóknál, valamint a szerelhetőség megtartása érdekében réz betéteket használtam a csavarok rögzítésére. A servo karok rögzítését jobban megerősítettem, új, erősebb szervomotorokat választottam a robothoz. Valamint további két lábbal kiegészíthetővé tettem a robot modelljét. Az újra tervezés során fontos célkitűzésem volt, hogy a robot lábainak kinematikai kialakítását ne változtassam, azaz a robot lábainak mozgási algoritmusuk ugyanolyan maradjon, csak paraméterei változzanak.

Az önálló laboratóriumom során a robot összeszerelésére nem maradt időm csak annak megtervezésére, melyet a 3. ábra mutatok be. Ezen kívül kiválasztottam a robot új meghajtó motorját, valamint az inverz kinematikai modell paramétereit, melyeket a 2.1 pontban részletezem.



3. ábra: Önálló laborom alatt tervezett robot 3D-s terve.

1.3 Célkitűzések

Célom az volt, hogy a robotot elődjeim munkájának megtartásával tudjam tovább fejleszteni, kiegészíteni. Elkészítsem a robot új mechanikai vázát, valamint új járási algoritmust tervezek, mely alkalmas többirányú mozgás megvalósítására, ez nagyban kiegészítené a robot mozgási lehetőségeit. Szeretném Lóránt munkáját, melyet az 1.1 pontban összefoglaltam, újra működésbe hozni az általam tervezett roboton. Annak érdekében, hogy a fent említett céljaimat meg tudjam valósítani megfogalmaztam az alábbi követelményeket:

- Készüljön el a robot mechanikai megvalósítása a kritikus alkatrészek letesztelésével.
- A robot alkalmas legyen bármilyen irányú mozgás megvalósítására, függetlenül annak nézési irányától.
- A robot mozgás közben alkalmas legyen orientációjának megváltoztatására.
- A robot újra integrálható legyen ROS platform alá.

- A robot vezérelhető legyen ROS platformon keresztül.

A fentebb megfogalmazott céljaim eléréséhez először tesztelnem kell a robot tervek szerkezeti paramétereit és összeszerelhetőségét. Be kell kalibrálnom a robot motorjain lévő szervokarokat, hogy azok megfelelő tartományban mozogjanak. Az általam tervezett járási algoritmus működését meg kell vizsgálnom. egy a Babits Mátyás által készített szimulációban, mielőtt a tényleges roboton tesztelném. Továbbá ki kell egészítenem a robot beágyazott kódját, hogy alkalmas legyen az új motorok használatára és a változott mechanikai paraméterek feldolgozására. Végül újra integrálnom kell a robot rendszerét a ROS környezetbe.

2 Nyomtatás, tesztelés, összeszerelés

Szakedolgozatom megkezdésekor először a robot legyártásával, illetve összeszerelésével foglalkoztam. A végleges gyártás előtt teszt elemeken próbáltam ki az összeszerelhetőségét, kijavítottam a felmerülő hibákat a tervben. A 2.1 pontban részletezem a robot mechanikai paramétereit, melyeknek nagy része változott Mátyás tervei óta. Az alkatrészek gyártása során felmerült problémákat, változtatásokat a 2.2 részben, míg az összeszerelést és a robot kalibrálását a 2.3 fejezetben fejtem ki bővebben.

2.1 Mechanikai paraméterek

Annak ellenére, hogy az Önálló laboratórium beszámolója [4] során már részleteztem a robot új mechanikai paramétereit, szerettem volna ebben a dokumentumban is kifejteni azokat. Ennek két fő oka, hogy a robot pontosabb működése érdekében pontosítottam a robot paraméterein, ugyanis a szimuláció során szabad szemmel is észrevehető volt a paraméterek kerekítése, ezt az 4.1 pontban fejtem ki jobban. Másik oka az, hogy a lentebb definiált paramétereket a dolgozatom során később használom.

A geometriai modell az 1. táblázatban szereplő paraméterek alapján eltolásokból és forgási pontokból épül fel, ahol a d_z , d_x , d_y paraméterek a teljes lábat transzformálják annak első forgási pontjába. Ahonnan a láb további részeit az első motor szöge forgatja, majd a d_1 tovább tolja. A második csukó forgatása után a d_2 paraméter tolja tovább a harmadik csuklóba a láb végpontját, majd a d_3 és d_{3x} által eltol pont adja meg a robot lábának tényleges pozícióját. Ezzel meghatározva a láb előrejele számol geometriai egyenletét, melyet Babits Mátyás jegyzetéből emeltem át a (2.1) egyenlettel együtt, mely jól mátrix szorzásokkal jól leírja a fent említett eltolásokat és forgatásokat. Az egyenletben lévő C_i és S_i paraméterek szinusz és koszinusz műveleteknek felelnek meg.

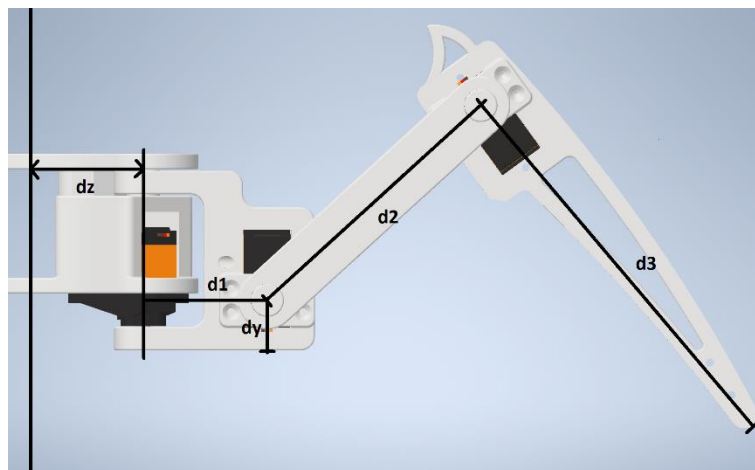
Láb	d_x (cm)	d_y (cm)	d_z (cm)	d_1 (cm)	d_2 (cm)	d_3 (cm)	d_{3x} (cm)
Jobb első	5.2462	1.285	7.4962	3.22	7.5	10.7593	-0.45
Jobb hátsó	-5.2462	1.285	7.4962	3.22	7.5	10.7593	0.45
Bal első	5.2462	1.285	-7.4962	3.22	7.5	10.7593	0.45
Bal hátsó	-5.2462	1.285	-7.4962	3.22	7.5	10.7593	-0.45

1. táblázat: A robot lábaihoz tartozó mechanikai paraméterek

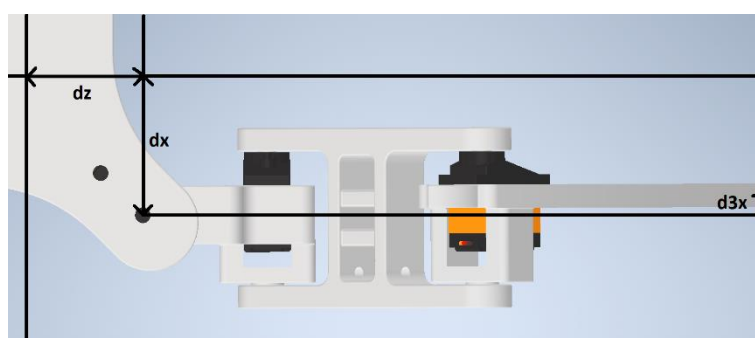
$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_1 & 0 & S_1 & 0 \\ 0 & 1 & 0 & 0 \\ -S_1 & 0 & C_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_2 & -S_2 & 0 \\ 0 & S_2 & C_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_3 & -S_3 & 0 \\ 0 & S_3 & C_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & d_3x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2.1)

Az 1. táblázat lábanként tartalmazza azon paramétereket, melyeket az inverz kinematikai modell felhasznál a motorok új pozíciójának állításához. A robot koordináta rendszere a szokványostól eltérő módon épül fel, a robot szemszögéből nézve a z koordinátatengely mutat előre és az x tengely a z koordinátatengely irányából nézve jobbra. Az előbb említett két tengely nulla pontjai határozzák meg a robot középpontját felülnézetből nézve. Az y tengely az előbb említett középpontból mutat fölfelé, nulla pontja a robot testének legalsó pontjában van. Ezen analógiát követve a d_z és d_x paraméterek minden lábra meghatározzák az első motor tengelyét a robot testének nulla pontjához képest, ezzel eltolva a teljes lábat abba a forgáspontba. A d_y paraméter eltolja a láb kezdőpontját a második motor tengelyének magasságába, ugyan ez az első motorra nincs hatással, hiszen a tengelye mentén történik az eltolás, viszont a második motor e tengely menti eltolását így be lehet állítani. A d_1 eltolja a második motor tengelyét sugár irányba, ezzel meghatározza az első és második csukók tengelyei közötti távolságot. A d_2 hasonló módon a harmadik motor tengelyét helyezi odébb, ezzel meghatározva a második és harmadik motor tengelyei közötti távolságot. A d_3 paraméter a láb végpontja és a harmadik csukló közötti távolságot határozza meg. Végül a d_3x a második és harmadik csukló tengelye mentén tolja el a láb végpontját az első motor tengelyéhez képest. Erre azért van szükség, mert a láb utolsó alkatrésze kissé eltolva érinti a földet az első forgástengelyhez képest. A fenti magyarázat megértésének elősegítése érdekében készítettem a 4. ábra és az 5. ábra, melyek jól mutatják a lábakhoz tartozó paramétereket.



4. ábra: A robot lábának paraméterei oldalnézetből.



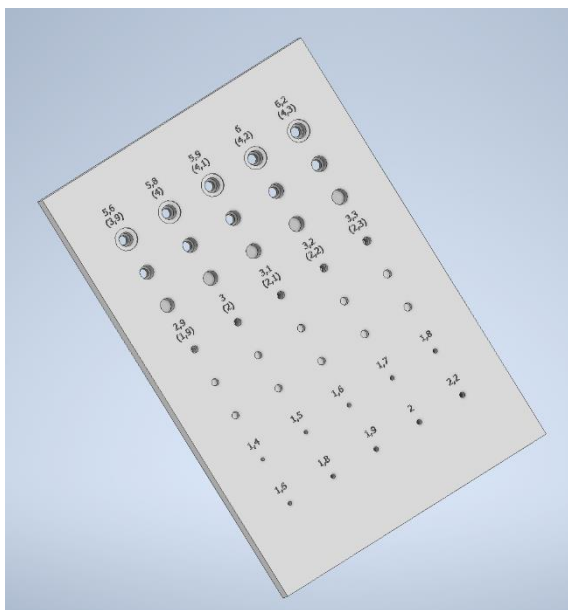
5. ábra: A robot lábának paraméterei felülnézetből.

2.2 Nyomtatás

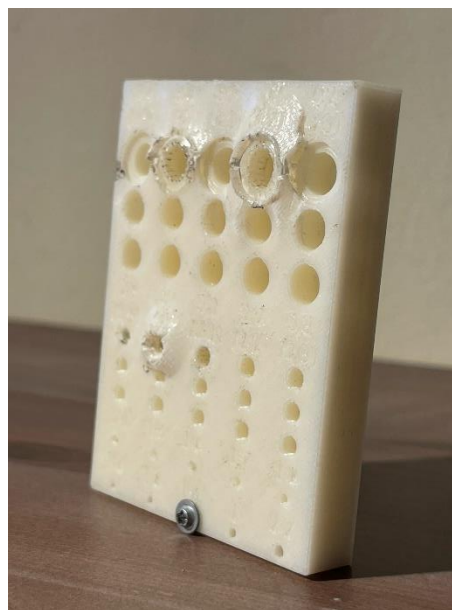
A robot gyártásának megkezdése előtt fontosnak tartottam pár teszt nyomtatás elvégzését, ugyanis ezen terv előtt sosem használtam műanyagba ágyazható réz menetes betétet, ezért külön tesztelést igényelt, erről a 2.2.1 pontban írok. Valamint felmerült bennem egy lehetséges összeszerelési probléma, melyet az 2.2.2 pontban részletezem

2.2.1 Csavarok tesztelése

A menetes betétek, valamint minden egyéb csavar tesztelése érdekében létrehoztam egy egyszerű tesztelési panelt, melyen tized milliméter átmérőű különbségű furatok vannak. Így a végleges tervre olyan átmérőjű furatok kerülnek, melyekbe a legjobban illeszkednek a betétek és egyéb csavarok. A 6. ábra mutat be a tervezett panelt, melyen jól láthatóak a különböző méretű furatok. Minden átmérőjű furatból többet terveztem, hogy különböző rögzítési lehetőségeket is ki tudjak próbálni.



6. ábra: Teszt panel modellje



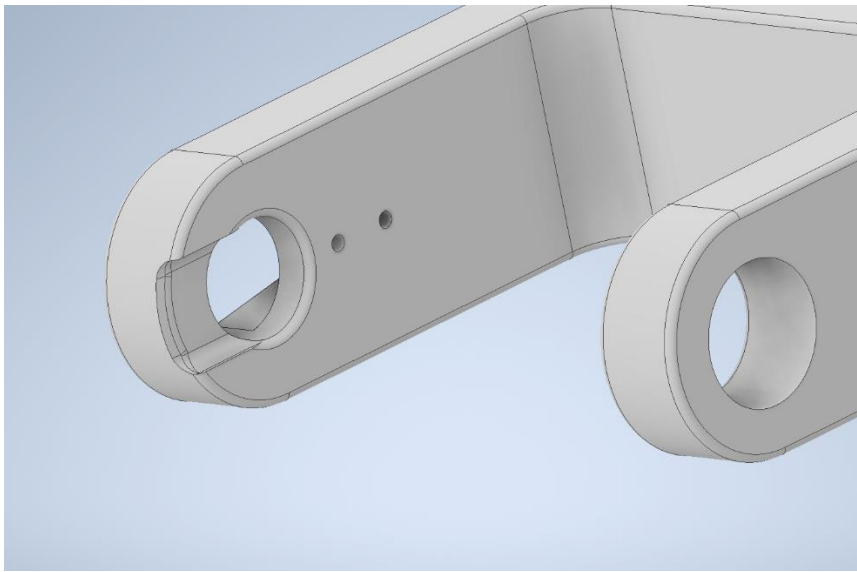
7. ábra: Panel a tesztelés után.

A betétek, melyeket a roboton használok, fogakkal rendelkeznek és többféle behelyezési módszerre találtam példát. Az egyik esetben egy satu segítségével lehet belepréselni a műanyagba a rézbetétet és a csavart fentről rögzíteni, ekkor terhelés hatására kizárólag az előbb említett fogak tartják a betétet. Ennél egy biztosabb megoldás, a peremes betét használata, melybe a furat másik oldaláról érdemes belecsavarni a csavart. Ezen felhasználáskor terhelés hatására a fogak helyett egy széles perem tartja helyén. Erre a rögzítésre a xy ábra mutat példát., azon helyeken, ahol megoldható volt, én is így alkalmaztam. További lehetőség a betét behelyezése közben melegítés használata, melyre egy forrasztópáka alkalmas, ekkor réz jó hővezetőképességéből adódóan a betét körüli műanyag megolvad és a nyomtatáskor létrejött rétegrendek elmosódnak és jobban körül ölelik a betét fogait. Ezzel egy sokkal erősebb és biztosabb rögzítést alkotva, mely a tesztek során beigazolódott. A tesztek eredményét az xy ábra mutatja, azon esetek, mikor a panel belülről tört ki és felpúposodott azok az esetek mikor melegítést is alkalmaztam.

Az erősebb rögzítési lehetőséget végül nem alkalmaztam a végleges megoldásban, mert a melegítés hatására megolvadt műanyag az esetleges oldal irányú erőkkkel szemben nem tartotta a betétet, ezért a folyamat közben eldőlt az alkatrész. Így nem tudtam minden esetben biztosítani a pontos behelyezést és a melegítés nélküli változat is megfelelő tartóerővel rendelkezett.

2.2.2 Összeszerelési újra tervezések

A robot alkatrészeinek végleges nyomtatása előtt készült egy lábnyi alkatrész, tesztelés céljából. A teszt alkatrészek összeszerelése során egy fő probléma lépett fel, melyre előzetesen számítottam. A lábak csuklóinak összeszerelésekor a 8. ábra látható elem két szárát túlzottan szét kellett feszíteni, ezért a végleges tervhez a képen is látható bevágást adtam. A bevágás mellett megnagyobbítottam a távolságot a két szár között ezzel is elősegítve az összeszerelést, ez nem visz tengely menti mozgást a csuklóba, mert a szervokarok felhelyezésekor az egyik oldalra húzza a csuklót.



8. ábra: A könnyebb összeszerelhetőség kedvéért bevágás a robot csuklóin.

2.3 Összeszerelés és kalibrálás

3 Omnidirekcionális járási algoritmus

Egy négylábú robot mozgása elsőre egyszerűbbnek tűnhet, mint egy hat, vagy nyolc lábúé, hiszen kevesebb láb irányítását és lépési pozícióját kell megvalósítani. Gondoljunk csak a hangyák és pókok mozgására, melyek igen kaotikusnak tűnnek, nehéz megállapítani hogyan mozognak. Azonmban kicsit jobban megvizsgálva elsődleges becslésként kimondhatjuk, hogy egyensúlyozás nélküli, statikus [\[HIVATKOZÁS?\]](#) járás esetén azon robotok, melyeknek több lába van stabilabb és gyorsabb mozgást tudnak megvalósítani. Ez abból adódik, hogy a robot stabilitásának megtartása érdekében legalább három lábának mindenféleképpen a földet kell érintse és a robot tömegközéppontja a lábak által alkotott háromszögön belül kell maradjon. Amíg egy négy lábú robot a fentiek alapján egyszerre csak egy lábát tudja felemelni egy hat vagy nyolc lábbal rendelkező esetben már három, illetve négy lábat, a szimmetria megtartása érdekében, emel egyszerre a levegőbe, ezzel egy sokkal gyorsabb és stabilabb mozgást elérve. A fentiek alapján sejtjük, hogy egy négy lábú statikus mozgást megvalósító algoritmus sokkal lassabb, szaggatottabb, talán egy fokkal bonyolultabb eredményhez vezet, mint egy több lábbal rendelkező esetben.

A járási algoritmus újra tervezésekor elsődleges szempontom volt, hogy a régi programmal ellentétben a robot ne csak egyenesen tudjon menni és a tengelye körül tudjon forogni [\[HIVATKOZÁS\]](#), hanem bármelyik irányba el tudjon indulni, függetlenül attól, hogy milyen irányba áll az eleje. A robot újra tervezésekor, e célt figyelembe véve, úgy alakítottam ki a terveket, hogy az a tengelyekre szimmetrikus legyen, így a robot előre és oldalra a lépéssorrenden kívül ugyanúgy tud lépni. Azt változtatva, hogy mennyit lépjen előre és oldalra, a robot tetszőleges irányba mozgatható úgy, hogy végig egy irányba néz, tehát mechanikailag minden irányba mozgatható.

A fent említett általános mozgási algoritmus tervezésekor figyelembe vettem, hogy a robot ROS [\[HIVATKOZÁS\]](#) környezetbe való integrálása a célom, melyre az **XY** pontban térek ki részletesebben. A mozgási parancs két paramétert vár, a cél pozíció koordinátáit a robot koordináta tengelyében, azaz az elmozdulás vektort, illetve azt az elfordulás mennyiséget, melyet a célpontba érve elforduljon a robot. Algoritmusom több különálló részre bontható, először kiszámolja a legnagyobb láb lépés távolságot, melyet a lábak elérési tartománya határol. Majd kiszámolja a lelépendő távolság és az elfordulandó

szög alapján azt a lépéstávolságot, illetve lépésenkénti elfordulást, melyet a robot ténylegesen lépni fog. A tényleges lépéstávolság nem minden esetben egyenlő a maximummal, hogyha a lelépendő távolság nem egész számú többszöröse a maximum lépéstávolságnak. A továbbiakban az algoritmus a mozgás irányának függvényében összeállítja a lábak lépésének sorrendjét, majd végül kiszámolja minden egyes lábhoz a hozzá tartozó új pozíciót.

3.1 Láb elérési tartománya

A mozgási algoritmus elkészítése előtt, meg kellett vizsgálnom a robot lábainak elérési tartományát, melyek a mechanikai kialakításuk határol. A robot egy lábának elérési tartományának nevezzük azt a területet, melyen belül a láb mechanikailag képes kinyúlni [HIVATKOZÁS?]. A legtöbb ezzel a témakörrel foglalkozó cikkben egy téglalappal egyszerűsítik le a lábak elérési tartományát az egyszerűbb számolás érdekében [HIVATKOZÁS]. Az egyszerűsítést én nem alkalmaztam, mert a téglalap alakú elérési tartomány esetében a robot előre nagyobb távolságot tud lépni, mint oldalra. Számomra elsődleges cél volt a robot szimmetriájának megtartása így a felhasználtam a teljes mechanikailag megengedett tartományt, melyet 3.1.1 fejezetben részletezek. Az elérési tartomány megválasztásakor figyelembe vettem a robot lábainak számának bővítési lehetőségét is, melyre a 3.1.4 fejezetben térek ki bővebben.

3.1.1 Az elérési tartomány

A továbbiakban bemutatom egy láb elérési tartományának felépítését és paramétereit, elegendő egy tartományt részleteznem, mert a többi ennek tükrözésével kiszámolható. A robot koordináta rendszere úgy épül fel, hogy a robot testének középpontja az origó és a négy láb a négy síknegyedben helyezkedik el. A robot eleje a z tengely fele, míg a jobb oldala az x tengely fele néz, így a jobb mellső lábának koordinátái pozitívek. Az y tengely mentén a robot hasmagassága állítható.

Egy láb mechanikai paramétereit és összeállítását a 2.1 fejezet részletezi, az elérési tartomány szempontjából az a fontos, hogy a három forgó pont közül az első az y tengely körül forog, míg a második és a harmadik az x tengely körül. A robot lábának inverzkinematikai modellje x, y, z koordinátákat használ a pozíció megadására. Viszont a robot lábának kialakítása miatt a láb egy pozíciója sokkal jobban leírható egy α szög és

egy r sugár segítségével, ahol előbbi az első forgó pont szögének felel meg, utóbbi a második és harmadik forgó ponttal egy egyenes mentén növelhető a távolság.

A fent leírtak alapján következik, hogy az elérhető tartomány egy kör valamilyen szelete, a láb mechanikai paraméterei korlátozzák, hogy mennyire messzire tud nyúlni, ezzel egy maximum nyúlási távolságot számolható ki. A körszelet nagyságát tovább korlátozza az első forgó pontba épített szervó forgási korlátja, melynek teljes tartománya ugyan 145° viszont a tipikus használati tartománya csak 90° fok így én az utóbbi értéket használtam fel. Egy láb elérési tartományát tovább szűkíti az, hogy a végpont nem képes teljesen a y tengely körüli forgó pont alá hajolni, hanem csak egy adott paraméter értékéig képes behúzni azt. Ebből következik, hogy az elérési tartomány határolja egy kisebb körív is, melyen belüre nem tudja húzni a lábát a robot.

Összefoglalva **...kép alapján összefoglalni... meg képre hivatkozni stb.**

3.1.2 Láb pályája

A robot lábának mozgási pályáját egy egyenes határolja. A pálya járás közben minden esetben érinti a 3.1.3 pontban részletezett középpontot. A pálya irányultságát elsősorban a robot mozgási iránya szabja meg, melyet az xy képen α szög jelez. A robot mozgási pályája minden esetben két ponton metszi az elérési tartományt, ezen két pont segítségével könnyedén kiszámolható a láb lépésének hossza, melyet a 3.2 fejezet tárgyal részletesebben.

3.1.3 Elérési tartomány középpontja

A mozgási algoritmus tervezését nagyban megkönnyíti az, ha a láb pályája, függetlenül annak szögétől, minden esetben átmegy egy fix ponton, ezzel meghatározva egy láb elérési tartományának C_i középpontját [**HIVATKOZÁS**]. Az xy képen látható C_i pontot a 3.1.1 fejezetben részletezett elérési tartományba illesztettem be, így annak koordinátáit sugár-szög párral adtam meg. A szimmetria megtartásának érdekében a C_i szöge a teljes lefedettség felével egyezik meg, míg sugara az elérési tartomány két körívének átlaga.

képletet beszúrní erről

A mozgás irányok közötti váltást tovább egyszerűsíti a C_i pont, ugyanis így mozgásiránytól függetlenül létezik olyan pont mely mindig részese a mozgás pályájának. A járási algoritmus befejezésekor keletkezik olyan pont, mely ebben a középpontban van.

3.1.4 Az elérési tartomány továbbfejlesztése

3.2 Maximális lépéstávolság

A maximális lépéstávolság az a távolság, amely a robot mozgás iránya függvényében meghatározza mi az a legnagyobb lépés, amely megtétele esetén még éppen egyik láb sem lép ki az elérési tartományából. Ezen szám kiszámításához először ki kell számolni mind a négy lábhoz tartozó lépéstávolságot. A járási algoritmus egyszerűsítése érdekében a négy lépéstávolság minimumát vettem, mert így biztosan egyik láb sem fog kilépni a saját területéről.

3.2.1 Láb pályájának metszete a lépési tartománnyal

A fent említett lábankénti lépéstávolság meghatározásához először elérési tartomány és a láb pályájának metszéspontjait számoltam ki, mivel a láb pályája minden esetben érinti az elérési tartomány középpontját, így minden esetben két metszéspont keletkezik. A számítás során kihasználtam azt, hogy az elérési tartomány egy 90° -os, körívhez hasonlít, mert így a számítás során felhasználhattam azt, hogy a négy láb elérési tartománya leír egy teljes kört, melyet a síknegyedek határolnak. A számítás során elegendő volt a lábak forgáspontját az origóba tolnom, így az elérési tartományt határoló két körív a négy lábra két kört alkotott. A tartományt határoló két egyenes pedig a tengelyekre esett.

A metszéspontok kiszámítását egy lábhoz számolja ki a függvény, minden lábhoz tartozik egy síknegyed és mindegyiket reprezentáltam egy olyan vektorral, melynek mind az x , mind az y koordinátájának abszolút értéke egy, tehát csak előjelükben van különbség. Ezzel létrehozva egy eltolás vektort.

A láb pályájának egyenesét megszorozva ezzel az eltolás vektorral kiválasztottam, hogy melyik lábhoz tartozó metszéspontokat számolja az algoritmus. Függetlenül attól, hogy melyik síknegyedhez tartozó metszéspontok fontosak, mindig mindkét körre, x tengelyre és y tengelyre meghatározza az összeset. Egy egyenes és egy körnek legfeljebb két metszéspontja lehet, míg két egyenesnek egy-egy. Ezért a legfeljebb hat metszéspontot megszorozza a fentebb említett eltolásvektorral, ennek eredménye az lesz, hogy azon pontok kerülnek, a két kör esetében, a pozitív tartományba, a két tengely esetében a két kör sugarának pozitív értékei közé, melyek a keresett elérési tartományhoz tartoznak. A

megfelelő tartományba eső pontokat elmenti az algoritmus egy pontokból álló listába későbbi felhasználásra. A lista használata azért előnyös, mert így, ha egy harmadik pont hibásan bekerülne a metszéspontok közé leellenőrizhető lesz.

Mivel a számítások a szimuláció után a mikrokontrolleren fognak megvalósulni float változók felhasználásával. Ezek nem a legpontosabb számábrázolást valósítják meg, ezért felhasználtam egy epsilon változót a számítási pontosság tűréshatárának beállítására.

3.2.2 Lépéstávolság

A maximális lépéstávolság kiszámítása előtt a négy láb lépéstávolságát számolja ki az algoritmus. A 3.2.1 részben leírtak alapján megkapott két pont közül az lesz a megfelelő, melyet kivonva az elérési tartomány középpontjából a robot mozgásával közel megegyező irányú vektort kapunk. A számítás során a fentebbiekhez hasonlóan felhasználtam egy epsilon változót a pontatlanságból eredő hiba elkerülésére.

A négy lábhoz tartozó lépéstávolság kiszámítása után a maximális lépéstávolság kiszámítása már nem okozott problémát. Ugyan a változó neve ebből a környezetből nézve pontatlan, azon legnagyobb távolságra utal, melyet még minden láb leléphet mielőtt elérné az elérési tartomány szélét. Ezek után egyértelművé válik, hogy a négy lépéstávolság minimumát kell venni az érdekében, hogy semmikor ne léphessen egyik láb sem ki az elérési tartományból.

3.3 Lépéstávolság kiszámolása a teljes távolság függvényében

A 3.2 fejezetben részletezett maximális lépéstávolság számításakor a robot fizikai korlátait használtam fel, azonban a tényleges lépéstávolság kiszámításhoz további paramétereket is figyelembe kell venni. Ezek a lelépendő távolság, illetve az út alatti elfordulás, melyet a későbbiekben relatív elfordulásnak nevezek.

Jelen helyzetben a robot mozgása csak akkor egyenletes, ha a lelépendő távolság nem egész többszöröse a maximális lépéstávolságnak a lépések legvégén egy maradék, kisebb, lépés marad, mely megtöri a mozgást. A robot algoritmusában ezen kis lépés eltüntetése érdekében a lelépendő távolságot elosztva a 3.2 fejezetben számolt maximális lépéstávolsággal kiszámolja azt, hogy hány lépés alatt tudná a robot megtenni az utat. A lépésszám számolásakor a maximális lépéstávolságot használta az algoritmus, ezért a ka-

pott eredmény a lépésszám minimumát képezi. A robot csak egész számú lépést tud megtenni, ezért az előbb számított lépésszámot egész számra felkerekíti, majd elosztja vele a lelépendő távolságot, így megkapva azt a lépéstávolságot, mellyel egységes lépésekkel lépi le a robot a kiadott távolságot. A fentihez hasonló módon kiszámolja a lépésenkénti forgási szöget is.

Abban az esetben, ha a robotnak forognia is kell mozgás közben, elképzelhető olyan variáció, jellemzően kis lelépendő távolság esetén, hogy a kiszámolt lépésszám megtétele alatt nem a robot nem tud elegendőt forogni. Ekkor az algoritmus újra számolja a lépésszámot a maximális lépésenkénti elfordulással és a fentihez hasonló módon kiszámolja a lépésenkénti lépéstávolságot és elfordulási szöget.

3.4 Lépéssorrend kiválasztása mozgásirány függvényében

3.4.1 Kritikus szög

3.5 Láb új pozíciójának kiválasztása

3.6 A mozgási algoritmus hibái

A robot forgása

4 Szimuláció

4.1 Modell betöltése a szimulációba

4.2 Algoritmus tervezése szimulációban

5 Áttérés fizikai robotra

6 Fejlesztési lehetőségek

6.1 Kanyarodás megvalósítása

7 Irodalomjegyzék

- [1] ROS, „Robot Operating System,” [Online]. Available: <https://www.ros.org/>.
- [2] M. Babits, „Járó robot készítése,” 2018.
- [3] L. M. Massár, „Pókszerű, járó robot fejlesztése,” 2020.
- [4] S. Komáromi, „Pókszerű, járó robot fejlesztése,” 2021.