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.



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Komáromi Sándor

Diplomaterv Útmutató

Konzulens

BUDAPEST, 2022

Tartalomjegyzék

[Összefoglaló 5](#_Toc103207633)

[Abstract 6](#_Toc103207634)

[1 Bevezetés 7](#_Toc103207635)

[1.1 Pókszerű járó robot 7](#_Toc103207636)

[1.2 Új mechanikai modell 8](#_Toc103207637)

[1.3 Célkitűzések 9](#_Toc103207638)

[2 Nyomtatás, tesztelés, összeszerelés 10](#_Toc103207639)

[2.1 Mechanikai paraméterek 10](#_Toc103207640)

[2.2 Nyomtatás 10](#_Toc103207641)

[2.3 Összeszerelés 10](#_Toc103207642)

[2.4 Szervo 10](#_Toc103207643)

[3 Szimuláció 11](#_Toc103207644)

[4 Omnidirekcionális járási algoritmus 12](#_Toc103207645)

[4.1 Láb elérési tartománya 13](#_Toc103207646)

[4.1.1 Az elérési tartomány 13](#_Toc103207647)

[4.1.2 Láb pályája 14](#_Toc103207648)

[4.1.3 Elérési tartomány középpontja 14](#_Toc103207649)

[4.1.4 Az elérési tartomány továbbfejlesztése 15](#_Toc103207650)

[4.2 Maximális lépéstávolság 15](#_Toc103207651)

[4.2.1 Láb pályájának metszete a lépési tartománnyal 15](#_Toc103207652)

[4.2.2 Lépéstávolság 16](#_Toc103207653)

[4.2.3 16](#_Toc103207654)

[4.3 Lépéstávolság kiszámolása a teljes távolság függvényében 16](#_Toc103207655)

[4.4 Lépéssorrend kiválasztása mozgásirány függvényében 17](#_Toc103207656)

[4.4.1 Kritikus szög 17](#_Toc103207657)

[4.5 Láb új pozíciójának kiválasztása 17](#_Toc103207658)

[5 Áttérés fizikai robotra 18](#_Toc103207659)

[6 Irodalomjegyzék 20](#_Toc103207660)

[Függelék 21](#_Toc103207661)

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, 2015. 10. 19

...…………………………………………….

Kővári Bence

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 – amellett, 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.

# 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.

A szakdolgozatom során két másik hallgató munkáját folytattam, elsődleges feladatom az önálló laboratórium során tervezett robot legyártása illetve új járási algoritmus készítése, mely megvalósítja a fent említett többirányú mozgást.

## Pókszerű járó robot

A robot készítése Babits Mátyás munkájával kezdődött, Aki négylábú robotot alkotott meg, mely első jellegre pókszerű volt. A robot négy lába egyenként három csuklóból álltak, olyan elrendezésben, mint a pókok, rákok, vagy hangyák lábai. A robot minden alkatrésze 3D nyomtatással készült. 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 alap programmal, mely be tudta mutatni a robot járási képességeit, kezelte a szervomotorok állását, valamint kiszámolta a lábak végpontjából inverz kinematikai modellel a szervomotorok szögét. A robot mozgása limitált volt, tudott előre menni, valamint forogni jobbra és balra.

Kép

Mátyás után Massár Lóránt Mátyás fejlesztette tovább a robotot, akinek elsődleges fókusza 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 robotot egy ESP-01 wifi modullal a távoli vezérlés megvalósítása érdekében. megvalósította a robot ROS rendszerbe való integrálását, ROSSerial node segítségével kiépítette a robottal való kommunikációt, míg az RVIZ és hector\_slam nodeok segítségével létrehozta a feltérképező algoritmust.



. ábra: Massár Lóránt Mátyás által készített pókszerű robot.

## Új mechanikai modell

Az önálló laboratóriumomon kezdetekor, mikor kézhez kaptam a Lóránt által készített robotot, elsődlegesen a robot mechanikai hibái voltak problémák, több helyen el volt törve. Ezért mechanikailag teljesen újra terveztem a robotot, erősebb illesztéseket használtam a csuklóknál, valamint a szerelhetőség megtartása érdekében réz betéteket használtam a csavaroknál. A szervo karok rögzítését jobban megerősítettem, új szervomotorokat választottam a robothoz és mechanikai lehetőséget adtam a robot továbbfejlesztésére, két lábbal kiegészíthetővé tettem azt. Az újra tervezés során fontos célkitűzésem volt, hogy a robot lábainak kinematikai kialakításához ne nyúljak, azaz a robot járásának modellje ugyanolyan maradjon, csak paraméterei változzanak.

Az önálló laboratóriumom során a robot összeszerelésére nem maradt időm csak annak megtervezésére, melyet a 2. ábra mutat. Ezen kívül kiválasztottam a robot új meghajtó motorját, valamint az inverz kinematikai modell paramétereit, melyet az 2.1 pont tárgyal részletesebben.



2. ábra: Önálló laborom alatt tervezett robot

## Célkitűzések

Célom az volt, hogy a robotot elődjeim munkájának megtartásával tudjam tovább fejleszteni, kiegészíteni. Elkészítsem a robot új mechanikai vázát, valamint új járási algoritmust tervezzek, mely alkalmas többirányú mozgásra. Szeretném Lóránt munkáját, melyet az 1.1 pontban összefoglalok, újra működésbe hozni, az általam tervezett roboton. Annak érdekében, hogy a fent említett céljaimat meg tudjam valósítani megfogalmaztam a következő követelményeket:

* Készüljön el a robot mechanikai megvalósítása, a terv kritikus részeit teszteljem le.
* A robot alkalmas legyen bármilyen irányú mozgás megvalósítására, függetlenül annak nézési irányától.
* A robot újra integrálható legyen egy ROS alapú rendszerbe.

A fentebb megfogalmazott céljaim eléréséhez először tesztelnem kell a robot tervének szerkezeti paramétereit és összeszerelhetőségét. A robot összeszerelésének befejezése elött a járási algoritmust tesztelnem kell egy Babits Mátyás által készített szimulációban. 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.

# Nyomtatás, tesztelés, összeszerelés

## Mechanikai paraméterek

## Nyomtatás

tesztelés-> insert tesztelése -> maradtam a melegítés nélkül, láb összeszerelésének tesztelése

problémák, újratervezés

értékelés -> kevés újratervezésre volt szükség, az insertek egyből jók voltak

## Összeszerelés

értékelés

## Szervo

# Szimuláció

# 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 sejthetjük, 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 koordinátait 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.

## Láb elérési tartománya

A mozgási algoritmus elkészítése elött, 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 4.1.1 fejezetben részletezek. Az elérési tartomány megválasztásakor figyelembe vettem a robot lábainak számának bővítési lehetőségét is, melyre a 4.1.4 fejezetben térek ki bővebben.

### Az elérési tartomány

A továbbiakban bemutatom egy láb elérési tartományának felépítését és paramétereit, elegendő egy tartományt részleteznem, mert a többi ennek tükrözésivel kiszámolható. A robot koordináta rendszere úgy épül fel, hogy a robot testének középpontja az origó és a négy láb a négy síknegyeden 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 szervo 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 értéjéig képes behúzni azt. Ebből következik, hogy az elérési tartomány határolja egy kisebb körív is, melyen belűre nem tudja húzni a lábát a robot.

Összefoglalva …kép alapján összefoglalni… meg képre hivatkozni stb.

### Láb pályája

A robot lábának mozgási pályáját egy egyenes határolja. A pálya járás közben minden esetben érinti a 4.1.3 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 eléréi tartományt, ezen két pont segítségével könnyedén kiszámolható a láb lépésének hossza, melyet a 4.2 fejezet tárgyal részletesebben.

### 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 Ci középpontját [HIVATKOZÁS]. Az xy képen látható Ci pontot a 4.1.1 fejezetben részletezett elérési tartományba illesztettem be, így annak koordinátáit sugár-szög párral adtam meg. A szimmetria megtartásának érdekében a Ci szöge a teljes lefedettség felével egyezik meg, míg sugara az elérési tartomány két körívének átlaga.

képletet beszúrni erről

A mozgás irányok közötti váltást tovább egyszerűsíti a Ci 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.

### Az elérési tartomány továbbfejlesztése

## 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.

### 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.

### Lépéstávolság

A maximális lépéstávolság kiszámítása elött a négy láb lépéstávolságát számolja ki az algoritmus. A 4.2.1 részben leírtak alapján megkapott két pont közül az lesz a megfejelő, melyet kivonva az elérési tartomány középpontjából a robot mozgásával közel megegyező irányú vektort kapunk. A számítás során a föntebbiekhez 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.

## Lépéstávolság kiszámolása a teljes távolság függvényében

A 4.2 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 algoritmusa ezen kis lépés eltüntetése érdekében a lelépendő távolságot elosztva a 4.2 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 kapott 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.

## Lépéssorrend kiválasztása mozgásirány függvényében

### Kritikus szög

## Láb új pozíciójának kiválasztása

## A mozgási algoritmus hibái

A robot forgása

# Áttérés fizikai robotra

# Fejlesztési lehetőségek

## Kanyarodás megvalósítása

# Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | K. Nahtkasztlija, „Az idegen szavak toldalékolása,” június 2009. [Online]. Available: http://www.pcguru.hu/blog/kredenc/az-idegen-szavak-toldalekolasa/5062. |
| [2] | P. Koopman, „How to Write an Abstract,” október 1997. [Online]. Available: https://users.ece.cmu.edu/~koopman/essays/abstract.html. [Hozzáférés dátuma: 20 október 2015]. |
| [3] | W3C, „HTML, The Web’s Core Language,” [Online]. Available: http://www.w3.org/html/. [Hozzáférés dátuma: 20 október 2015]. |

Függelék