

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM

GÉPÉSZMÉRNÖKI KAR

MECHATRONIKA, OPTIKA ÉS GÉPÉSZETI INFORMATIKA TANSZÉK

HORVÁTH ZSONGOR ÉS KISS-MEIROSU BALÁZS MÁRK

A CISZTERCI REND NAGY LAJOS GIMNÁZIUM ÉS KOLLÉGIUM

11. OSZTÁLYOS TANULÓI

TUDOMÁNYOS DIÁKKÖRI KONFERENCIA DOLGOZAT

Négy lábú robot mozgás elemzés LEGO Mindstorms elemkészlettel

Témavezető:

Dr. Czmerk András

Budapest, 2024

Tartalomjegyzék

Absztrakt	3
1. Bevezetés	4
2. Irodalomkutatás	5
3. Magáról a Mindstorms EV3-ról	6
Bevezetés	6
A Készlet tartalma és Kiadási verziók	7
Programozási lehetőségek és Oktatási verzió	7
4. A robot mozgásának logikájának bemutatása, a kutya felépítése	8
4.1 Rendelkezésünkre álló eszközök	8
4.2 Négylábú mozgás kutyamódra (elképzelés)	9
4.3 A robot mechanikai felépítése és a négy lábú mozgás megvalósítása	10
Központi rész	10
csípő és rotáció irányadás	11
Lábak és azok kapcsolata a motorhoz	13
A lábak felfüggesztése a motorhoz	14
Talpak, lábfejek	16
A fejrész bemutatása	17
4.4 Mozdásmechanika és a fizikai problémák kiküszöbölése	20
Súlypontáthelyezés	20
Fejlődés szakaszai	21
4.5 Mérések	24
5. Programozás, a projekt szoftver specifikációja	27
5.1 Bevezetés	27
5.2 Eszközök és környezet	27
5.3 Módszertan	27
5.4 Funkcionális követelmények	28
5.5 Nem-funkcionális követelmények	28
5.6 Technikai követelmények	28
5.7 Fő funkciók és működési logika kódrészletekkel	29
a. Előre mozgás (move_forward)	29
b. Megállás (stop_motors)	29
c. Hátramenet (move_backward)	29
d. Balra és jobbra nézés az akadály kikerüléséhez (check_left_and_right)	30
e. Akadályérzékelés és hangjelzés (announce_obstacle)	30
5.8 Felhasználói interakciók	31
6. Összegzés	32
Források	33

Absztrakt

A projekt célja egy olyan periodikusan ismétlődő mozgás megvalósítása volt, amely lehetővé teszi, hogy egy négy lábon járó robot stabilan és hatékonyan közlekedjen két motor segítségével, különböző sebességek mellett, minimalizálva a súlypontáthelyezéssel kapcsolatos problémákat. A megvalósításhoz LEGO Mindstorms EV3 platformot használtunk, amely lehetőséget biztosít a robot mozgásának precíz vezérlésére és a lábak szinkronizált működtetésére. A robot stabilitása érdekében figyelembe kellett venni a súlypont dinamikáját, valamint a járás közbeni mozgás és erők hatását a robot egyensúlyára.

A projekt során a mechanikai és programozási kihívások megoldására koncentráltunk, különös figyelmet fordítva a lábak elrendezésére, az egyes motorok működésére és a robot dinamikai viselkedésére. Az Intelligent Brick porthiánya miatt két motorral oldottuk meg a robot járását, ami merőben megváltoztatta a robot dinamikai és programozási feltételeit. A kísérletek során különböző sebességi beállításokat teszteltünk és optimalizáltuk a járás stabilitását, hogy a robot a különböző sebességeken is képes legyen a járófelületektől függően, zökkenőmentesen közlekedni anélkül, hogy jelentős súlypontáthelyezési problémák lépnének fel.

A projekt hozzájárult a robotikai irányítási rendszerek két motoros, egyszerűbb mechanikai konstrukciók stabilitásának és mozgásának optimalizálásához. Az eredmények hasznosíthatók olyan alkalmazásokban, ahol a költségvetési korlátok, a mechanikai egyszerűség és a mozgás hatékonysága kulcsfontosságú tényezők.

1. Bevezetés, motiváció

Gyermekkorunk óta mindkettőnket érdekelt a robotika és a LEGO robotikai elemkészlete, melynek eredményeképp már régebben beszereztünk egy-egy Mindstorms EV3 robotot, amellyel rengeteg izgalmas projektet terveztünk. Vizsgálatainkhoz a Boston Dynamics Spot robotkutya modelljét vettük alapul, mivel ez a robotkutya mozgásának és funkcióinak lenyűgöző realizmusával inspirált minket. A projekt keretében egy kutya mozgását LEGO Mindstorms EV3 robottal rekonstruáltuk. A modell teljes mértékben LEGO alkatrészekből épült fel, ami lehetővé tette a kreatív tervezést és a könnyű variálhatóságot. A programozáshoz a Visual Studio Code és a LEGO Mindstorms EV3 Home szoftvereket használtuk, melyekben Python nyelven, valamint a Mindstorms saját blokkal programozott felületén dolgoztunk. A projekt során külön figyelmet fordítottunk a robot mozgásának szimulálására, így megismerhettük a kinematika alapelveit és a szenzorok működését. Mindezek mellett fontosnak tartottuk a projekt dokumentálását is, folyamatosan nyomon követve a fejlődést és a kihívásokat, amelyekkel szembesültünk. A jövőbeli tanulmányainkat mi is a Műegyetem falai között szeretnénk tölteni, ahol még mélyebb tudást szerezhethetünk a mérnöki tudományok terén. Szabadidős tevékenységként is ezzel foglalkozunk, hiszen a LEGO és a robotika nemcsak a tanulásunkat segíti, hanem a kreativitásunkat is ösztönzi. Célunk, hogy a robot mozgatása, programozása során a mozgás szekvenciájának haladásra gyakorolt hatását vizsgálva fizikai és informatikai tudásunkat gyarapítsuk, ezen felül pedig megtanuljunk csapatban dolgozni, ami egyetemi éveink alatt is elengedhetetlen lesz. Külön öröm számunkra, hogy munkánk eredményeit a BME Tudományos Diákköri Konferenciáján is bemutatathatjuk, ahol nemcsak a tudásunkat, hanem a csapatmunkánkat is megmérettetjük.

2. Irodalomkutatás

A Boston Dynamics Spot nevezetű robot kutyáját szeretnénk volna valamilyen formában legóból elkészíteni és azt különbözőképpen programozni. Ehhez természetesen több internetes forrást is használtunk, melyekhez a link a „Források” bekezdésben található. Ihletet merítettünk mások elképzeléseiből, megvalósításaiból, és azok segítségével próbáltuk megalkotni saját modellünket. A „LEGO” által kiadott összeépítési útmutatók, illetve videók között részletes leírás nem található erről a Spot robotról. Robotunk építéséhez saját fejlesztéseket hajtottunk végre és a robot programját pedig teljes mértékben mi írtuk; az egész munka saját fejlesztésünk eredménye. A felmerülő programozási, mechanikai, illetve technikai problémákkal mi küzdöttünk meg.

Nyomtatott forrásokra, irodalomkutatásra a négylábú állat anatómiájának - jelen esetben kutya – elemzéséhez, megfigyeléséhez, továbbá a fizikai törvények pontos értelmezése végett volt szükség.

A fontosabb fizikai törvényekről, főként a súlypontról [3], illetve az egyensúlyról [2], erőkről, lendületről és a lendületmegmaradásról [1] fizika tankönyvekből gyűjtöttünk információt.

3. A Mindstorms EV3 eszközkészlete

Bevezetés

A munkánk során használt készlet a LEGO Mindstorms EV3 a LEGO által kifejlesztett robotikai készlet harmadik generációja, amely jelentős előrelépést jelent a korábbi verziókhoz képest. Az EV3 az „Evolution” (fejlődés) rövidítése, amely arra utal, hogy a rendszer az előző generációkhoz képest számos újítást és fejlesztést tartalmaz. Az EV3 hivatalos bejelentése 2013. január 4-én történt, és a készlet 2013. szeptember 1-jén vált elérhetővé a piac számára. Az új kiadás célja, hogy a robotikai oktatás és szórakoztatás világában egyaránt innovatív megoldásokat kínáljon.

A vezérlő felépítése, programozása

Az EV3 legnagyobb újítása a robot "agyát" képező Intelligent Brick [4], amely az előző modellekhez képest jelentősen megnövelt számítási teljesítményt kínál. A robot központi egysége most egy erősebb ARM9 alapú processzorral rendelkezik, amely lehetővé teszi a gyorsabb adatfeldolgozást és bonyolultabb algoritmusok futtatását. Ezen kívül a fejlesztők egy USB csatlakozót és egy SD kártya foglalatot is integráltak a rendszerbe, amelyek a bővíthetőség és a programok tárolásának kényelmét biztosítják.

A programozás terén az EV3 számos lehetőséget kínál. Az alapértelmezett programozási környezet a LEGO Mindstorms EV3 Home Edition, amely a LabVIEW grafikus programozási nyelvet használja. A felhasználók blokkprogramozással irányíthatják a robotot, így könnyen és intuitívan készíthetnek bonyolult robotikaprogramokat. Az EV3 azonban nem csupán a grafikus programozásra korlátozódik: a rendszer támogatja a MicroPython programozási nyelvet is, amely lehetőséget biztosít a fejlettebb felhasználók számára a kód közvetlen írására és futtatására.

További fejlesztés, hogy az EV3 rendszer képes futtatni alternatív operációs rendszereket is, mint például az *ev3dev*, amely egy Debian alapú Linux rendszer. Ezáltal a felhasználók teljes mértékben testre szabhatják a robot működését, és bővebb szoftveres lehetőségekhez férhetnek hozzá.

A készlet tartalma és kiadási verziók

A LEGO Mindstorms EV3 két fő verzióban érhető el: a Home Edition és az Education Edition. Mindkét kiadás tartalmazza az alapvető hardver elemeket, mint az EV3 Intelligent Brick, két nagy teljesítményű motor, egy közepes motor, egy érintés szenzor és egy színérzékelő szenzor. Ezen kívül mindkét verzióban megtalálhatóak a szükséges alkatrészek a robot összeállításához és működtetéséhez. Azonban a két verzió között jelentős különbségek is vannak, amelyek a célcsoport igényeihez igazodnak.

A Home Edition a háztartási és hobbi felhasználók számára készült, és olyan kiegészítőket tartalmaz, mint egy infravörös szenzor és egy távirányító, amely lehetővé teszi a robot távoli irányítását. A távirányító segítségével a felhasználók egyszerűbb vezérlési feladatokat hajthatnak végre, például a robot mozgását vagy egyéb alapvető parancsok végrehajtását.

Az Education Edition viszont az oktatási intézmények számára készült, és olyan szenzorokat tartalmaz, mint egy ultrahangos szenzor, egy giroszkópos szenzor, valamint egy lítium-ion akkumulátor, amely hosszabb működési időt biztosít a robot számára. Az ultrahangos szenzor lehetővé teszi a robot számára, hogy érzékelje a környezetét, és reagáljon a különböző akadályokra, míg a giroszkópos szenzor a robot orientációját méri, így lehetővé téve a precízebb navigációt és stabilitást.

Programozási lehetőségek és oktatási verzió

A LEGO Mindstorms EV3 különösen nagy hangsúlyt fektet az oktatási alkalmazásokra. Az EV3 Home Edition és Education Edition egyaránt lehetőséget biztosít arra, hogy a felhasználók megismerkedhessenek a programozás alapjaival, valamint a robotika és a mérnöki tudományok világával. Az EV3 rendszer lehetőséget ad a felhasználók számára arra, hogy különböző algoritmusokat, szenzorokat és motorokat használjanak a robot működtetésére, miközben fejlesztik problémamegoldó képességeiket és kreativitásukat.

A rendszer programozása kezdetben a grafikus blokkprogramozáson alapul, amely könnyen elsajátítható, de a későbbi szoftveres támogatás (pl. MicroPython és ev3dev) lehetővé teszi a fejlettebb alkalmazások és bonyolultabb programozási technikák használatát. Ennek köszönhetően az EV3 a kezdő és haladó felhasználók számára egyaránt megfelelő választás.

4. A robot mozgásának logikájának bemutatása, a kutya felépítése

Ebben a fejezetben mutatjuk be a robot teljes felépítését, funkcióit, működését, ezek mellett még kitérünk egyes mechanikai problémákra, amelyek felléptek a modell megépítésekor.

4.1 Rendelkezésünkre álló eszközök

A felhasznált Mindstorms szettből a kész robot négy motort tartalmaz, melyekből kettő vesz részt a lábak mozgásában, egy a csípőt és végül a negyedik a fejet mozgatja. Ezen felül még a fejrészben egy infravörös szenzor található.

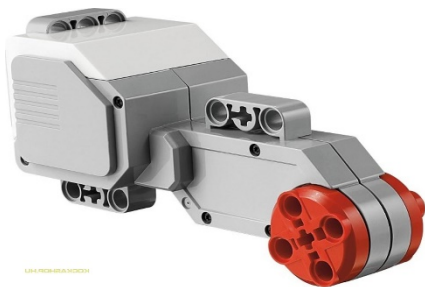


4.1.1 Infrared sensor

Az infravörös szenzor (1. ábra) adatokat mérhet Proximity, Beacon és Remote módban, és numerikus kimenetet kaphat. Összehasonlíthatja az érzékelő adatait egy bemeneti értékkel és logikai (igaz vagy hamis) kimenetet kaphat.

A robot felépítéséhez számos lego technic elemet használtunk fel.

Két fajta motort használtunk: az erősebbeket a lábak esetén („large motor”-okat) és kisebb teljesítményű „medium motorokat” a csípőnél, illetve fejrészénél használtuk.



4.1.2 Large motor



4.1.3 Medium motor

A „Large motor” (4.1.2 ábra) be- és kikapcsolható motor, szabályozható teljesítményszinttel, vagy bekapcsolhatja a motort meghatározott ideig vagy fordulatszámon.

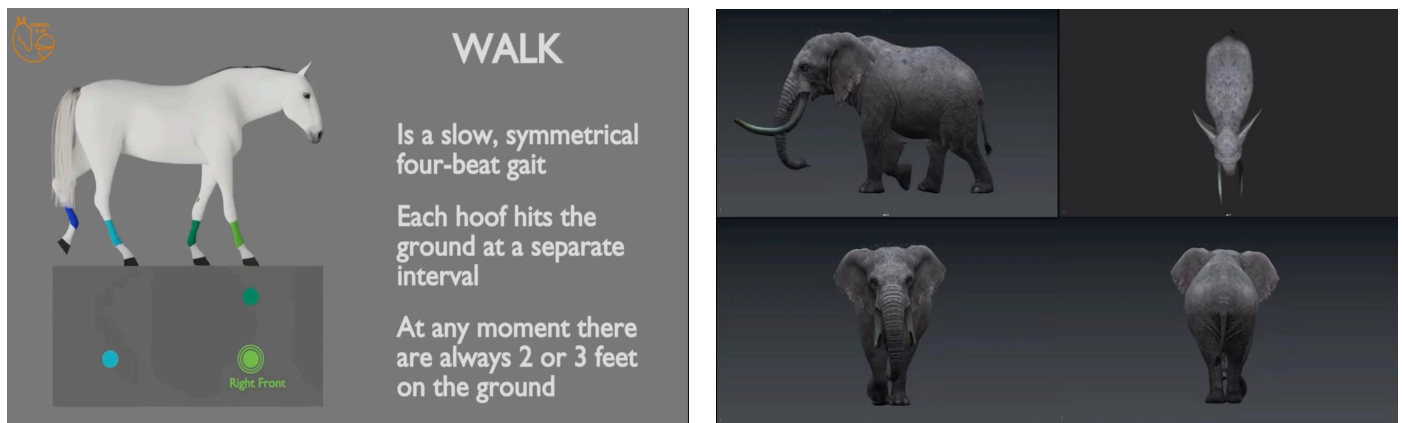
A „medium motor” (4.1.3 ábra) be- és kikapcsolható motor, szabályozható a teljesítményszint vagy bekapcsolhatja a motort meghatározott időre vagy fordulatszáma

4.2 Négylábú mozgás kutyamódra (elképzelés)

Miért pont a kutyákra jellemző sétáló mozgás? Ez egy nagyon lényeges kérdés. A futást teljesen kizártuk modellünkénél. A futás minden állatnál egy bonyolultabb szimulációt követelne és jóval erősebb szerkezetet, jelen helyzetben hosszabb munkát igényelne.

A sétálás elemzéséhez a következő állatok járását vizsgáltuk.

Az elefánt esetében a sétálása közbeni testsúly áthelyezését úgy teszi lehetővé, hogy nagy

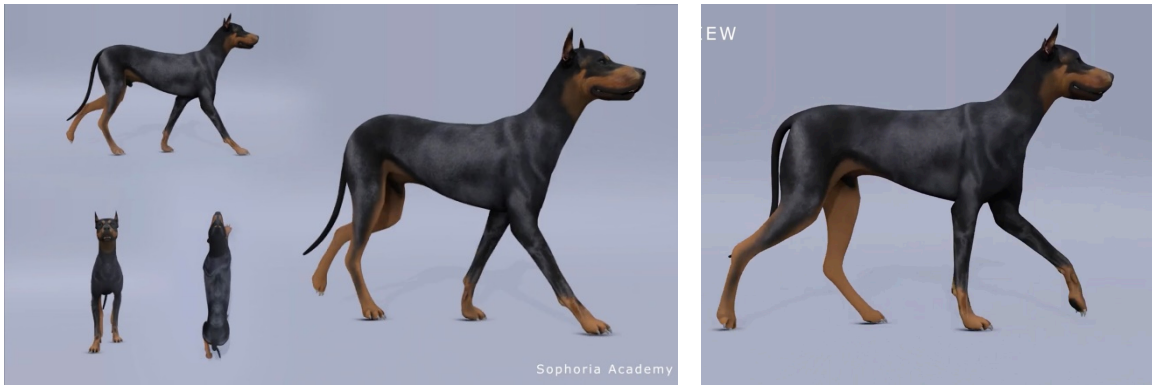


4.2.1 Ló és az elefánt járása

testtömegét, áttesz egyik oldaláról a másikra és két jobboldali lábát szinte egyszerre viszi előre, de mégis van az a pillanat amikor a 3 lába egyszerre van a földön. A ló esetében mindig 2 vagy 3 láb van egyszerre a földön.

Számunkra inspirációt alapvetően a Boston Dynamics „Spot” robotkutyája nyerte el és a kezdeti terveink is annak a robotnak a LEGO-beli modellezése felé vezettek minket. Továbbá kulcsfontosságú, hogy a kutya gyaloglása esetén egyszerre 3 láb van lent, és ez adja a járás stabilitását. Az ábrán látható módon az egyik oldalon lent van két láb, testének arra féltekéjére helyezi a testsúlyát. Ennek elemzésére még a súlypontáthelyezésnél és a járásnál ki fogunk térni. Ezen a két képen két egymást követő fázisa látható a kutya sétájának. A kutya mozgása is nagyon hasonlít az elefántéhoz. Féltelmes testsúlyáthelyezés figyelhető meg.

Másrészt olyan periodikusan ismétlődő mozgást szerettünk volna megvalósítani a négy lábra mindössze két motorral, amely különböző sebességeken stabilan képes járni, minél kevesebb súlypontáthelyezési problémával.



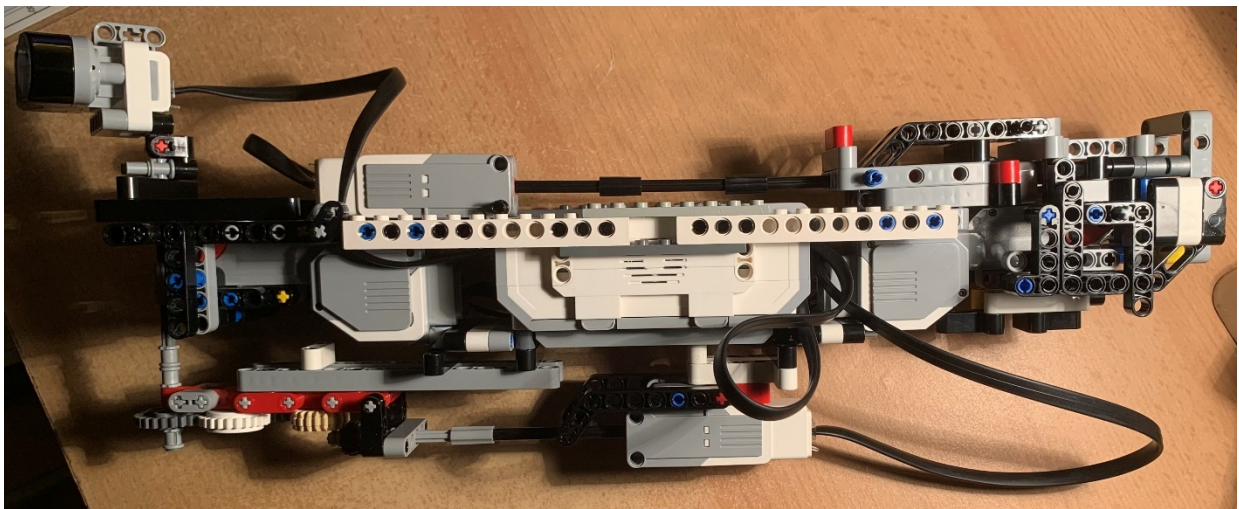
4.2.2 A kutya három lábéríntkezése a talajhoz

4.3 A robot mechanikai felépítése és a négy lábú mozgás megvalósítása

Magát a sétáló mozgást két large motorral kellett megoldjuk. Ami azt eredményezte, hogy a lábakat egymáshoz képest eltolva kellett felfüggeszünk. A legnagyobb tömegű testrészt természetesen a kutya törzse. Az egész robot mindössze 1,164 kg-ot nyom, teljes testhossza lábaival együtt 52 cm, maga a törzs csak 40 cm méretű lett.

Központi rész

Ezen a képen látható a végleges törzs.



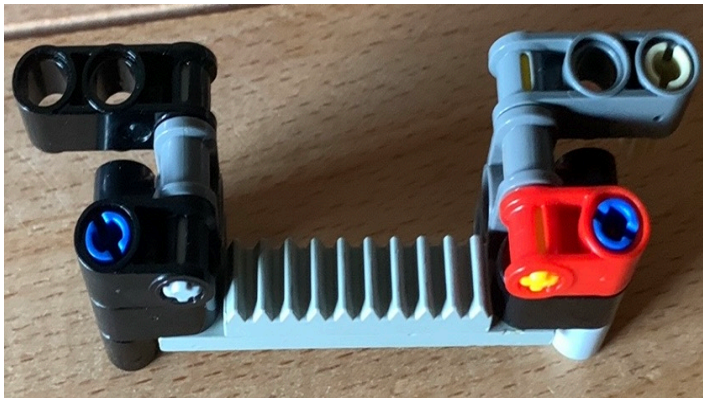
4.3.1 A robot központi része

Ugyan a Lego elemek szabványos méretűek, összeillesztésük precíz, azonban összetett funkciók kialakítása esetén olyan problémák merülhetnek fel, amikor az egyes elemek elhelyezési igénye és elhelyezési lehetősége nem egyezik, és ennek következtében nem pontosan úgy

illeszkednek, ahogy azt az alkotó kívánja. Ilyen technikai problémák fel szoktak lépni minden ilyen projektnél. A képen középen látható a Mindstorms szíve, a BRICK, ami egyfajta számítógépként funkcionál, ehhez kapcsolódik a két large motor, melyek a lábakért felelnek,

Csípő és rotáció

Az irányváloztatásra egy „váltót” építettünk, amely képes irányt váltani a láb mozgatásával párhuzamban. Ezt egy fogaskerék, egy fogasléc és két csukló segítségével valósítottuk meg.



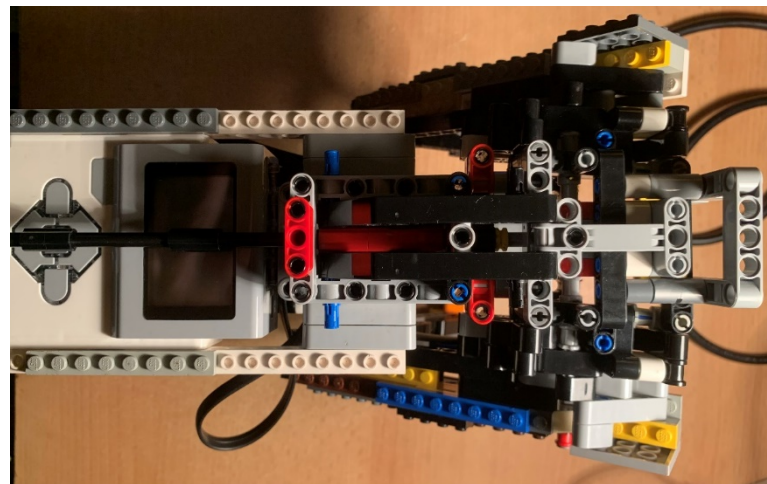
4.3.2 Házi készítésű fogasléc



4.3.3 Csukló alkatrész

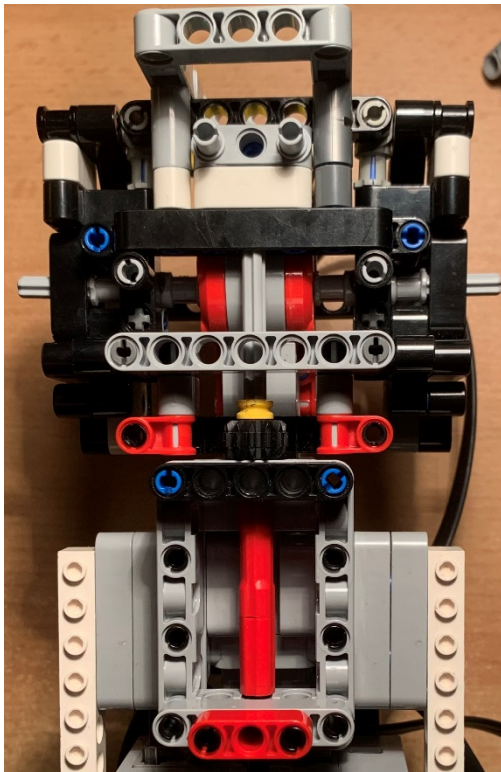
Az otthon készített fogasléc sajnos nem vált be az anyagok szabványossága miatt, ugyanis a fent látható képen az alkatrész párosan jön ki. A legonál viszont az anyagok páratlanra vannak szabva, hogy ne forduljon elő asszimetria. Emiatt ezt sajnos alkalmazni nem tudtuk és muszáj volt egy gyárit beszerezni.

A csípőt a fenti medium motor segítségével szabályoztuk.



4.3.4 Végleges csípő

A csípő működése: a fenti medium motor összeillesztett rudak segítségével hajt meg egy fogaskereket. A fogaskerék meghajt egy fogaslécet, ami így jobbra-balra mozog. Fontos, hogy legyünk tekintettel arra, amikor a motort vezéreljük, hogy egy adott fordulatszámot sem pozitív, sem negatív irányban se lépjünk át. Ha ezt megtesszük, a fogaskerék el fogja érni a lécszélét és ez akár a motor károsodását is okozhatja. A 7-es méretű fogasléc esetén a 0.2-es fordulatszámot optimálisnak találtuk.



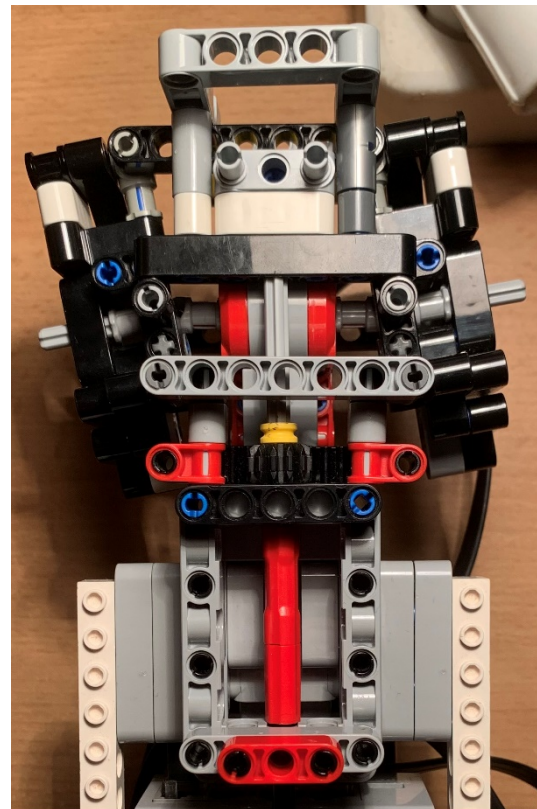
4.3.5 Alaphelyzet

A képen az alaphelyzet látható. A robot csak ebből a kiinduló helyzetből fog egyenesen menni.

A képen látható belülről a mechanizmus. A váltó alatt a large motor helyezkedik el, amiből két rúd ível ki. Azokhoz egy-egy csukló illeszkedik, amelyekből egy-egy másik ugyanolyan méretű rúd jön ki, és végükre illeszkednek a lábak. Előfordultak technikai nehézségek az építés során. Például úgy kellett fixálni magát a vázat alulról és felülről egyaránt, hogy azok ugyanúgy mozdíthatók legyenek. Ez ahhoz a problémához vezetett, hogy a fogaskerék mozdítása nehézkes lett, melyet speciális anyagokkal küszöböltük ki.

A második képen jobbra a rotáció látható. A kanyarodást legszebben hátrafelé való mozgással lehet elvégezni, melynek menete a következő. Tegyük fel, hogy a robotunk álló helyzetből, tőle jobbra szeretne elmenni; az első két lába fix, nem képes irányváltásra. Ezt a problémát azért nem lehet megoldani előrefele haladás közbeni rotációval, mert a kutya törzse egybefüggő (nincsen gerince, amivel ezt lehetne szabályozni) és a test elején sincsen irányt váltó mechanizmus.

Azt tapasztaltuk, hogy hátrafele mozgásnál tudunk irányt változtatni, mivel maga a csípő is hátul van, nincs maga előtt semmi, amit tolnia kellene. Ezt a szöveget kellő ideig tartva előbb utóbb egy kört fog tenni a kutya. A tolatás során a csípő felőli rész húzza a teste elülső részét.

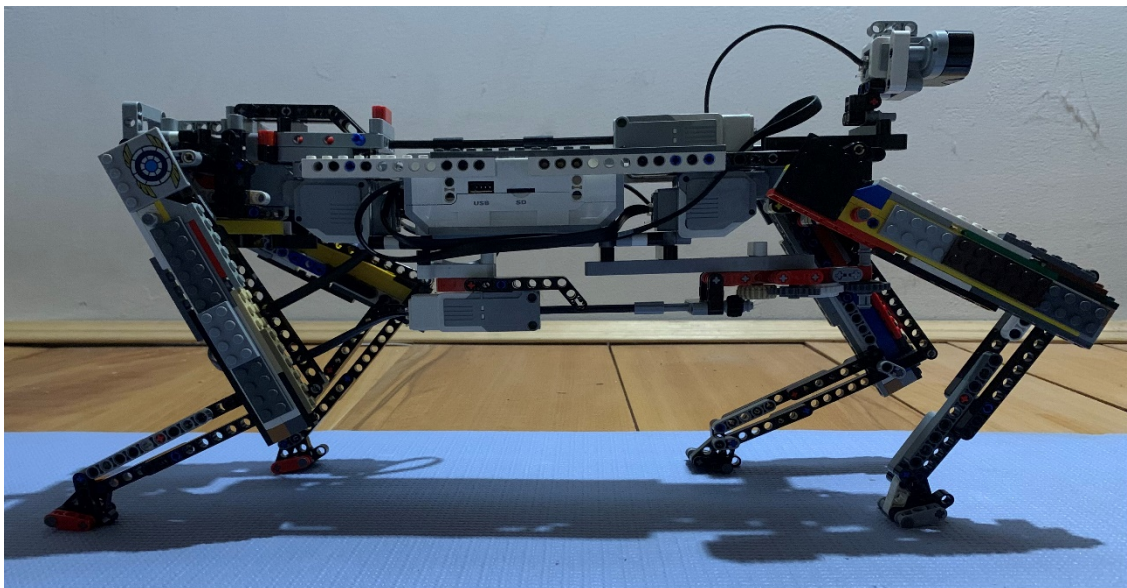


4.3.6 Rotáció

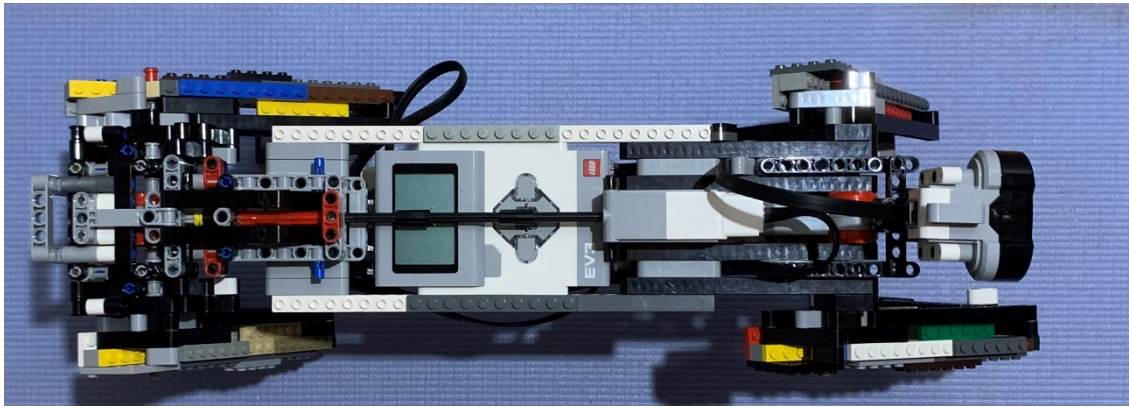
Lábak és azok kapcsolata a motorhoz

A robot mozgásának a legbefolyásolóbb része a lábak kapcsolata, azok érintkezési tényezői a talajjal. Itt több fizikai tényező is érvénybe lép. Magának a lábnak a kapcsolata a motorhoz, melynek kellően rugalmasnak kell lennie ahhoz, hogy a mozgás minél harmonikusabb legyen. emellett képesnek kell lennie újra és újra visszatérnie ugyanabba a pozícióba.

A lábak állása: hogy mozgást tudjunk előidézni négy lábbal, úgy kell elhelyezzük a lábakat, hogy mindkét pár láb közül az egyik hátrébb álljon, mint a másik és azok pont ellentétesen álljanak a másik pár lábbal (JE -BH; BE-JH). Vagyis, ha a hátsó pár lábai közül a jobboldali előrébb van, mint a bal, akkor az első pár lábainál pont a fordítottja lesz, a bal lesz előrébb. Ennek azért kell így lennie, mert ha elindítjuk a motort egy bizonyos fordulatszámom, akkor az egyszerre két lábat hajt meg és az előrébb lévő lábáról előre tolja magát és közben előre jön az addig hátrébb levő láb. Abban az esetben, ha azonos lábak lennének elől, egyszerre akarná előre vinni a kettő azonos hátsót, ami azt eredményezné, hogy felborul, mert nincsen, ami megtartaná a súlyt azon az oldalon ahonnan a lábakat elmozdítjuk. Ha mindegyik láb azonos pozícióban lenne, akkor nem lenne előrehaladás. A súlypontáthelyezésnél lesz erről bővebben szó.



4.3.7 A járáshoz alkalmas lábelhelyezés



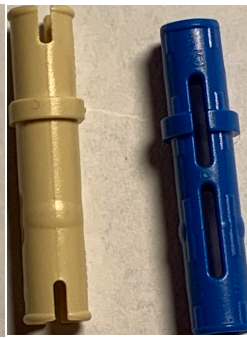
4.3.8 A járáshoz alkalmas lábelhelyezés, felülről

A lábak felfüggesztése a motorhoz

A lábak a képen látható kettes csatlakozókba egy világosbarna duplapinnel kapcsolódnak. A csatlakozóból egy X tengely vezet a motorhoz, amely fix. Ezáltal a motor által a csatlakozó egy teljes köríven fog végig haladni.



4.3.9 csatlakozó



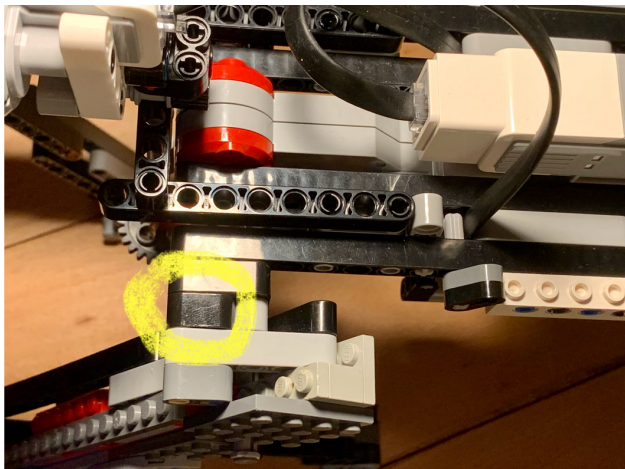
4.3.10 pinek

A világosbarna speciális dupla pin gyors és könnyed elmozdulást eredményez a két vagy 3 általa összefogott anyagra, mivel képes elfordulni a lyukban. A képen látható, hogy a kék pinnel a világosbarna sokkal simábban van megtervezve pont azért, hogy

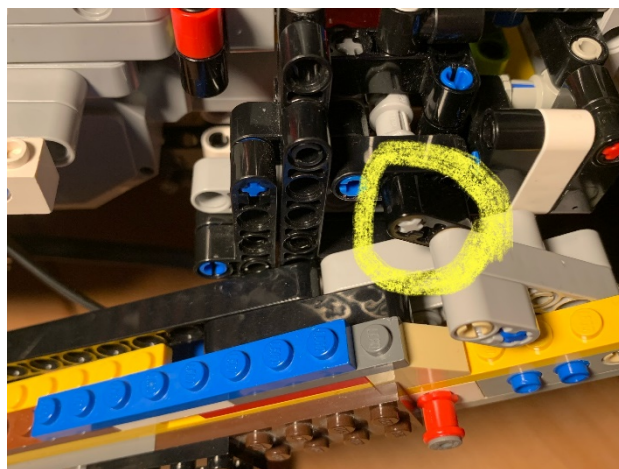
gyorsítson egyes folyamatokat. Ezek az anyagok elengedhetetlenek a felfüggesztésnél.

Fontos a szög, amelyben a motor indulása előtt a csatlakozó áll. A mi kutyánk két féle járást képes elvégezni: azt mikor két ellentétes láb van lenn a földön - csak kétlábás érintkezés, a másik pedig, mikor eltolva indulnak el a lábak: egy elemelkedik egy pillanatig, míg a másik három a földön van.

A kétlábás megoldásnál kezdő pozícióban az összes csatlakozó vízszintesen egymáshoz képest ugyanúgy áll, 180°-ban. Ha a két large motort ebben a helyzetben ugyanakkora teljesítményen hajtjuk meg, akkor a két átellenes lábról fogja előre és áttolni a magát a másik kettőre.



4.3.11 első két láb csatlakozója

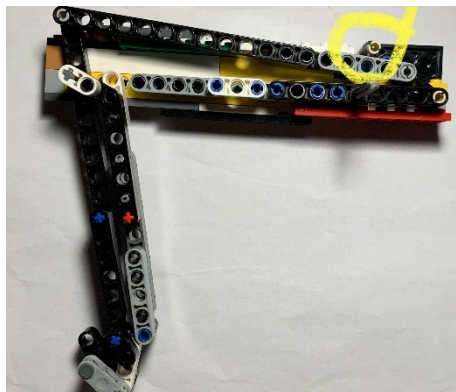
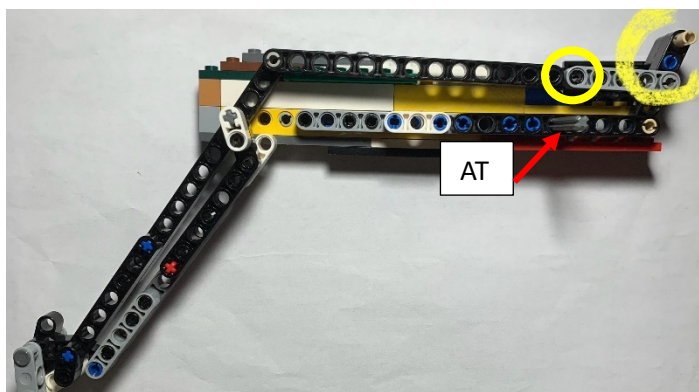


4.3.12 csípőcsatlakozás

A stabilabb megoldásnál el kell tolnuk a csípőhöz rögzített csatlakozókat 45°-kal lefelé, az elülsők 180°-on maradnak. Így toltuk el a lábak leérkezését, lépéssorozatát. A motor egy adott frekvencián forog körbe-körbe ezzel megfordítva az X tengelyeket, amik fixek és azok megfordítják a csatlakozókat, amik más szögben álnak egymástól. A csatlakozók valójában egy 360°-os kört tesznek meg, de nem ugyanaz a kiindulási helyzetük.

A képen az elfordítások lathatók. (4.3.11, 4.3.12)

A lábak összetétele: A lábak változtatható állapotúak ki és becsukhatóak. A lábak állási szöge a csatlakozó helyzeti állásától fog függni. A felfüggesztéshez csatlakozik egy 5 szegmenses lécs, amely vagy tolja, vagy húzza a rákötött hosszú rudat. Ezen rúd több kisebb rúdból épül fel - ugyanis kell egy akadály, hogy a felső hosszú lécs ne menjen túlságosan ki.



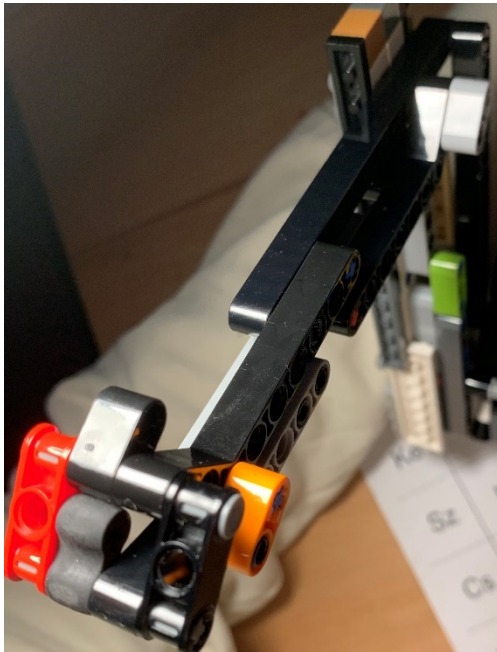
4.3.13 Lábak szerkezete

A lábakat két tengely tartja a testhez: az alsó tengely (AT), mely csak arra szolgál, hogy megtartsa a testet és hogy magát a lábat odaszorítsa a testhez. A felső tengely (FT) a láb mozgatására szolgál.

A térd csatlakozási pontján egy szimpla könnyedén mozdítható pin helyezkedik el. A lábszár fixen tartja a talpat. Több talpfajtát is készítettünk, ugyanis különböző felszínre különböző tapadású, méretű és felületű lábfej kell.

Talpak, lábfejek:

A robotot főként hajópadlón és fa parkettán teszteltük. A hajópadló, amin teszteltünk hézagos és nem azonos szintű falapokból állt, emellett még durva lejtéssel is rendelkezett. Ha nincs túl nagy szintkülönbség, akkor a robot a következő lábfejjel gond nélkül képes haladni.



4.3.14 Fix lábfej

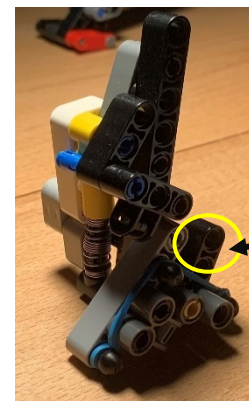
A képen egy merev lábfej látható, amely teljesen alkalmas a vízszintes/enyhén domború talajon való járásra. A lábfej merevségét a lábhoz rögzített „L” alakú lécszár adja. A faalapú járófelülethez elegendő tapadási anyagot tartalmaz, amely ebben az esetben a talpban található gumis LEGO csatlakozó. Az „L” lécszár talphoz való csatlakozásánál közvetlen balra mellette található egy elferdített kettős csatlakozó. Ez arra szolgál, hogy a robot lába mozdtatása elején míg fel nem emeli a lábát, legyen egy olyan felület, ami könnyen csúsztható (ezért nem gumi) és egyben tartást is ad.

Ezen kívül próbáltunk leutánozni egy valódi kutyalábat, ami valószínűleg a következő fejlesztésben fog bekerülni a mechanizmusba.



4.3.15 kutya lábfej

Megfigyeltük, hogy amint a kutya lába a földet éri és ráhelyezi testsúlyát, besüllyed és a lábujjai tágra nyúlnak. A mi elképzelésünkben ezt a süllyedést egy rugós alkatrészrel próbáltuk rekonstruálni, hogy amint a súly

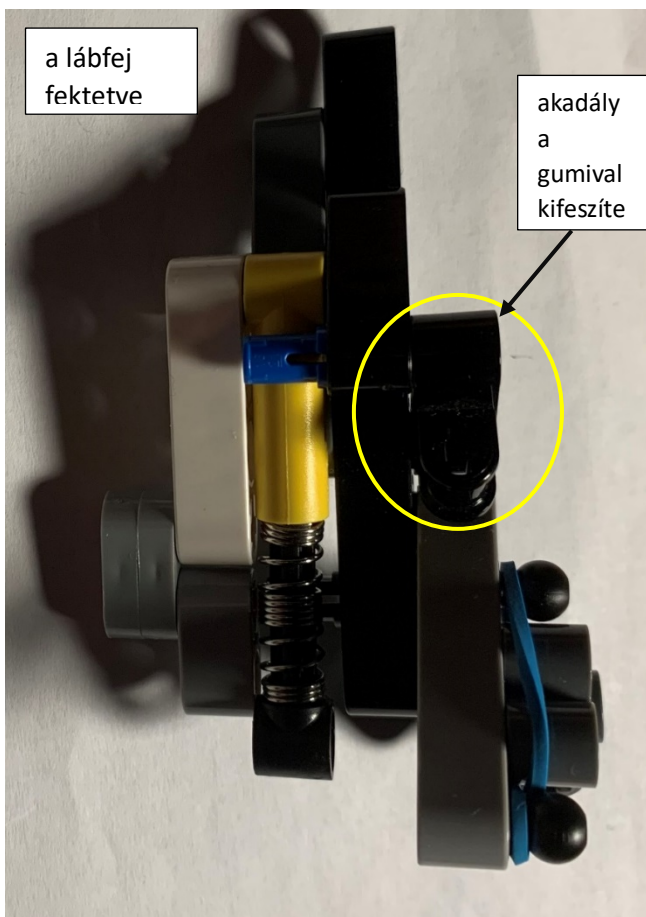


„T” lécs egyben akadály is

4.3.16 kutyalábfej modell

ránehezedik az nyomódjon le és amint eltevéődik a súly húzódjon össze.

A lábujjak kinyílásáért és összehúzódásáért egy gumival összekötött egy 5x1-es lécs és egy T lécs felel melyek a nyomás hatására kinyílnak.



a lábfej fektetve

akadály a gumival kifeszíte

Ahhoz, hogy a robot ne dőljön el semelyik irányba sem, a lábfejen belül is különböző alátámasztásokat kell létrehozunk. Itt ezeket akadályoknak hívom. Ezek arra szolgálnak, hogy ellentartsanak a rájuk nehezedő nyomásnak ezzel korlátozva a lábujjak kinyílásának nagyságát.

A fejrész bemutatása

A mozgatható fej nagyban módosította a működtetés eszköztárát. A mozgatható fejjel nem csak az előtte lévő tárgyakat látja a robot, hanem az őt körülvevő dolgokat is. A fejet egy medium motor vezérlésével tudjuk mozgatni.

4.3.17 Kutya technic lábfej

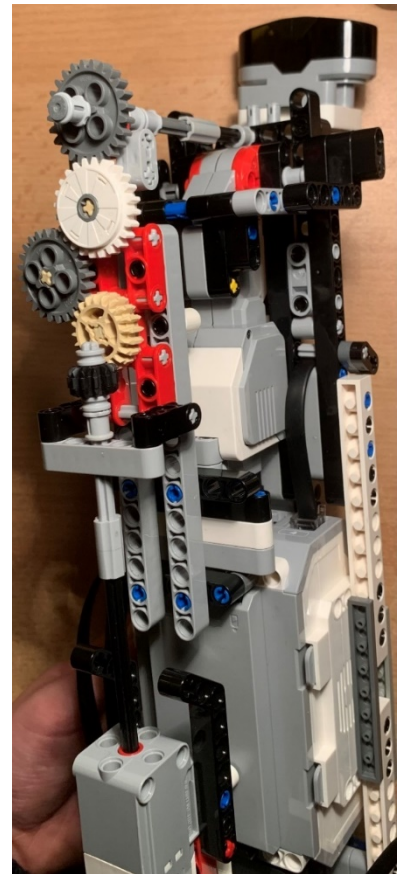
Első típus:

A motor meghajt egy X tengelyt, amire egy kis fogaskerék van rá rögzítve. A kis fogaskerék láncban meghajt nála négy, majdnem kétszer nagyobb fogaskereket. A lánc végén lévő utolsó fogaskerékből felfele kiível egy X tengely, amely a forgást továbbítja a fejnek.

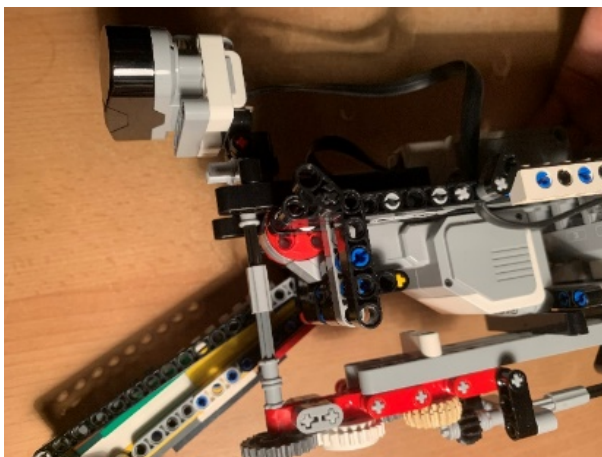
A fogaskerek nem teljesen ugyanakkorák. A merőlegesen való forgatást csak egy arra kifejlesztett fogaskerékkal lehetséges megvalósítani. A képeken ezek a fekete és a világosbarna fogaskerek.

Probléma:

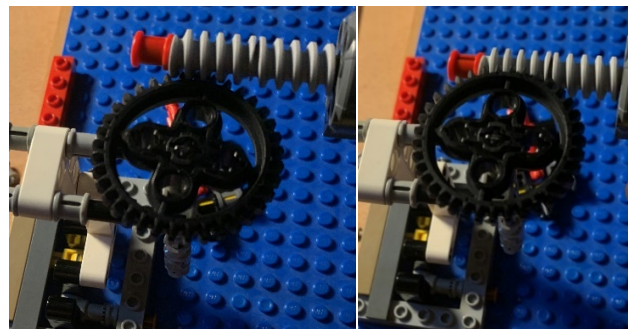
A fej teljes mértékben forgatható. De sajnos a hiba fejbe illeszkedő vezetékkel van, amely ellent tart a fej mozgásának. A fej balra való mozgatásánál azt tapasztaltuk, hogy a túl nagy ellenerő miatt egyes fogaskerek kiakadtak. Ugyanis a kábel balról csavarodva van bekötve a fejbe.



4.3.18 fogaskerekes mechanizmus



4.3.19 fogaskerekes mechanizmus oldalról



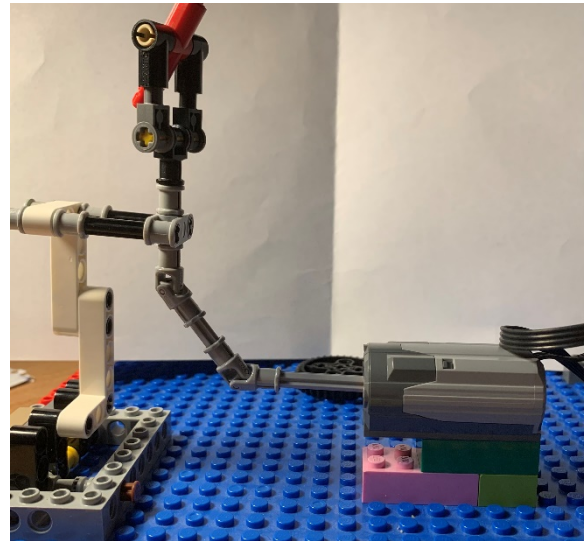
4.3.20 csigás forgatás

Megoldás: valamilyen módon a kábelt felhozni a robot tetejére és nem valamelyik oldaláról kivezetni vagy csigás megoldást használni. A csiga fogaskerék csatlakoztatása esetén: ha a fogaskerék kezdeti állapota fent helyezkedik el, akkor a csiga fent tartja a fogaskereket (fontos: csak többélű fogaskerékkal működik). A csiga előnye, hogy nem lehet kizökkenteni a menetéből, többfeleképpen lehet elhelyezni-ezzel szemben igen lassú.

Második típus:

A fejet nem csak fogaskerekes úton lehet meghajtani, lehet tengelyes úton is. A motorra merőleges irányba kell hozni a tengelyt. Az egyensúly megtartása és a súlypont minél lentebb tartása érdekében a mechanizmus által keltett forgást fel kell hozni. Mi ezt kétféleképpen is megcsináltuk.

A csuklók lehetővé teszik a mozgás irányának megváltoztatását több irányba is.



4.3.21 fix csuklókkal

Harmadik típus:

A mozgása körülbelül ugyanúgy néz ki, mint a második típusé csak oldalra nem tud olyan nagy mértékben elfordulni. A csuklók helyett hajlítható gumi axle van (X tengely) benne. Erősebb is mint a csuklós verzió.



4.3.22 gumis tengelyes megoldás

Mindkét esetben csak a látvány kedvéért szereltem fel a mechanizmus tetejére plusz dolgot. Olyan volt, mintha propeller lett volna az egyenletes-körmozgás miatt.

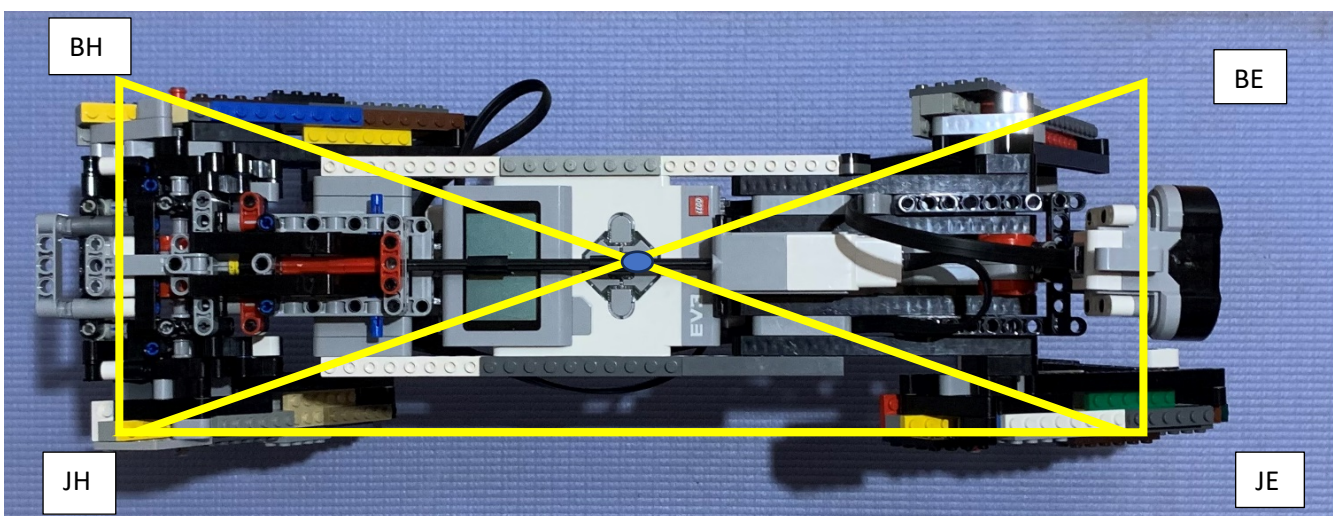
4.4 Mozgásmechanika és a fizikai problémák kiküszöbölése

Ebben a fejezetben a robot fejlesztésének szakaszait, a kezdetektől a modell végleges felépítéséig felmerülő problémáit fogjuk bemutatni.

A robottal idén szeptemberben kezdtünk el komolyabban foglalkozni.

Súlypontáthelyezés

A megfigyelt négylábú állatok, legalábbis a kutya és az elefánt esetében a lépés a testsúlyuknak féltekés áthelyezéssel történik. A robotunk nem rendelkezik gerinccel, mellyel könnyedén tudná testsúlyát egyik testféltekéről a másikra áttenni. A mi robotunk fix központtal rendelkezik. A robot súlypontja szinte a robot felezővonalának a közepén van. Eleinte ez egy kicsit hátrébb helyezkedett el, de mivel a fejrészt rászerteltük, ezáltal magát a súlypontot nem csak a függőleges tengelyen mozdítottuk el(lefelé), hanem kicsit vízszintesen is a fogaskerékes rész miatt. A robot súlypontját álló helyzetben az ellentétes lábakra húzott két egyenes metszéspontjából kapjuk meg. A robot lépése közben, mikor az egyik lábát felemeli, a súlypontja elmozdul és az egyik átló megszűnik. A talajon lévő három láb derékszögű háromszöget alkot. A robot akkor fogja megtartani az egyensúlyát, ha a súlypontja a háromszögön belülre esik. A mi esetünkben a haladásnál mindig az előrébb lévő láb adja az alátámasztást, hiszen arról is fogja magát tovább tolni. Az elől lévő láb nélkül felborulna. Ellenpáras járás esetében megszűnik a háromszög; helyette csak az átellenes lábakat összekötő egyenesek maradnak. Az egyensúly akkor fog megmaradni, ha a súlypont ráesik az egyenesre. Megfigyelés: megfigyeltük, hogy különböző mozgásforma célszerű különböző járófelületeken. Amennyiben a járófelület tapadási tényezője meghalad egy bizonyos értéket, abban az esetben



4.3.23 Súlypont elhelyezkedése

ellenpáras járásra van szükség a súrlódási erő csökkentése érdekében. Ez fordítva is igaz: a tapadási tényező csökkenése végett növelni kell a súrlódási erőt, tehát háromlábás mozgásra van szükség. Az irányváltás közben is eltolódik a súlypont, ezen felül a hirtelen nagy lendület is kibillentheti a robotot.

Fejlődés szakaszai

1. Prototípus

A robot ebben a szakaszban még nagyon kezdetleges. A csípő késznek mondható, viszont a lábak még sok módosítást igényeltek. A talpak rendkívül könnyen hajlíthatóak voltak, ezen felül a lábak sem álltak a megfelelő pozícióban. Ez a modell a nyári verzió. Nyáron a Micropython és Pythonnal foglalkoztunk, ugyanis a célok egyike az volt, hogy a Pythonban való programozást is ez által más megvilágításból gyakorolni tudjuk. Színesebbé téve ezzel számunkra a programozás világát.

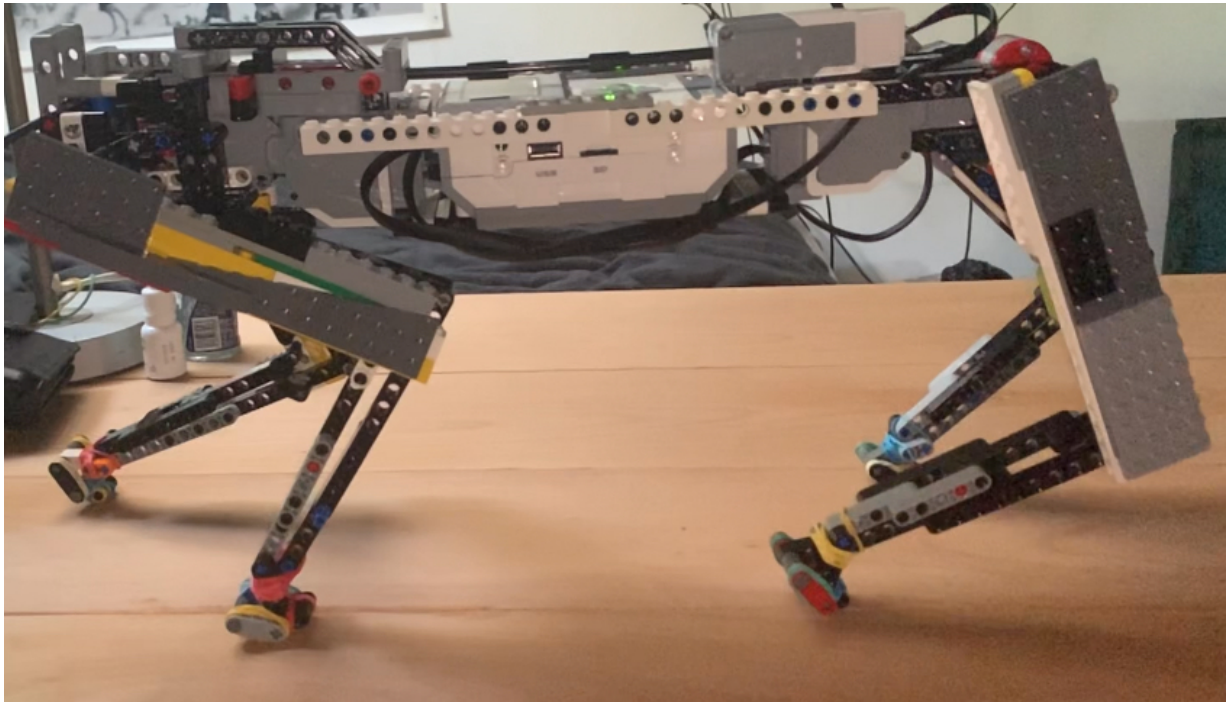


4.3.24 első prototípus

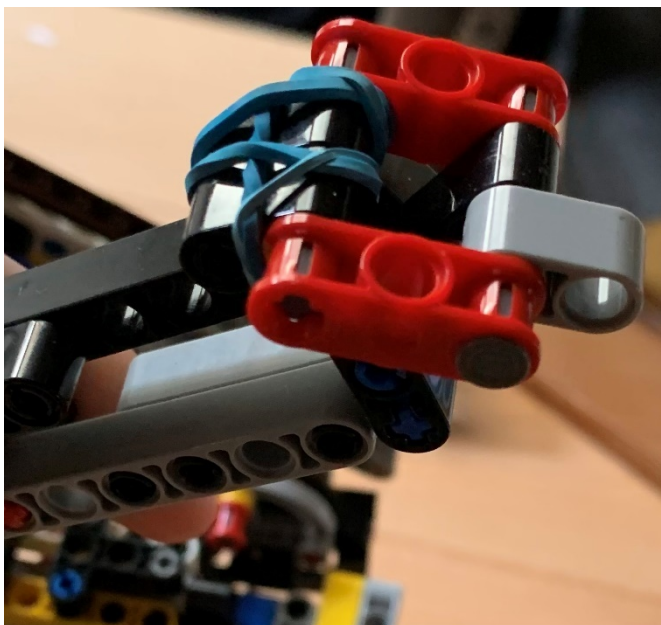
Itt négy-öt nagy probléma lépett fel. A csípő rotációja nem működött, ugyanis a csípőt meghosszabbító csatlakozó megakadt a stabilizáló lécekben. A robot lábai rossz szögben álltnak, túlságosan becsúsztak a test alá. Ez az első pár lábnak a rossz felfüggesztése miatt van. Hátrébb helyezkedik el maga a láb a kelleténél. A lábfejek rendkívül mozdíthatók voltak. Ezt a két problémát csak a következő prototípusnál vettem észre. Ennek a robotnál a célja az volt, hogy el tudja végezni azokat a mozgásokat, amire utasítjuk. A két motor párhuzamos mozgatására a micropython nem volt képes, a Mindstorms EV3 home viszont természetesen igen.

2. Prototípus

A modellt ebben a fázisában a saját blockprogramjában teszteltük, hogy kapjunk egy vázlatot arról, hogy mit szeretnénk megvalósítani Pythonban. Tapasztalat: a robot nem volt képes előre haladni mert az első két lába túlságosan a test alatt volt. Nem tudta eléggé alátámasztani magát sem, emellett a súrlódási erő nagysága miatt nem ért időben a megfelelő helyre.



4.3.25 második prototípus



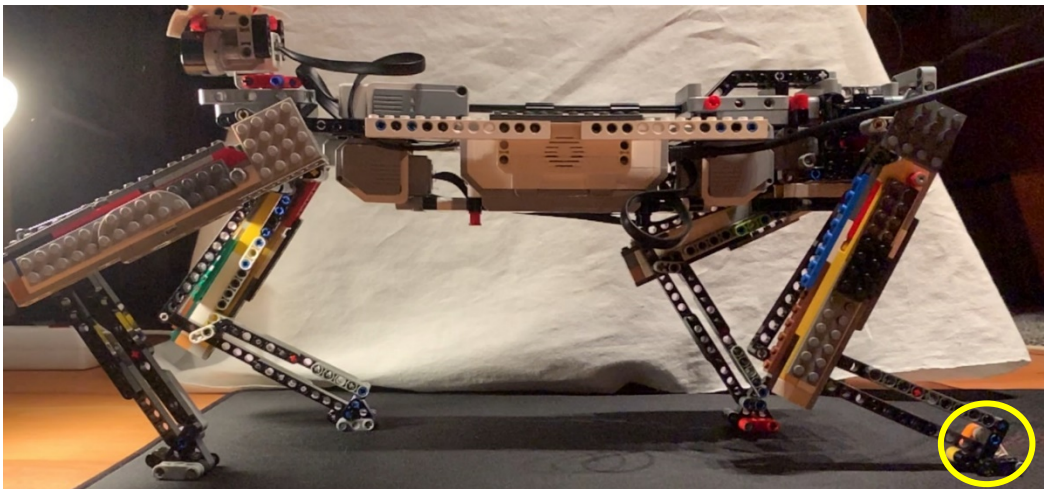
4.3.26 talp prototípus

A lábfejeket befűtött gumikkal próbáltuk meg fixálni annak érdekében, hogy ne veszítse el teljesen a rugalmasságát a lábfej, de az csak növelte a súrlódási erőt, amely visszahúzta a robotot. Lassított felvételeket készítettünk a hibák könnyebb észrevétele érdekében. És ekkor vettük észre, hogy az első két láb alsó felfüggesztése, tengelye eggyel hátrébb van illesztve, mint a hátsó lábnak.

Változtattunk a gumik mennyiségén, de sajnos ez sem állta meg teljesen a helyét ezért a befőttes gumi helyett bevezettünk egy legó gumi csatlakozót. A különbség a helyzetük. A befőttes gumi esetén a gumi közvetlen ért le a földre, ami növelte a súrlódási erőt. Nekünk olyan pontot kellett megkeressünk, ahol pont elegendő tapadást adunk. Ez a talpban van, mert itt a gumival együtt a többi alkatrész is leér.

3. Prototípus

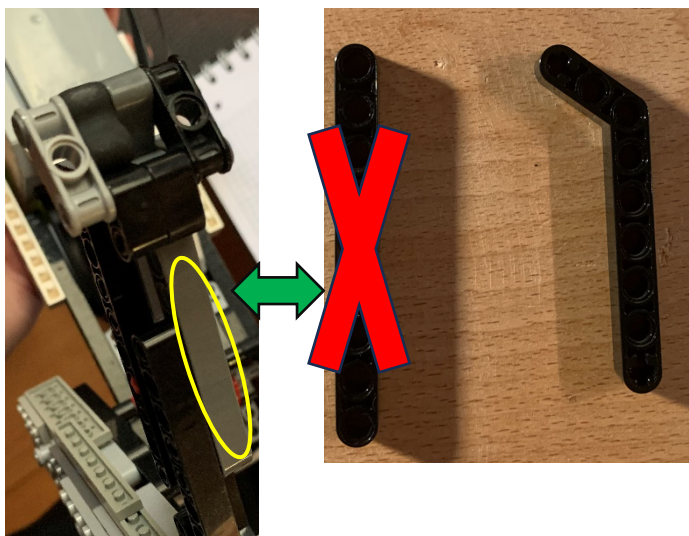
Ebben az állapotban a robot már járóképes, mert megegyeznek az ellentétes lábak állásának szögei. Ennél a modellnél viszont a lábfejeket fixálni kellett, mert a súrlódás miatt mindig



4.3.27 hármas prototípus

elállítódott. Elkezdtünk komplett programsorokat írni, mérni és viszonyítani külső tényezőkkel.

A képen látható, hogy a lábfejek össze vannak csúszva.



4.3.28 L rúd beillesztése a láb merevítése

A gumis anyag egy szintben van a talp többi részével.

Az eltolódást egy L léccel orvosoltuk.

A több programozási lehetőség miatt a fejet mozgathatóvá alakítottuk.

Könnyen elbotlott kisebb hézagokban is, ezenfelül a kissé meredek talaj is gondot okozott.

4. Prototípus

Ez modellünk végső formája. A lábakba szerelt lengéscsillapítókcal egy könnyedebb zökkenőmentesebb járást valósítottunk meg. Ezzel egyúttal a kilengés mértékét is csökkentettük. A 4.3.30 láb alsó részén elhelyezett sarkantyú alátámasztást biztosít emelkedők esetén (például: lejtőn).



4.3.29 Fix lábfej



4.3.30 láb, lengéscsillapítókkal



4.3.31 prototípus

4.5 Mérések

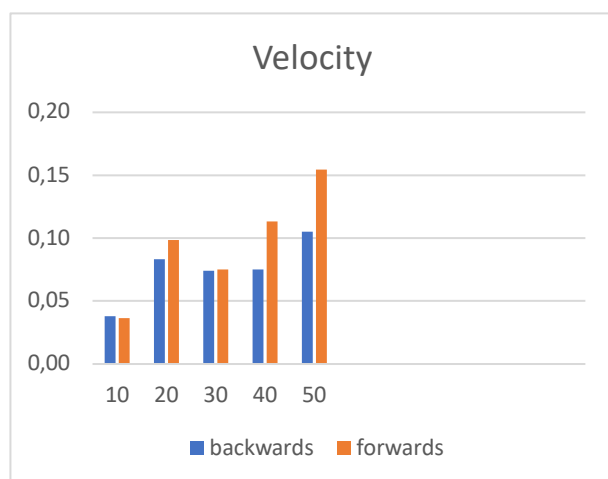
Méréseinket a 3 prototípussal végeztük. Méréseink során a robotot különböző járófelületeken teszteltük (például: hajópadló, szőnyeg, parketta) és ehhez próbáltunk különböző lábfejeket építeni. Az adatokat Excelben rögzítettük.

Érdesebb felületeken csökkentettük a gumis anyagok számát, illetve csúszósabb tereken növeltük azt.

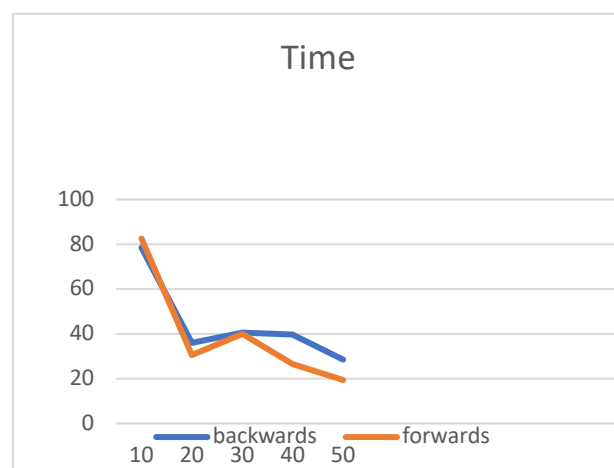
Méréseink során a robotot különböző járó felületeken teszteltük (például: hajópadló, szőnyeg, parketta) és ehhez próbáltunk különböző lábfejeket építeni.

Érdesebb felületeken csökkentettük a gumis anyagok számát, illetve csúszósabb tereken pedig növeltük azt.

power (in Ev3 home)	time	velocity (m/s)	time	velocity (m/s)
	backwards	backwards	forwards	forwards
10	75,62	0,04	76,62	0,04
20	34,49	0,09	36,22	0,08
30	47,8	0,06	23,63	0,13
40	37,89	0,08	19,53	0,15
50	25,55	0,12	17,82	0,17



4.3.32 sebesség függvényében

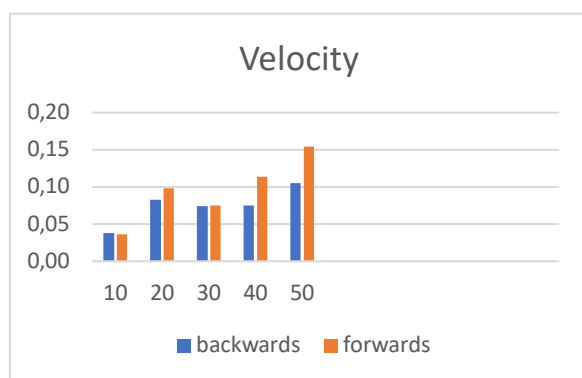


Az alábbi táblázat mutatja a robot parkettán történő ellenpáras járását 3 méteren. Az irányítást itt az EV3 home-ban tettük meg.

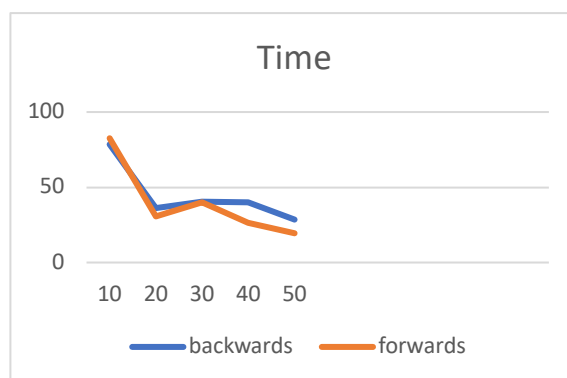
Méréseinket ugyanúgy a robot hárompontos járása esetén is elvégeztük.

power (in Ev3 home)	time	velocity (m/s)	time	velocity (m/s)
	backwards	backwards	forwards	forwards
10	78,5	0,04	82,58	0,04
20	36,08	0,08	30,46	0,10
30	40,58	0,07	40	0,08
40	39,85	0,08	26,46	0,11
50	28,53	0,11	19,44	0,15

Tapasztalat: a robot közel a 30 as teljesítményig ugyanolyan sebességgel haladt hátrafele, mint előre. Ebből azt a következtetést vonhatjuk le, hogy a hárompontos járás biztosabb mozgást ad, mint az ellenpáras. Az ellenpáras csak 20-ig volt szinte teljesen ugyanaz.



4.3.33 sebesség függvényében



4.3.34 az idő függvényében

5. Programozás, a projekt szoftver specifikációja

5.1 Bevezetés

A projekt célja egy önálló, alapvető navigációra képes robotkutya fejlesztése, amely érzékeli a környezetében található akadályokat, és képes megkerülni azokat. A robot az EV3 Mindstorms platformon épül fel és Python programozási nyelvvel van vezérelve. Az ev3dev operációs rendszer, amely Debian-alapú, támogatja a fejlesztést, lehetővé téve, hogy Pythonon kívül más nyelvek is alkalmazhatók legyenek a későbbiekben.

5.2 Eszközök és környezet

A robot hardveres környezetét a LEGO EV3 brick képezi, amely egy mikrokontroller, kifejezetten robotikai alkalmazásokhoz fejlesztve. Az operációs rendszer, ev3dev, egy Debian-alapú Linux, amelyet SD-kártyára telepítettünk, és széles körű szoftver-támogatást nyújt Pythonhoz, C++-hoz, JavaScripthez és más nyelvekhez is, így megkönnyítve a bővíthetőséget.

A fejlesztési környezethez a Visual Studio Code-ot használtuk, amelyhez egy ev3dev browser extension (bővítmény) társul. Ez a bővítmény egyszerű feltöltési és távoli kódfuttatási lehetőséget biztosít. Az SSH-távoli elérés lehetővé teszi a terminál-hozzáférést és a program monitorozását, hibakeresését valós időben.

5.3 Módszertan

A Python választása a széleskörű támogatottsága és egyszerű szintaxisa miatt történt, különösen hasznos egy prototípus-fejlesztéshez. Az **ev3dev2** Python könyvtár hozzáférést biztosít az EV3 motorokhoz, érzékelőkhöz, gombokhoz és hangszóróhoz.

A kódot több modulra bontottuk, amelyek külön funkciókat szolgálnak:

- **Mozgásvezérlő modul:** A robot alapvető mozgásfunkcióit kezeli.
- **Érzékelő modul:** A szenzorokkal való adatgyűjtést és adatfeldolgozást végzi.
- **Interakciós modul:** A felhasználói visszajelzéseket, hangjelzéseket és kijelzőműködést vezérli.

A fejlesztés iteratív módon történt, moduláris teszteléssel és hibakereséssel, amely során az SSH protokoll és a Visual Studio Code ev3dev bővítménye jelentős támogatást nyújtott.

5.4 Funkcionális követelmények

A robot fő funkcionális céljai:

- **Előre mozgás** akadálymentes úton.
- **Akadályérzékelés** infravörös szenzorral, amely észlelés esetén megállítja a robotot, és hangjelzéssel figyelmeztet.
- **Oldalirányú körbenézés** a szabad irány azonosításához, és elfordulás a megfelelő irányba.
- **Hangjelzés** a felhasználó számára, amely információkat ad a robot helyzetéről és az akadályokról.

5.5 Nem-funkcionális követelmények

- **Interfész:** A robot a Visual Studio Code ev3dev bővítménnyel tölthető fel és vezérelhető.
- **Válaszidő:** A robot gyorsan reagáljon az akadályokra (pár milliszekundumos késleltetés).
- **Hangjelzés:** Az ev3dev beépített hangszóróját használva biztosít figyelmeztetést.
- **Felhasználói beavatkozás:** A robot EV3 gombokkal vezérelhető, a jobb gomb a program indítását, a bal gomb pedig a leállítást végzi.

5.6 Technikai követelmények

- **Hardver:**
 - LEGO Mindstorms EV3 vezérlőegység (brick)
 - Nagy motorok a robot mozgatásához (OUTPUT_A és OUTPUT_D)
 - Közepes motorok a csípő és szenzormozgatáshoz (OUTPUT_B és OUTPUT_C)
 - Infravörös szenzor az akadályok érzékelésére (INPUT_4)
- **Szoftver:**
 - Operációs rendszer: ev3dev Debian-alapú Linux
 - Programozási nyelv: Python 3
 - Könyvtárak: ev3dev2 (motor, szenzorok, hang, gombok vezérléséhez)
 - Visual Studio Code IDE ev3dev browser extension-nel

5.7 Fő funkciók és működési logika kódrészletekkel

a. Előre mozgás (move_forward)

Az alábbi kód a robot előre mozgását valósítja meg a MoveTank mód segítségével, amely szinkronizálja a két nagy motor mozgását. A sebesség paraméterezhető, és alapértelmezés szerint a motorok 50%-os teljesítménnyel futnak.

```
# előre mozgás
def move_forward():
    print("move forward")
    tank.on_for_rotations(SpeedPercent(-50), SpeedPercent(50), 1)
```

Ez a funkció addig működik, amíg akadályt nem érzékel. A speed változó segítségével a sebesség testreszabható.

b. Megállás (stop_motors)

A stop_motors függvény a robot megállításhoz használható. Kikapcsolja az összes motort, és visszaállítja az alaphelyzetet, amikor akadályt észlel vagy a felhasználó leállítja a programot.

```
# motorok leallitasa
def stop_motors():
    tank.off()
```

Ez a funkció minden aktív motort kikapcsol, megszüntetve a robot mozgását.

c. Hátramenet (move_backward)

A hátrameneti funkció visszatolatja a robotot, hogy távolodjon az akadálytól. A motorok 40%-os sebességgel működnek, egy teljes fordulatot tesznek meg hátrafelé.

```
# hatra mozgás
def move_backward():
    print("move backward")
    tank.on_for_rotations(SpeedPercent(40), SpeedPercent(-40), 1)
```

A -40 sebesség érték azt jelenti, hogy a motorok fordított irányban futnak.

d. Balra és jobbra nézés az akadály kikerüléséhez (check_left_and_right)

Ez a funkció a szenzormotort -90 és +90 fokkal mozgatja balra és jobbra, hogy körülnézzen az akadály elkerülése érdekében. Az infravörös szenzorból származó adatok alapján dönthető el, hogy melyik oldal szabad.

```
# balra jobbra nez
def check_left_and_right():
    # bal ellenorzes
    sensor_motor.on_to_position(SpeedPercent(50), -90)
    time.sleep(0.5)
    left_clear = ir.proximity > 30 # finomhangolható az ellenorizando
    tavolsag

    # jobb ellenorzes
    sensor_motor.on_to_position(SpeedPercent(50), 90)
    time.sleep(0.5)
    right_clear = ir.proximity > 30 # finomhangolható az ellenorizando
    tavolsag

    # motor vissza kozepre
    sensor_motor.on_to_position(SpeedPercent(50), 0)

    return left_clear, right_clear
```

A függvény visszaadja a balra és jobbra mért távolságot, amely segít eldönteni, melyik irányba érdemes elmozdulni.

e. Akadályérzékelés és hangjelzés (announce_obstacle)

Az akadályérzékeléshez az infravörös szenzor használatos, amely egy 40-es távolságértéknél közelebbi objektumot akadályként érzékel. Ha akadályt talál, a robot hangjelzést ad.

```
# hanggal figyelmeztet, csak angol nyelv használható
def announce_obstacle():
    sound.speak('Obstacle detected! Stopping and checking for clear path.')
```

A speak függvény segítségével hangos figyelmeztetést ad a robot, jelezve az akadály észlelését.

f. Irányváltás hátralépés és elfordulás útján (move_back_left, move_back_right)

Ha az akadály érzékelése során a bal vagy jobb oldal szabad, a robot hátralép, majd az egyik irányba elfordul, hogy elkerülje az akadályt.

hatra balra

```
def move_back_left():
    print("move back left")
    csipo.on_for_rotations(SpeedPercent(-30), 0.2)
    tank.on_for_rotations(SpeedPercent(40), SpeedPercent(-40), 5)
    csipo.on_for_rotations(SpeedPercent(30), 0.2)
```

hatra jobbra

```
def move_back_right():
    print("move back left")
    csipo.on_for_rotations(SpeedPercent(30), 0.2)
    tank.on_for_rotations(SpeedPercent(40), SpeedPercent(-40), 5)
    csipo.on_for_rotations(SpeedPercent(-30), 0.2)
```

A move_back_left és move_back_right funkciók különböző irányokba fordítják el a robotot, hogy elkerüljék az akadályt.

5.8 Felhasználói interakciók

A robot a gombok megnyomására reagál, a következőképpen:

```
if btn.right: # Start
    running = True
    print("Program started")

    if btn.left: # Stop
        running = False
        stop_motors()
        print("Program stopped")
```

A Button osztály lehetővé teszi a robot számára, hogy érzékelje, ha a felhasználó megnyomta a jobb vagy bal oldali gombot.

6. Összegzés

A projekttel egy olyan periodikusan ismétlődő mozgást valósítottunk meg, amely lehetővé teszi, hogy egy négy lábon járó robot stabilan és hatékonyan közlekedjen két motor segítségével. A rendszert úgy terveztük meg, hogy minimalizálja a súlypontáthelyezés okozta instabilitást, amely gyakran problémát jelent az ilyen típusú járórobotoknál. Az eredmények azt mutatták, hogy a kétmotoros megoldás nem minden esetben biztosít stabil járást, azonban csökkenti a komplex mechanikai struktúrák szükségességét, lehetővé téve a költséghatékony, de jól működő négy lábon járó robotok fejlesztését. A robotot mechanikai és programozási szempontból is optimalizáltuk, annak érdekében, hogy képes legyen hatékonyan alkalmazkodni különböző sebességi beállításokhoz anélkül, hogy a mozgás közben jelentős súlypont-áthelyezési problémák lépnének fel. Az eredmények alkalmazhatók olyan robotikai rendszerek fejlesztésében, ahol a mechanikai egyszerűség fontos szempont.

A fent bemutatott kód részletek és modulok segítségével egy egyszerű, alapvető akadálykerülő funkciókat végrehajtó robotkutya prototípust fejlesztettünk. A kód könnyen bővíthető és módosítható, lehetővé téve további funkcionalitások hozzáadását, például a mesterséges intelligencia és a gépi tanulás integrálását.

A roboton az egyik legnagyobb végrehajtható fejlesztés, hogy minden lábat egy külön motor hajtson meg, azonban ehhez kettő Intelligent Brick-et kell összehangolnunk a porthiány következtében. Emiatt a lábakat majd nagyobb teherbírásra kell átterveznünk, hogy a jelenlegi súly kétszeresét is képes legyen megtartani. A terveink között szerepel, hogy a robot magától döntse el, hogy melyik mozgást hajtja végre különböző járófelületeken. LEGO elemek korlátozottsága miatt, a robot egyes részeit érdemes lesz lecserélni direkt erre a célra legyártott alkatrészekkel. Módunkban áll ezeket az alkatrészeket 3D nyomtatás technológiájával létrehozni. Amennyiben ezek a módosítások nem megoldhatóak kettő LEGO Mindstorms EV3 Intelligent Brick összehangolásával, akkor más platformra váltva tervezzük megvalósítani ezen fejlesztéseinket. A jelenlegi robot a két motor által végzett lábmozgatás miatt nem alkalmas Gyro sensor beépítésére. Lehetőségünk van Gyro sensor alkalmazására, amivel érzékelni tudnánk a súlypont elmozdulást. A Gyro sensor jelzésére, négy motor esetén a lábakat külön tudnánk vezérelni. A Gyro sensort csak négy motor esetén tudnánk alkalmazni.

Források

[1] **Mozaik Fizika 11-12** közép és emelt szintű érettségire készülőknek (dr.Halász Tibor, dr. Jurisits József, dr. Szűcs József)

[2] **Fizika Ennyit kellene tudnod:** Gulyás János, Rácz Mihály, Tomcsányi Péter, Varga Antal

[3] **Fizika:** George Gamow, John M. Cleveland

[4] **Lego EV3 User guide**

https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser_Guide_LEGO_MINDSTORMS_EV3_11_Tablet_ENUS.pdf 2024.11.10.

Mindstorms és alkatrészek

<https://ev3-help-online.api.education.lego.com/Education/en-gb/page.html?Path=blocks%2FLEGO%2FMediumMotor.html>

<https://ev3-help-online.api.education.lego.com/Education/en-gb/page.html?Path=blocks%2FLEGO%2FMotor.html>

https://ev3-help-online.api.education.lego.com/Education/en-gb/page.html?Path=editor%2FUsingSensors_Infrared.html

<https://www.kockaaruhaz.hu/lego/lego-mindstorms/45509-lego-mindstorms-ev3-infrared-sensor-infravoros-szenzor.html> kép

<https://www.gyoriebrehab.hu/post/mire-j%C3%B3-a-farkask%C3%B6r%C3%B6m>

Youtube

https://www.youtube.com/watch?v=B_DLrNIQCo8&t=420s

https://www.youtube.com/watch?v=MC8_ylHKBOc

<https://www.youtube.com/watch?v=OxsmLYQWCJk>

<https://www.youtube.com/watch?v=beA-0t1TeUI>

<https://www.youtube.com/watch?v=ji7uis-tFqw>

<https://www.youtube.com/watch?v=IGW8zYPxFxE>

<https://www.youtube.com/watch?v=ZwKdHJdCMgg>