

SZAKDOLGOZAT

Rusznák Norbert

Debrecen

2010

Debreceni Egyetem
Természettudományi és Technológiai Kar
Szilárdtest Fizika Tanszék

Felderítő Robotok

Navigáció belső szenzorok segítségével

Témavezető:

Dr. Szabó István

Egyetemi Docens

Tanszékvezető

Készítette:

Rusznák Norbert

Mérnök informatikus

Debrecen

2010

Tartalomjegyzék

1. Bevezetés	4
2. A robotok	5
2.1. Általános ismertető	5
2.2. Robotok csoportosítása.....	8
2.2.1. Ipari robotok	8
2.2.2. Kutatásban felhasznált robotok.....	10
2.2.2.1. Mobil robotok.....	10
2.2.3. Otthoni és szórakoztató robotok	12
2.2.4. Gyógyászatban felhasznált robotok.....	13
2.2.5. A feladatomban felhasznált robot.....	14
3. Navigáció	16
3.1. Műholdas helymeghatározás	17
3.2. „Gépi látás”.....	20
3.3. Inerciális rendszerek	21
4. Érzékelők	22
4.1. Általános ismertetés.....	22
4.2. Felhasznált szenzorok.....	24
4.2.1. Gyorsulásmérő.....	25
4.2.1.1. Hitechnic gyorsulásmérő	27
4.2.2. Iránytű.....	28
4.2.2.1. Hitechnic iránytű	28
4.2.3. Giroszkóp.....	30
4.2.3.1. Hitechnic giroszkóp.....	30
5. A projekt	30
5.1. Projekt ismertetése.....	30
5.2. A futtatókörnyezet	32
5.3. A program bemutatása.....	34
6. Összefoglalás	44
7. Köszönetnyilvánítás.....	45
8. Irodalomjegyzék	45

1. Bevezetés

Talán nincs is olyan ember a világon, akinek ne lenne valamiféle fogalma a robotokról. Leginkább ez annak köszönhető, hogy több tucatnyi film készült már ezekről a gépekről. A sci-fik egyik fő témája, hogy az univerzális automaták átveszik az irányítást a Föld lakói fölött, esetleg céljuk elpusztítani bolygónkat. De tudjuk azt, hogy a gépek jelenlegi mesterséges intelligenciája nem képes arra, hogy önálló döntéseket hozzanak, csupán az emberek által vezérelt utasításokat hajtják végre.

Rohamosan fejlődő világunkban egyre gyakrabban hallunk híreket, hogy milyen okos és intelligens szerkezetek látnak napvilágot, amelyek nagyban megkönnyítik az életünket. A takarító, fűnyíró robotok már valóságosak, de az emberi ötletességnek köszönhetően egyre több felhasználási területen használhatóak.

Bár hazánk meglehetősen lemaradt robotika terén, manapság egyre több robotfejlesztő versenyről lehet hallani, ami azt jelenti, hogy van rá érdeklődés. Szerintem, ha az iskolákban több robottal ismerkednének meg a tanulók, jobban vonzaná őket ez a terület, hiszen sokkal izgalmasabb dolog maga a programozása is, mint csupán számítógép előtt ülve látni egy végeredményt. Japán vezető állam a robottechnológiában, az ottani gyerekek már általános iskolában találkoznak a LEGO robotokkal, amin egyébként én a szakdolgozatomat készítettem. Egy átlag személyi számítógép árából majdnem két ilyen programozható egység megvehető.

Mivel számomra ez egy rendkívül érdekes terület, nem volt kérdéses mit is válasszak szakdolgozati témának. Kiskorom óta érdekelnek a technikai újdonságok. Egyik álmom, hogy egyszer építek egy saját robotot és azt fejlesztgetem, míg nem sikerül olyat alkotnom, amely a mesterséges intelligencia lehető legtöbb lehetőségét kiaknázza.

Projektemben egy olyan érzékelővel felszerelt robot mozgását mérem, amely olyan körülmények között is képes eredményt szolgáltatni, ha megszűnik minden külső kommunikációs és adatszerzési lehetőség. Hiszen egy felderítő robotnál fontos a navigációs képesség. Emellett általános leírást adok a robotokról, jellemzem őket, majd szeretném bemutatni, amit az érzékelőkről és navigációról tudni érdemes.

Céлом egy belső szenzorokat felhasználó navigációs rendszer kifejlesztése volt. Néhány esetben előfordulhat, hogy egyes globális helymeghatározó és irányító rendszerek vagy egyéb külső forrásból származó adatokat feldolgozó architektúrák nem működnek megfelelően. Napjaink legismertebb és legelterjedtebb ilyen rendszere a GPS (Global Positioning System). Sajnos még a GPS-nél is előfordulhat néhány esetben, hogy megszűnik a kommunikáció, le van árnyékolva a jeleket vevő antenna, alagúton haladunk át vagy fölöttünk egy vonalban vannak a műholdak. Ez esetben nem tudunk jeleket venni, ilyenkor segíthetne az általam elkészített projekt, amiben felhasználok egy gyorsulásmérőt, egy giroszkópot és egy digitális iránytűt. Ezek önmagukban nem nyújtanak magas fokú megbízhatóságot, de más rendszerekkel kiegészítve sok esetben hasznosnak bizonyulhat.

A felépített robot elvárása volt, hogy dinamikusan kövesse a környezet változásait, a tervezett célt minden körülmény között teljesítse, hiszen nem előre megadott pozícióra állítom, hanem bárhol képes eredményeket szolgáltatni.

Források után kutatva tapasztaltam, hogy kevés magyar nyelvű irodalom érhető el a témáról, így én is szeretném gazdagítani a hazai szakirodalmat.

Ezen dokumentum értelmezése megkíván bizonyos fokú tudást a fizika és informatika területeiről valamint alapfokú java programozási ismereteket.

2. A robotok

2.1. Általános ismertető

A robot szóra sokáig nem volt általánosan elfogadott definíció, inkább többféle meghatározás terjedt el az irodalomban. Mára azonban a Nemzetközi Szabványügyi Szervezet definiálta a hivatalos értelmezést, melyet nagyjából így hangzik magyarul: Automatikusan vezérelt újraprogramozható, univerzális, programozható beavatkozó három vagy több tengelyen, amely lehet egy helyben rögzített vagy mozgatható, ipari automatizálási alkalmazásokra használható. (ISO8373)¹

¹ <http://www.docstoc.com/docs/21202778/DEFINITION-OF-A-ROBOT-Manipulating-industrial-robot-as-defined/>

De a legtöbb irodalomban úgy fogalmazzuk, ha bizonyos feltételeket elér, akkor robotnak nevezhetjük, egyik ilyen legfontosabb elvárás, hogy az emberhez hasonló fizikai vagy szellemi munkát végezzen.

Általánosan önálló, saját memóriával rendelkező, intelligens mechatronikai szerkezetek, vagyis elektromechanikai, elektronikai és mechanikai részekből állnak. Mozdulatokra vagy fizikai feladatok elvégzésére képesek. A rátöltött program segítségével különböző feladatokat végre tudnak hajtani, ami történhet emberi irányítás alatt vagy önállóan. Tevékenységei közé tartozhatnak olyan célok kivitelezése, amelyek veszélyesek, nagy pontosságot igényelnek, nehezek, emberek számára lehetetlenek végrehajtani vagy az elvégzendő munka monoton jellegű, illetve egyszerűen csak megkönnyítik a mindennapjainkat

A kifejezés Karel Čapek 20. századi cseh írójánál jelenik meg először. A szó a szláv robota kifejezésből ered, jelentése: szolgaság, munka. Az első programozható berendezést George Devolt szabadalmaztatta 1954-ben.

Ezek az univerzális automaták érzékelők segítségével gyűjtenek információkat a külvilágból.

A robotoknak a következő képességeik lehetnek:

1. tanítható mozgásra illetve viselkedésre
2. munkát végez, a környezetében lévő tárgyak állapotát megváltoztatja
3. érzékeli a környezete jeleit, ami szenzorok útján fog fel
4. adatokat dolgoz fel, a kapott jeleket feldolgozza
5. a feldolgozott adatokat és az algoritmus alapján megváltoztatja a viselkedését

Tulajdonképp ez egy olyan számítógép, ahol érzékelő és beavatkozó eszközök állnak összeköttetésben. A rajta futó program alapján képes a külvilágból származó jeleket fogadni, feldolgozni, értékelni és ezek alapján valamilyen utasítást végrehajtani, azonban csak az algoritmusban rögzített problémák megoldását tudja véghezvinni. Ha olyan nehézségbe ütközik, amely nincs lerögzítve, akkor azt figyelmen kívül hagyja, így programozásánál minden eshetőségre gondolnunk kell.

Isaac Asimov megalkotta a robotika három törvényét:

1. „A robotnak nem szabad kárt okoznia emberi lényben, vagy tétlenül tűnie, hogy emberi lény bármilyen kárt szenvedjen.”
2. „A robot engedelmeskedni tartozik az emberi lények utasításainak, kivéve, ha ezek az utasítások az első törvény előírásaiba ütköznének.”
3. „A robot tartozik saját védelméről gondoskodni, amennyiben ez nem ütközik az első vagy második törvény bármelyikének előírásaiba.”

A robotoknak 3 generációját különböztetjük meg:

- I. generáció
 - alkalmazása a 60-as évekre jellemző leginkább
 - nem rendelkeznek szenzorokkal
 - tárgyak cipelésére használják
 - a programozás alacsony szinten zajlik, a program meghatározza a robot valamennyi mozdulatát
- II. generáció
 - a robotok már szenzorok segítségével érzékelik a külvilágot, ennek alapján feldolgozzák a környezet jeleit, így robotok módosítani tudják saját viselkedésüket
 - már lehet magas szinten programozni
 - 70-es években használják
- III. generáció
 - mesterséges intelligenciát felhasználva képesek viselkedési algoritmusok alapján döntéseket hozni
 - bonyolult, összetett feladatokat tudnak végrehajtani
 - általános jellegű programírás elterjedt, konkretizálni, csak az adott hardverre kell
 - robotok felhasználási köre kibővül, innentől felhasználják az űrkutatásban, laboratóriumban és sok más helyen.[6]

2.2. Robotok csoportosítása

A robotokat számos szempont szerint csoportosíthatjuk. Én most felhasználásuk szerint szeretném osztályozni, aztán részletezni őket.

- Felhasználásuk szerint
 - Ipari robotok
 - Kutatásban felhasználtak
 - mobil robotok
 - Otthoni
 - Gyógyászati

2.2.1. Ipari robotok

A VDI 2860 (1981) szerint: „az ipari robot univerzálisan állítható többtengelyű mozgó automata, amelynek mozgásegymásutánisága (utak és szögek) szabadon – mechanikus beavatkozás nélkül – programozható és adott esetben szenzorral vezetett, megfogóval, szerszámmal vagy más gyártóeszközzel felszerelhető, anyagkezelési és technológiai feladatra felhasználható.”

A robotok 95-98%-a ehhez a kategóriához tartozik. Leginkább a nagy pontosságot igénylő feladatokhoz alkalmazzák őket. Gyakran helyváltoztatásra sem alkalmasak (standig), csupán egy kart tudnak mozgatni, amivel a munkát végzik. Vagy épp három tengely mentén, előre lerögzített pályán tudnak mozogni, az előzőhöz képest sokkal nagyobb mértékben megnövelve a munkaterületük nagyságát. Alkalmazásánál fontos szempontoknak tartják a munkaerő-költség csökkentését, ha egyenletesen megfelelő minőségi termékeket szeretnének gyártani, illetve ha bővíteni akarják a termelékenységet, vagy ami a legfontosabb, veszélyes környezetben használhatóak, így nem kell kockáztatni az emberek életét és egészségét.

Az ipari robotok általában két részből állnak, a karból és az ezt mozgóató vezérlőegységből. A mozgó rész szerkezetileg három fő komponensből áll: karból, csuklóból és műveletvégzőből. A kar mozog a feladatvégzés pozíciójába. Mozgása lehet egyenes vagy elforduló irányú. Több tengelypozícióból is elérhető ugyanaz a cél. A csuklóval állítható be, hogy megfelelő irányból történjen a munkavégzés. A megfogó szerkezet lehet pneumatikus (sűrített levegővel működő), mechanikai vagy elektromágneses.

Tömegtermeléshez kiválóak, gyorsak, nem fáradnak el és nagyobb terhet képesek emelni vagy cipelni, mint az emberek és mindig ugyanolyan precízen dolgoznak. Jellemzőek az autógyártásban, ahol szinte valamennyi munkafolyamatot robotok végeznek az összeszereléstől, a hegesztésen át a festésig. Védelmi funkciót is beépítenek, a vészleálló elektronika áramtalanítja a kart, például abban az esetben, ha ütközést érzékel.

Az ipari robotoknak legalább 5-6 tengelyük van, hiszen végre kell hajtaniuk ugyanazokat a tevékenységeket, amelyeket egy emberi kéznek. A gépek felépítését emberi végtagról mintázzák. A tengelyek vezérlését külön-külön motor látja el, amik egymástól függetlenül működnek. A motorok többsége 3 fázisú szervomotor inkrementális jeladóval és fékkel. A jeladók a tengely helyzetét továbbítják a vezérlőegységhez. A fékek a tengelyek helyes pozícióját őrzik meg, nehogy a kelletténél továbbforduljanak. A robotkarnak pontosan kell mozognia, akkor is ha a terhelés változik. A vezérlő a bejövő jelek alapján mozgatja a robotot. Nagyfokú összehangoltság kell a motorok precízen történő pontos mozgatásához, amelyet megnehezít a rájuk ható gravitációs erő is. Ez sok számítási feladattal jár együtt.

Az ipari robotok gyártói általában különböző színeket használnak termékeiken, így egyszerűen meg tudjuk különböztetni őket.[8]



1. ábra - Ipari robot alkalmazása az autógyártásban [I]

2.2.2. Kutatásban felhasznált robotok

A kutatási robotok a 80-as évektől léteznek. A kutatók a robotokat saját maguk építették aprólékosan a semmiből. Számos autonóm robot szoftvere nyílt forráskódú és számtalan ilyen eszköz elérhető fejlesztés céljából.

A kutatásban felhasznált robotoknál lényeges dolog, hogy képesek legyenek helyváltoztatásra, ezeket hívjuk mobil robotoknak. Ezen kategóriához sorolhatjuk még az űrkutatási és felderítő robotokat is, amelyeket olyan közegben is használhatnak, ami az ember számára elviselhetetlen vagy épp túl veszélyes. A katonaság aknakereső és megsemmisítő, ellenséges területeket felderítő, csapásmérő és többek között még őrző-védő robotot is használ, hogy ne veszélyeztessenek emberi életet. De más alkalmazási területeken is bevetik. Léteznek tűzoltó, régészeti robotok. Űrkutatásban leginkább Mars-expedíciókban használják élet utáni kutatásra és anyag-minta gyűjtésére.

2.2.2.1. Mobil robotok

Kiemeltem a mobil robotokat a kutatás témakörből, mivel a robot, amivel dolgoztam, ide tartozik. Ilyen típusú robotokat nem csak a kutatásban használnak, de legnagyobb részük ide tartozik, jelenleg ezek állnak a robotika fejlesztés középpontjában. Ezen kívül alkalmazzák a szórakoztatásban, katonaságban és még az iparban is előfordulnak.

A mobil robotok automata berendezések, amelyek mozgásra képesek egy adott környezetben, tehát nincsenek lerögzítve semmilyen fizikai pozícióhoz. A mobil robotok mindenféle környezetben előfordulhatnak:

- víz alatti robotok:
víz alatti autonóm járműveknek hívják őket(AUV)
- földi robotok:
leggyakrabban kerékkal felszereltek, de más „mozgatószerve” is lehet, ez esetekben láb használata a jellemző, amelyből előfordulhat kettő: humanoidok, vagy akár több is: kedvelt állatok vagy rovarok
- légi robotok:
pilóta nélküli légi járművek

A mobil robotok leggyakrabban a következő mozgási formákat használják:

- láb:
ez alapján beszélhetünk lépegető, ha vízszintes talajon munkálkodnak, illetve mászó robotról, ha lejtőn vagy meredek falon tevékenykednek. Legelterjedtebb az emberi láb, android vagy az állatokról mintázott lábak.
- kerék:
akadályokon nem képes átjutni, inkább ki kell kerülni azokat, cserében gyors helyváltoztatásra képesek. Két, három, négy vagy hatkerekű konstrukciókat alkalmaznak.
- lánctalp
- repülő robotok

A robotok mozgása történhet:

- kézi irányítással:
egy személy teljes mértékben felel a navigálásért. Ez lehetséges közvetlen összeköttetésben a robottal vagy vezeték nélküli kapcsolat segítségével.
- őrzötten:
az akadályokat képes érzékelni és elkerülni, de a vezérlő személy átveheti az irányítást.
- vonalkövető módra:
egy előre megfestett vonalat követnek, ha akadály kerül a vonalra nem tudják kikerülni.
- Önállóan, véletlenszerűen:
mozgásuk véletlenszerű, például a Friendly Robotics fűnyíró.
- Önállóan, irányítva:
legalább a minimális információt ismeri a végcéllal kapcsolatban, szükség esetén irányítható.
- Autonóm:
teljesen automatikus, de néhány autonóm berendezés támogatja a kézi irányítási funkciót is.[7]

2.2.3. Otthoni és szórakoztató robotok

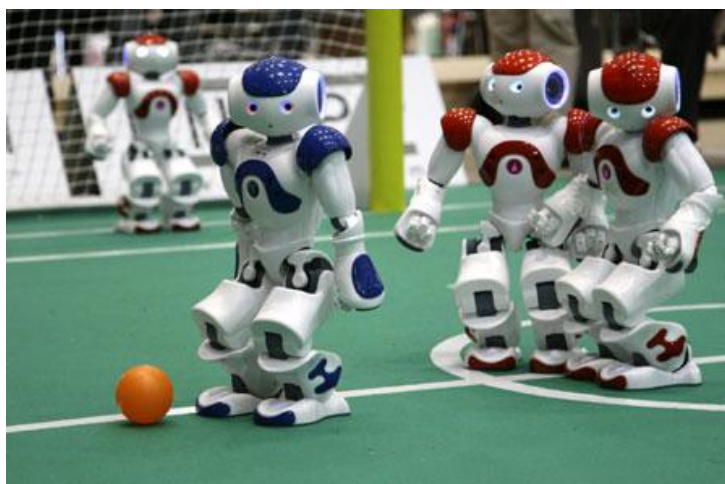
A mesterséges intelligencia fejlődésének köszönhetően számuk egyre inkább növekszik. Az Egyesült Államokban többszörösére nőtt az elmúlt években a felhasználásuk. Ha ilyen ütemben gyarapodnak tovább, előfordulásaik utolérlik az ipari robotokét, sőt előbb-utóbb meg is fogják előzni azokat. Az otthoni robotok képesek valamilyen vagy több háztartási munkát ellátni, mint a porszívózás, fűnyírás stb. Akadnak olyanok is, amik segítenek az időseken, betegeken, figyelmeztetik a gyógyszereik bevitelére és magányos világukba egy kis vidámságot visznek, esetleges baj esetén pedig értesítik az illetékes közeget.

A szórakoztató robotok általában hasznos munkát nem végeznek, csupán szórakoztatnak. Az animatok, állatszerű kinézetre és viselkedésre mintázott robotok. Leghíresebb a Sony által gyártott Aibo (Artificial Intelligence RoBOt, vagyis Mesterséges Intelligenciájú Robot) robotkutya. A mesterséges értelmet jelenlegi állapotában kihasználó gép átprogramozható, így új viselkedésformákat is megadhatunk neki, egyébként a parancsszavakat megérti és képes a tanulásra.



2. ábra - A Sony Aibo nevű robotkutya [II]

Különbéféle versenyeket rendeznek annak felderítésére, hogy épp milyen szinten áll a tudományág. Egyik leghíresebb és legcélratoróbb a RoboCup, hiszen itt egyszerre kell, hogy megfelelő legyen a mechanika, az érzékelés, a gyors válasz valamint a gépi értelem. A cél az, hogy a robotok labdarúgásban legyőzzék az aktuális emberi ellenfelet, ami még jelen pillanatban nagyon-nagyon messze áll a törekvéstől.

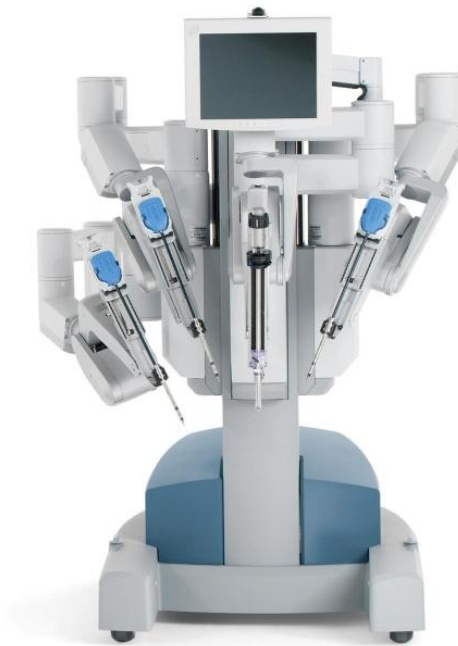


3. ábra - Robotfocicsapat 2009-ben [III]

Magyarországon a robotgyártás gyerekcipőben jár. Nincs kereslet a robotokra. Az iparban is keveset használnak. A saját készítésű robotokkal mindig probléma van, mert nem megfelelő a hardvere. A szoftvergyártásban vannak eredményeink, de ha valaki vesz egy robotot, akkor azt a mechanikával együtt veszi.

2.2.4. Gyógyászatban felhasznált robotok

A legismertebb gyógyászati automatát da Vincinek hívják. Segítségével nehezen vagy veszélyesen hozzáférhető helyeken végezhetnek műtétet az emberi szervezetben. Ráadásul csak kis metszést kell ejteni, így csökkenthető a vérveszteség, gyorsabb a gyógyulási idő és az eszköz az orvos kezének remegését is képes kiküszöbölni. Úgy működik, hogy az orvos a számítógépes rendszere előtt foglal helyet, és kezével vezérli a robot karjait, a manipulátorok leutánozzák az orvos mozdulatait. A beavatkozást a műtétet végző személy a gép végtagjaira erősített kamerák által mutatott 3 dimenziós képen kísérheti figyelemmel. A karok végződése cserélhető csipeszre, szikére vagy fogóra.



4. ábra - A "da Vinci" műtéti robot

Orvosi célokra használják a nanorobotokat is. A parányi méretű botok a véráramba jutva fejtik gyógyító hatásukat. Előnye abban rejlik, hogy megközelíthetetlen helyeken tud hozzáférni az emberi sejtekhez. Mozgékonyak, gyorsan felismerik a sejteket és fejlett navigációval rendelkeznek. Legújabb kutatások szerint nanorobotokkal megoldható lenne akár rákos sejteket is elpusztítani.

2.2.5. A feladatomban felhasznált robot

Feladatom elvégzéséhez a LEGO által gyártott Mindstorms NXT nevű csomagot használtam, amely megfelelőnek bizonyult a cél eléréséhez.

Az gyári NXT készlet tartalma:

- 619 darabból álló LEGO építőelem kerekkel, gumiabroncsokkal, fogaskerekekkel és egy pályával
- NXT téglá, amely egy mikroszámítógép. Ez vezérli a szenzorokat és motorokat, ezt kell programozni. Ebben beépítve megtalálható:
 - USB 2.0 port, 12 Mbit/s-os adatátviteli sebességgel
 - Bluetooth 2.0, frekvencia: 2,402-2,480 GHz
 - 100 x 64 pixeles fekete-fehér LCD kijelző

- 4 darab nyomógomb
- beépített hangszóró, 8 bites felbontású hangminőséggel
- ARM7 32-bites, 48 MHz-es alacsony fogyasztású processzor
 - 256 kilobyte flash memória
 - 64 kilobyte RAM
- 8-bites, 4 MHz-es AVR mikrokontroller
 - 4 kilobyte flash memória
 - 512 kilobyte RAM
- áramforrása 6 x AA (1,5 V) elem
- 1 darab érintés érzékelő. Akkor aktiválódik, ha benyomódik a rajta lévő kis gomb, tárgyak ütközésének érzékelésére kiválóan alkalmas.
- 1 darab hangszenzor. Hangszint mérésére szolgál. Különböző hangmintákat képes felismertni.
- 1 darab fényszenzor, amely meg tudja különböztetni a színeket és a környezete megvilágítását. Az RGB színkód alapján dolgozik.
- 1 darab ultrahangos szenzor, amely úgy működik, mint a radar, ezzel lát az NXT. 255 cm-en belül tud érzékelni.
- 3 darab szervomotor beépített fordulatszám érzékelővel, amely + / - 1°-kal meg tudja mondani mennyit fordult a motor. Nagyobb forgási sebesség esetén kisebb a forgatónyomaték, azaz erőtlenebb a motor, míg kisebb forgás esetén nagyobb a nyomaték. Egy biztonsági elemet is beépítettek, ami megakadályozza, hogy ne éghessen le a motor, ha nem tud forogni, akkor sem. Ha mi forgatjuk a motor, akkor az generátorként viselkedik, áramot termel.
- 7 darab 6 eres RJ12 kábel a motorok és érzékelők számára (nem összekeverendő az RJ11 telefonkábelrel). Analóg interfész esetén visszafelé kompatibilisek, csak a régebbi rendszerek használják, digitális esetén támogatja az I2C és az RS-485 – ös kommunikációt is. A kábelvonalak a következő jeleket sugározzák:
 - 1. analóg interfész, +9V, színe: fehér
 - 2. földpont, színe: fekete
 - 3. földpont, színe: piros
 - 4. +4,3V, színe: zöld
 - 5. I2C órajel, színe: sárga

- 6. I2C adatvonal, színe: kék
- felhasználói és robot építési kézikönyv
- CD grafikus NXT-G programnyelvvel

A saját robotom az alap robotból készítettem, úgy hogy leszereltem minden érzékelőt, helyette felvértettem egy giroszkóppal, egy gyorsulásmérővel és egy digitális iránytűvel.



5. ábra - A fejlesztett NXT robot

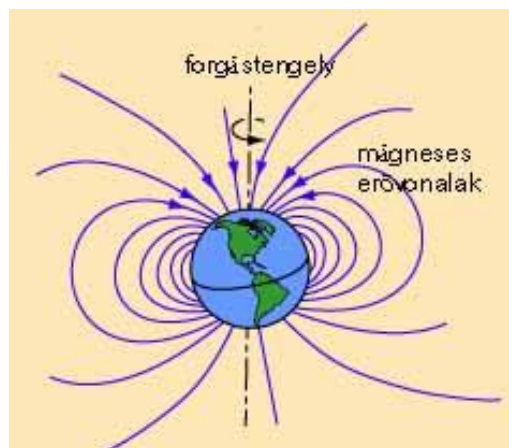
3. Navigáció

A navigáció a közlekedésben való tájékozódás meghatározását jelenti. Legfontosabb feladata meghatározni a jármű aktuális pozícióját, illetve annak elnavigálását egy adott célhoz előre meghatározott útvonalon vagy a legjobb útvonal meghatározásával. A navigációs rendszer legfontosabb feladata, ha szükségünk van rá, bármikor rendelkezésünkre álljon.

A navigáció a hajózásban a legjelentősebb mértékű. Többféle tájékozódási rendszert használnak a tengeren és óceánon. Használják a műholdas kommunikációt, de ezt csupán iránymutatásnak veszik, a GPS pontatlansága miatt. Helyette még mindig papír térképet használnak előszeretettel, berajzolják a tervezett útvonalat. Ezek megtervezéséhez adatokat gyűjtenek, megállapítják az esetleges kockázatokat. Az váratlanul felmerülő körülmények esetére kidolgoznak útvonalmódosítást is.

A hajózás jövője valószínűleg az inerciális rendszerekben rejlik. Ha ez esetben megadjuk a kiindulási koordinátákat, akkor a mozgásérzékelők segítségével az irány és a sebesség alapján meg tudjuk határozni az aktuális pozíciókat. Az inerciális rendszerek hatalmas előnye, hogy amennyiben megkapják a kiindulási pozíciót, nincs szükségük külső információra, nem befolyásolják őket az időjárási körülmények és a mozgási adatokat adja, nem az elméletit. Ennek a hátránya, hogy mindig az előző pozícióhoz képest számolja ki a jelenlegit, így ha hiba kerül a számításba, akkor a hibák összeadódnak. Ezen úgy lehet javítani, hogy kombináljuk más rendszerekkel, amelyek pontosítják a helyzetünket. Ezeket a rendszereket széles körben használják, más féle közlekedési formáknál is egyre inkább elterjednek.

A madarak a földi mágneses tere alapján tájékozódnak. Kutatók bebizonyították, hogy a vándorló szárnyasok ideghártyáján olyan sejt található, amely a mágneses teret „látja”, így lehetséges az, hogy a madarak képesek tájékozódni és repülni akár több ezer kilométert is. A kutatók azt is belátták, hogy fekvés és legelés közben a szarvasmarhák és a gímszarvasok mindig észak-déli irányú pozícióban helyezkednek el, így valószínűleg ők is érzékelhetik a mágneses teret.



6. ábra - A Föld mágneses mezője

3.1. Műholdas helymeghatározás

Ha a műholdas helymeghatározást említjük, rögtön a GPS jut eszünkbe. Az amerikai kézben lévő NAVSTAR GPS (Global Positioning System) globális helymeghatározó rendszer egy, a GNSS (Global Navigational Satellite System) globális helymeghatározó és navigációs rendszer alrendszere. A GPS mellett még sok hasonló rendszer van alakulóban, mint az orosz

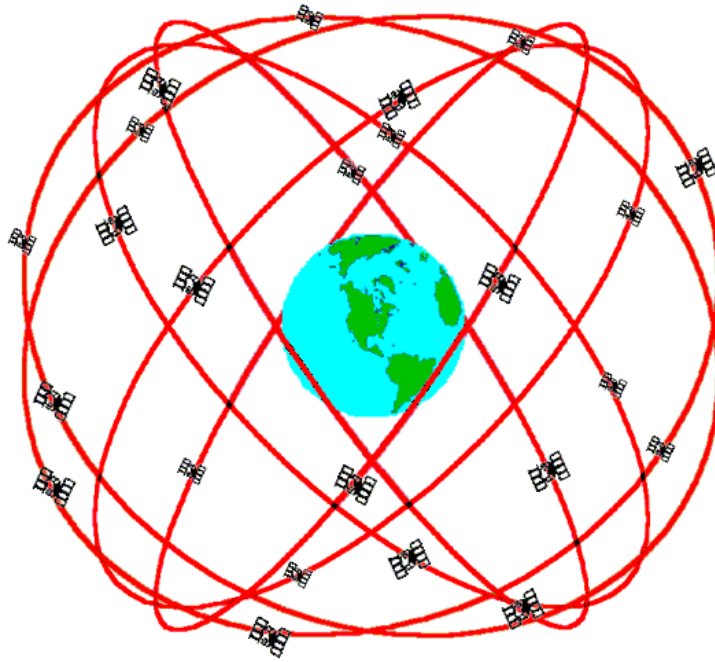
Glonassz és az európai Galileo és a Compass névre hallgató rendszert most építik ki a kínaiak. De napjainkban inkább csak a GPS-t használja mindenki. Az országok azonban függetlenedni akarnak az amerikaiaktól, mivel ők, ha gondolják akár szüneteltethetik is a műholdakkal való kommunikációt.

A GPS katonai céllal jött létre, de később bizonyos korlátozásokkal, például kisebb pontosság, ingyenesen hozzáférhetővé tették polgári felhasználásra is. A 24 darab fellőtt műhold 20200 kilométeres magasságban 6 különböző egyenletes eloszlású pályán, pályánként 4 darab kering a Föld körül. A műhold napelemeinek mérete 18 méter, várható élettartamuk akár 15 év is lehet, míg a kezdetekkor csupán 5 évet jósoltak nekik, súlyuk 2 tonna, teljesítményük 2 kW, keringési idejük 11 óra 56 perc. A rendszer fenntartásához a tönkrementek helyett újakat kell fellőni illetve a meglévőket fejleszteni.

A GPS-es helymeghatározás úgy működik, hogy 3 műhold távolságát mérjük a vevőtől, ezzel megkapjuk a vevő pozícióját. A távolságmérés oly módon történik, hogy a műholdba és a vevőbe is szerelnek egy nagy pontosságú órát. A műhold egy bizonyos időpillanatban kibocsájt egy rádiójelet, a vevő méri a saját órájával, hogy mikor érkezik be a jel, annak tudatában, hogy a rádiójel fénysebességgel halad. A vevő oldalon órahibával kell számolni, mert pontossága kisebb, mint a műhold órájáé, ezért egy negyedik műhold segítségével az óra hibáját korrigálhatjuk. A rádiójel kódokat és adatokat tartalmaz. Minden egyes műholdhoz különböző kód tartozik, így egyértelműen beazonosíthatóak. Többféle adatot sugároznak, például a műholdak helyzetét meghatározó pályaadatokat, a műhold órája és a földi atomóra közötti különbséget, valamint azt is, hogy melyik műhold működőképes és melyik nem, ugyanis, ha egy műhold nem üzemképes, akkor azt továbbítja a többi műhold felé is. A katonaság más típusú vevőkészüléket használ, amivel sokkal pontosabb helymeghatározás érhető el.

Az orosz műholdak folyamatosan meghibásodnak, csak a fele működőképes, így világméretben nem is terjedt el. Fejlesztésüket nem hagyják abba.

Az európaiak első műholdjukat 2005 decemberében lőtték fel és 2010-re tervezték, hogy teljeskörűen ki fog épülni a rendszer.



7. ábra - GPS műholdak elhelyezkedése a Föld körül [V]

A műholdas helymeghatározáshoz tartozik még két dolog. Az egyik a kiegészítő rendszerek, amelyek vagy geoszinkron műholdak vagy permanens állomások. Ezek a Földön helyezkednek el és fix pozícióval rendelkeznek, ők tudatják a műholdakkal a pontos pozíciójukat és azon adatok egy részét, amit a műhold a vevő felé kisugároz, ilyen a már említett műhold és földi óra közötti különbség. Kezdetben 5, ma 15 ilyen állomás létezik, melyekre az amerikai katonaság és a tengerészet felügyel.

A helymeghatározáshoz szükség van még egy eszközre, ami kommunikál a műholdakkal. A vevő egység megkapja a földrajzi szélességet, hosszúságot és a magasságot, a szatellitek jelerősségét. Egy úticélt megadva mutatja annak irányát és távolságát, illetve figyeli a mi irányunkat és azt is, tudja hogy milyen messze vagyunk a célunktól, annak köszönhetően, hogy másodpercenként frissíti az információkat. Ezen kívül még számos információt szolgáltat a felhasználói programtól függően. Ez lehet külön erre a célra gyártott berendezés vagy egy más célra is használható eszköz, amit csupán egy szoftverrel kell ellátnunk. Erre alkalmas lehet akár egy megfelelő mobiltelefon is. A mobilokba épített GPS vevőknek köszönhetően, az utóbbi időben csökkenni kezdett a hagyományos vevő eszközök eladása.

A GPS pontatlansága többféle forrásból eredhet, akkor is, ha feltesszük, hogy a vevő legalább 4 műholdat lát széles látókörben. Egyrészt a földi állomások pontatlanul közlik a műholdakkal a pályájukat, így a vevő felé tévesen közlik a műhold-vevő távolságot, szintén ugyanez a helyzet a műhold-óra pontosságának megállapításakor. Másrészt a légkör ionszférájában a rádiójelet ért ultrabolya sugárzás módosítja annak sebességét vagy épp a vevőkészülék nem a műholdjelet veszi, hanem más tárgyakról visszavert jelet.[9][10]

3.2. „Gépi látás”

A gépi látás technológiája napjainkban egyre inkább tért hódít. Ezen tudományterülethez tartoznak azok a rendszerek, amelyek önállóan képesek felismerni egy képen található alakzatokat. Azonban ezek intelligenciája messze elmarad az emberek idegrendszeren létrejövő látásától.

Általános esetre nem létezik algoritmus, de ha leírunk egy problémát és tudjuk, hogy mit keresünk, akkor algoritmusok segítségével megtalálhatjuk a keresett objektumot, illetve dönthetünk azzal kapcsolatosan, hogy szerepel-e a képen. Ehhez azonban rengeteg plusz információt tárolnunk kell (környezetéhez való viszony, térbeli elhelyezkedés), amelyek értékét feldolgozás útján csökkentjük, mivel ezzel a sok adattal nehéz számolnunk.

Gépi látást használnak a robot porszívók, amelyek kikerülik a bútorokat vagy épp a gazdáját felismerő robotkutya is. Valamint az orvosi képfeldolgozásban kísérletek zajlanak a készített képeken történő elváltozások, tumorok és daganatok automatikus felismeréséért. Gépi látást használnak a feladott levelek esetén a budaörsi Magyar Posta Zrt. által üzemeltetett Országos Logisztikai Központban is, ahol egy rendszer automatikus felismeri a nyomtatott és írott szövegeket, eldönti, hová kell továbbítani. Az iparban kamerák segítségével felismerhetőek a hibás termékek. Mint minden technológiában, a katonaság áll a legfejlettebb szinten, természetesen ezek nagy részét titokban tartják. De azt tudjuk, hogy létezik olyan rendszer, amely képes műholdak felvételeit autonóm módon kielemezni, valamint vadászpilóták a pilótát támogató automatikus célpont és repülőtér felismerés. Robotoknál kamerák segítik azok irányítását, ezért felismerhetővé és kikerülhetővé válnak az akadályok. Mára valósággá vált a népszerű „Knight Rider” nevű tv-sorozatban szereplő autó, amely képes emberi irányítás nélkül közlekedni. Először egy verseny keretében sikerült élesben tesztelni egy sofőr nélküli járművet, amely a Mojave sivatagban a 130 mérföldet 7 óra

alatt tette meg. Nemrég jelentette be a Volkswagen, hogy önműködő autók gyártásába fog. Az Audi TT-k feladata nem az, hogy állandó jelleggel maguktól közlekedjenek, inkább csak a veszélyes szituációkban vennék át a vezérlést. Probléma esetén pedig 20 mérföldes körzetben le lehet állítani a személygépkocsit. Az ennél jóval egyszerűbb sávelhagyási és követési távolságot figyelő rendszerek már eléggé elterjedtek.

Az egyik legnehezebb feladat a mozgó képen felismerni az alakzatokat. Ezt leginkább éldetektálással szokták megoldani. A gépi látást megvalósító algoritmust több részfeladatra kell felbontani:

- Előfeldolgozás
- Szegmentálás
- Feature detekció
 - Felismerés
 - Modell illesztés

Az előfeldolgozás során szűrők segítségével eltüntetjük a fotót leíró mátrixból a fölösleges információkat valamint a pixelek lényegtelen különbségeit. Szegmentálással a kép fölösleges részeit iktatjuk ki, ami minket nem érint. Ha bizonyos tárgyakat kell rögzítenünk fix közegben, akkor először tároljuk a háttér képét, így egyértelműen szegmentálhatjuk a felvétel minket érintő részét. Mivel ez túl általános eset, autonóm módon közvetlenül feldolgozni nem lehet. Ezért a következő lépésben a karakterisztikus jegyeket vizsgáljuk, a vizsgálandó célnak megfelelően. Általában vonalakat, köröket, geometriai formákat keresünk, ami lehet konkrét dolog is, mint szem, arc, daganat, rendszám-tábla vagy sok más hasonló dolog. A felismerés azt jelenti, hogy megtalálta az adatbázisában a felvételhez hasonló letárolt képet. A legtöbb képfeldolgozó rendszer tanítható, új képek bevitelével egyértelműbben azonosítja a tárgyakat. A modell illesztés dönt, a már algoritmizálással ismert kép megfelel-e a paramétereknek.

3.3. Inerciális rendszerek

Inertial Navigation System (INS) egy olyan navigációs rendszer, amely külső segítség nélkül, belső szenzorokkal képes a jármű sebesség, irány és pozíció meghatározására. Más külső navigációval kiegészítve szokták használni, mivel pontatlan eredményt ad és ezek a hibák halmozódnak és idővel már meglehetősen nagy az eltérés a valósághoz képest. Azonban időnként más, főként műholdas navigációval lekérve a pozíciót, megbízható rendszert kapunk.

Sőt bizonyos esetekben ki is egészítik egymást. Ha mondjuk alagútban utazunk és a járműből nem lehet műholdjeleket venni, akkor hasznosnak bizonyulhatnak az inerciális rendszerek. Használják repülőkön, hajókon, rakétákban, tengeralattjárókon.

A gyorsulásmérők és a giroszkópok vagy más mozgás érzékelők folyamatosan számolják az elmozdulás irányát és sebességét. A jármű aktuális pozíciója a mozgásérzékelők integrálásával kiszámítható. Ezeket részletesebben az érzékelők résznél ismertetem.

4. Érzékelők

4.1. Általános ismertetés

Az érzékelő olyan szerkezet, amely a mérendő mennyiségtől függő jelet bocsájt ki. Úgy is fogalmazhatunk, hogy bemenő jel a mérendő mennyiség, kimenő jel, pedig a szenzor által adott jel.

Több tulajdonság szerint jellemezhetjük őket, nézzük ezeket sorra.

- **Érzékenység:**
minél inkább követi a környezet változásait, annál jobb a helyzetünk, ugyanakkor minden más zavaró körülményt nem kell figyelembe vennünk ideális esetben. Az érzékenységet a következő képlet szerint írhatjuk fel:

$$S = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \frac{dy}{dx}$$

Szavakkal megfogalmazva ez azt jelenti, ha Δx megváltozik, mennyivel változik meg Δy .

- Milyen gyakran képes mintavételezni a mérendő jelet vagy jeleket.
- **Mérési tartomány:**
ha a mért értékek azon felső és alsó mérési értékek közé kerülnek, ahol az érzékenység megfelel a minimálisan elvártnak
- **Feloldás:**
érzékelő nem reagál a mérendő mennyiség elég kicsiny megváltoztatására. Lépcsős függvény esetén diszkrét érték lehet, a kimenő jel és a mérendő mennyiség kis mértékű változása nem jelenik meg a kimeneten.

- Hibafajták:

a különböző szenzoroknak különböző hibái lehetnek. Hiba akkor jelentkezik, amikor a mérendő és a mért mennyiségek nem egyeznek meg. A legelterjedtebb hibafajták a következők:

 - tranziens hiba:

a mért mennyiség nagymértékben megváltozik és az érzékelő ezt nem tudja követni
 - hiszterézis hiba:

érzékenységük nem megfelelő, csak késleltetéssel tudnak reagálni a mennyiségi változásokra
 - zaj:

az érzékelőn tapasztalható ingadozás, mely a szenzor működésében előálló hiba, nem pedig a mérendő mennyiség okozza
- Ideális érzékelő:
 - karakterisztikája lineáris, nullponti hibája zéró, érzékenysége megfelelően nagy, mérési tartománya és sáv szélessége végtelen, feloldása a nullához közelít, ennek megfelelően a környezeti változásokra azonnal reagál, nem rendelkezik a hibafajták egyikével sem.
- Reális érzékelő:

karakterisztikája nem vagy csak részben lineáris, feloldás, mérési tartomány és feloldás nem végtelen, hibák keletkezhetnek, érzékenységük változó.

A szenzorok lehetnek aktívak vagy passzívak. Az utóbbiak a természetes környezet fizikai jellemzőit érzékelik: hőmérsékletet, nyomást, fényt, hangot. Míg az előbbi kategóriához azok az eszközök tartoznak, amelyek maguk bocsátanak ki jelet és azok visszaverődését érzékelik: ultrahangos, lézeres távolságmérők, radarok, megvilágítást igénylő fényérzékelők, kamerák.[2]

4.2. Felhasznált szenzorok

Robotomat három különböző érzékelővel szereltem fel: egy gyorsulásmérővel, egy giroszkóppal és egy digitális iránytúvel.

Ezek a szenzorok napjainkban mindinkább kezdenek elterjedni. Egyes esetekben külön-külön alkalmazzák őket, máskor pedig kombinálva. Nagy technikai újítás a hordozható számítógépekbe szerelt gyorsulásmérő, amely észleli a leejtést, így még időben parkoló pályára tudja állítani a merevlemez író-olvasó fejeit, ezzel is védve a winchestert a nagyobb sérülésektől. Vagy épp a mobiltelefonokban lévő giroszkóppal a menü vagy alkalmazás annak megfelelően fordul el a készülékben, amilyen pozícióban tartjuk a berendezést. Illetve mindhárom kombinációja megtalálható már az iPhone-okban. Komplex rendszereket lehet kiépíteni gépkocsikban, kiegészítve más navigációs eszközökkel. Például, ha egy alagútban megyünk és megszűnik a GPS vétel, akkor előtérbe kerülhetnek ezek az eszközök, amiknek semmiféle külső kommunikációs adatcserére nincs szüksége, elegendő a Föld által kibocsájtott természetes mágneses jelek. Ezek a szenzorok alkalmasak megmérni az autó sebességét, mennyi távolságot tett meg ezen az úton és összesen, a jelenlegi ránk ható G-erő mértékét, vagyis igen sokféle mérési lehetőséget képes számunkra nyújtani. GPS-szel kombinálva, ha tudjuk a kiindulási pozíciónkat meghatározhatjuk a jelenlegit is.

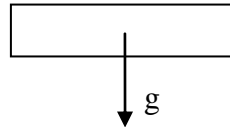
Feladatomban a Hitechnic cég által gyártott szenzorokat használtam. A vállalat 1986 óta gyárt LEGO robotok számára megbízható érzékelőket. Programozástanilag többféle nyelvet is támogatnak: Java, LabVIEW, RobotC. A Hitechnic olyan érzékelőket fejlesztett ki eddig, mint:

- gyorsulás/dőlés
- szín (pontosabb színérzékeléshez)
- iránytű
- elektro-optikai
- giroszkóp
- infra-kapcsolat (kommunikációhoz)
- infra-vevő (külső vezérléshez)
- infra-kereső (infra jel kereséséhez)

4.2.1. Gyorsulásmérő

A gyorsulásmérő egy olyan eszköz, amely egy vagy több tengely mentén méri a mozgás, lökés vagy rezgés relatív gyorsulását.

A gyorsulásmérőre hatással van a Föld gravitációs gyorsulása, ami kalibrációval kiküszöbölhető.



A gravitációs gyorsulás kiszámítható egy inga segítségével. Ha l az inga hossza, T a lengés időtartama, g a gravitációs gyorsulás, akkor:

$$T = 2\pi \sqrt{\frac{l}{g}} \Leftrightarrow g = \frac{4\pi^2 l}{T^2}$$

Nyugalmi állapotban egy testre 1 g erőhatás hat, szabadesés esetén pontosan zéró. Ha ezt vesszük az offset értékének és levonjuk a gyorsulásmérő értékéből, akkor a gravitációs gyorsulás értékét figyelmen kívül tudjuk hagyni a mérési feladatunknál.

A gyorsulás egy olyan vektor, amely mutatja a sebességváltozás mértékét egységnyi idő alatt. Jele: a , SI-beli mértékegysége: m/s^2 . Gyorsulás esetén a vektor a gyorsulás irányába mutat, lassuló mozgásnál, mozgással ellentétes irányú. Egyenletes mozgásnál a gyorsulás zéró. A klasszikus értelemben vett gyorsulás Newton II. törvényéből az erő(F) és a tömeg (m) alapján kifejezhető:

$$F = m \cdot a \Leftrightarrow a = \frac{F}{m}$$

Görbe vonalú pályán haladva a test sebességének nagysága és iránya is változik. A gyorsulásvektor a gyorsulás idő szerinti deriváltja, ahol v a sebesség, t az idő:

$$a = \frac{dv}{dt}$$

Ebből a mi esetünkben minden kifejezhető:

$$x = x(t), \quad y = y(t), \quad z = z(t)$$

$$v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt}, \quad v_z = \frac{dz}{dt}$$

$$a_x = \frac{dv_x}{dt} = \frac{d^2x}{dt^2}, \quad a_y = \frac{dv_y}{dt} = \frac{d^2y}{dt^2}, \quad a_z = \frac{dv_z}{dt} = \frac{d^2z}{dt^2}$$

A t_0 és t_1 időintervallum alatti sebességváltozást a következőképpen fejezhetjük ki:

$$v_x(t_1) - v_x(t_0) = \int_{t_0}^{t_1} a_x(t) dt$$

$$v_y(t_1) - v_y(t_0) = \int_{t_0}^{t_1} a_y(t) dt$$

$$v_z(t_1) - v_z(t_0) = \int_{t_0}^{t_1} a_z(t) dt$$

Egyenes vonalú egyenletes mozgás esetén, ahol $\frac{s}{t^2} = k$ egy konstans:

ha $\Delta t \rightarrow 0 \Leftrightarrow a = 2k$,

szabadesés esetén: $2k = g = a$, egyébként

$$a = \frac{\Delta v}{\Delta t} = \frac{2k(t + \Delta t) - 2kt}{\Delta t}$$

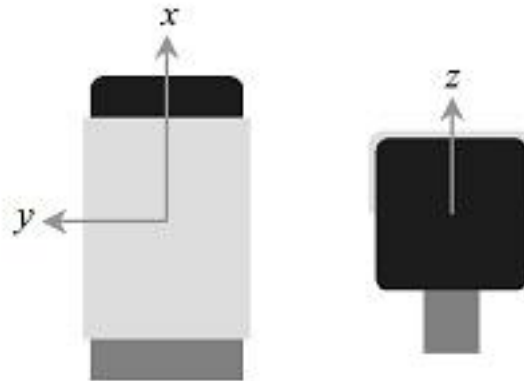
Napjainkban egyre több hordozható elektromos berendezésben használják, amely akár egy eszköz dőlésének iránya és mértéke alapján bemenetként is szolgálhat, ezt leginkább a szórakoztató elektronikában használják fel. Továbbá előszeretettel alkalmazzák járművekben, hogy értékeljék azok teljesítményét, menetstabilizátorban is van ilyen eszköz beépítve. Ezzel fel tudják mérni a motor és a fék teljesítőképességét, sőt a gépjármű stabilitását is, hogy mennyire rázkódnak az úton. De gyorsulásmérő figyelni a helikopterek rotorjának és egyéb eszközök forgó részeinek a fordulatszámát is. Használják még jármű ütközés vizsgálatánál és

többek között széllelkések mérésénél is, illetve az edzők játékosainak fizikai szintjük felméréséhez.

4.2.1.1. Hitechnic gyorsulásmérő

NXT Acceleration / Tilt Sensor (NAC1040)

Háromtengelyes gyorsulásmérő, mellyel a $-2g$ és a $+2g$ közötti mérési tartományban tudunk mérni. Az érzékelő a standard I2C kommunikációs portra csatlakozik. Az értékeit másodpercenként 100 alkalommal frissíti minden tengelyen. Szabványos Mindstorms burkolatot kapott, hogy összeköthető és építhető legyen más elemekkel is.



8. ábra - A gyorsulásmérő tengelyei [IV]

A következő táblázat tartalmazza, hogy melyik hexadecimálisban megadott memóriabeli címen milyen bájt típusú szenzor értékek találhatók.

Cím	Tartalom
42H	x tengely felső 8 bitje
43H	y tengely felső 8 bitje
44H	z tengely felső 8 bitje
45H	x tengely alsó 2 bitje
46H	y tengely alsó 2 bitje
47H	z tengely alsó 2 bitje

4.2.2. Iránytű

Az iránytű az ókori kínaiak által felfedezett navigációs eszköz, amely a Föld mágnes mezője alapján mutatja számunkra a megfelelő irányt. A hagyományos, analóg iránytű egy mutatót tartalmaz, ami a tengelye mentén elfordulva mindig az észak-déli irányt mutatja, a mutató északi irány felé néző részét külön jelzéssel látják el, általános esetben piros színűre van festve. Ahogy telt az idő, az eszköz sokkal megbízhatóbbá vált. A tájoló annyiban különbözik az iránytűtől, hogy nem csupán a főbb irányokat mutatja, hanem rendelkezik fokbeosztással is valamint folyadékot használnak a belsejében, ami csökkenti a tű hirtelen kilengését. Az iránytűt váltotta fel a GPS (Global Positioning System).

A mágneses iránytűk problémái, hogy fémek közelítése esetén kitérnek a helyes irányból, mivel módosítják az iránytűt körbevevő magnetikus teret. A mágnes északi és a földrajzi északi pólus nem egyezik meg. A földrajzi Északi-sark az a pont, amely a legközelebb található a Föld forgástengelyéhez, míg a mágneses tér a mágneses dél irányából északi irányába mutat, nagyjából 700 km, azaz $11,3^\circ$ eltéréssel (deklinációval) a földrajzától, ráadásul folyamatosan vándorol.

A digitális iránytűket használatuk előtt kalibrálni kell. Ha kalibráláskor a mágneses mezőre ható tárgyak vannak a közelében, akkor csak úgy működnek helyesen, ha használatukkor mindvégig ugyanott maradnak. Hatással lehet az eszközre akár egy TV, egy számítógép vagy egy mobiltelefon is. Kalibráláskor felméri az eszköz a körülötte lévő mágneses teret és így határozza meg az irányokat.

Digitális iránytű napjainkban már megtalálható egyes típusú órákban, mobiltelefonokban, néhány egyéb elektromos eszközben.

4.2.2.1. Hitechnic iránytű

NXT Compass Sensor (NMC1034)

Digitális iránytű, ami méri a Föld mágneses mezőjét és ez alapján visszatér egy 0 és 359 közötti számmal, annak megfelelően, hogy a szenzor milyen irányban áll. Az adatok másodpercenként 100-szor frissülnek. Két üzemmódban képes működni. Egyik az olvasási, másik a kalibrálási. Olvasási módban mindig az aktuális irányzékot számolja és szolgáltatja, míg kalibrálási módban korrigálni lehet a külső mágneses mező anomáliákat, ami saját

munkámat tekintve a motorok és elemek körül létrejön, ezzel pontosítva a visszakapott adatokat. Hogy minimálisra csökkentsük az eltérés mértékét ajánlott 10-15 cm távolságra elhelyezni az érzékelőt a szervomotoroktól és az NXT központi egységétől. A kalibrálás alatt a robot lassan és szűk ívben megtesz egy kört. A kör befejezése után a kalibráció befejeződik. A beállítások megőrződnek, ha közben még nincs is csatlakoztatva az érzékelő, egészen a következő kalibrálásig. Az iránytű is a standard I2C kommunikációs portra csatlakozik és a gyorsulásmérőhöz hasonlóan szabványos Mindstorms burkolatot kapott.

Az alábbi táblázat tartalmazza a szenzor regiszter kimeneteit és típusait.

Cím	Típus	Tartalom
00-07H	sztring	Érzékelő verzió száma
08-0FH	sztring	Gyártó
10-17H	sztring	Érzékelő típusa
18-3DH	bájt	Nem használt
3E, 3FH	sztring	Foglalt
40H	bájt	Nem használt
41H	bájt	Üzem mód
42H	bájt	Irány – másodlagos
43H	bájt	Irány – elsődleges
44, 45H	word	Irány (alacsony bájt, magas bájt)
46-7FH	bájt	Nem használt

- Érzékelő verzió szám: n.m alakban jelenik meg, ahol n a fő verziószám, m pedig a revíziószám.
- Gyártó mező tartalmazza a „HiTechnic” bejegyzést.
- Érzékelő mező tartalmazza a „Compass” bejegyzést.
- Üzem mód mezőt 0x00-ra mérésre, 0x43-ra pedig kalibrálásra állíthatjuk a szenzort
- Irány mező visszatér mérés üzemmódban az aktuális mérési eredménnyel, kalibrálás módban nem elérhető. 0x42 tartalmazza a másodlagos irányt, 0x43 az elsődleges irányt. Az irányt ki lehet számítani a következő módon:

$$\text{Irányzék} = (\text{másodlagos} * 2) + \text{elsődleges}$$

Ha problémás a jel, akkor a 0x44 az alacsony, 0x45 a magas bitértékű 16-os számrendszerbeli számot tartalmazza.

- A 0x41 mező alapértelmezett értéke a 0, ami a mérés üzemmódot jelenti. Be kell állítani a 0x43-at kalibrálási módba, ha vége van, akkor pedig vissza, 0-ra.

Az iránytű akkor ad helyes értéket, ha vízszintes síkban áll.

4.2.3. Giroszkóp

A giroszkóp a szögsebességet méri. Ha integráljuk, akkor megkapjuk a teljes szögelfordulást, így mindig tudjuk, hogy milyen irányban vagyunk a kiindulási pozícióhoz képest.

4.2.3.1. Hitechnic giroszkóp

NXT Gyro Sensor (NGY1044)

A giroszkóp, egy olyan érzékelő, mely egy tengely mentén észleli a forgást. Olyan számmal tér vissza, ami azt mutatja, mennyi a másodpercenkénti fordulat fokban kifejezve. Maximum +/- 360°/másodpercig képes mérni. Ezt a hardverelemet használják az egyensúlyozó robotok is. A szabványos szenzorportra csatlakozik, mint az előzőeknél tapasztalhattuk. A forgást másodpercenként 300-szor képes lemérni. Az offszet érték kiszámításához a szenzornak és így a robotnak is mozdulatlan, álló helyzetben kell lennie. Az offszet érték helyes megállapítása után ezt ki kell vonnunk a giroszkóp aktuális értékéből. Ha az eredmény mozdulatlanság esetén zéró, akkor megfelelő az offszet értékünk. Függőleges síkban való méréshez a szenzort fekete végével felfelé kell elhelyezni az NXT-n.[4]

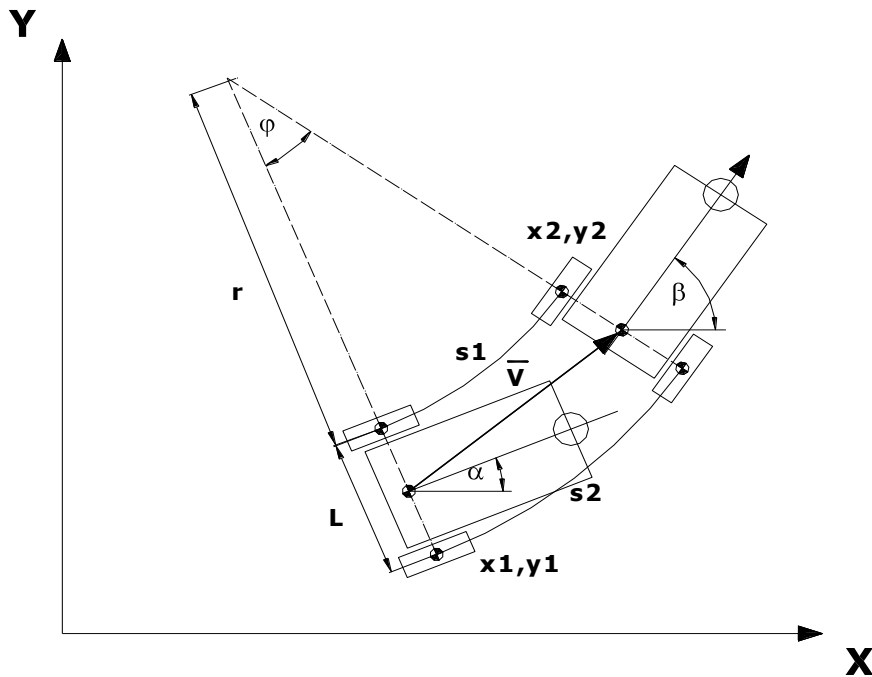
5. A projekt

5.1. Projekt ismertetése

Feladatom célja az volt, hogy méréseket végezzek egy LEGO Mindstorms NXT robottal és a már ismertetett Hitechnic cég gyorsulásmérőjének, giroszkópjának valamint digitális iránytűjének segítségével, amiket aztán feldolgozok. Ezen eszközök felhasználásával olyan programot készíteni, amellyel meg tudom mondani a robot irányát és gyorsulását.

A robot vezérlésénél előforduló egyik probléma az, ha elforgatom egy „x” fokkal, akkor valószínűleg a végeredmény nem az optimális „x” fokos elfordulást fogja adni. Köszönhető ez

annak, hogy a kerék kissé megcsúszik, leginkább a fordulás kezdetekor és befejezésekor, mivel ekkor változik a motor sebessége. Az sem mindegy, hogy milyen gyorsan szeretnénk kanyarodni, nagyobb kanyarodási sebességnél nagyobb a pontatlanság esélye és persze a felület is problémás lehet, csúszósabb felületen szintén nem kapunk pontos eredményt. Ezen esetekkel a hétköznapi életben is bárki találkozhat, gondoljunk csak télen a jeges útra, sokkal óvatosabban adunk gázt és fékezünk.



9. ábra - A robot elfordulása

Képletekből kiszámítható, hová kellene a robotnak elméletileg érkeznie, a betűk jelzései leolvashatóak az ábráról:

$$\varphi = \frac{s1 - s2}{L}$$

$$\beta = \alpha + \varphi$$

$$r = \frac{s1}{\varphi}$$

$$|\vec{V}| = 2 \cdot \left(r + \frac{L}{2} \right) \sin \frac{\varphi}{2}$$

$$x_2 = x_1 + |\vec{v}| \cdot \cos\left(\frac{\varphi}{2} + \alpha\right)$$

$$y_2 = y_1 + |\vec{v}| \cdot \sin\left(\frac{\varphi}{2} + \alpha\right)$$

Természetesen az egyenletes vonalú gyorsuló mozgást is ki tudjuk számolni. Esetemben egyszerű volt a dolgom, mert a kerékre a motorerőt áttétel nélkül vittem át, így a kerék is ugyanannyit forgott, mint a motor.

A sebesség-idő kiszámítása:

$$v(t_2) = \int_{t_1}^{t_2} a(t) dt + v(t_1) = a(t_2 - t_1) + v(t_1) \Leftrightarrow \Delta v = a \cdot \Delta t$$

Az elmozdulás a következőképp számítható, ahol a jelenti a gyorsulás:

$$r(t_2) - r(t_1) = \int_{t_1}^{t_2} (at + v_0) dt \Leftrightarrow s = \frac{1}{2} at^2 + v_0 t$$

Mint említettem ez csak ideális esetben van így, de a készített programot felhasználva meg tudom állapítani mennyire volt pontos a forgás, mérési adatokból statisztikát és következtetést tudok levonni.[1]

5.2. A futtatókörnyezet

A program Java nyelven íródott, annak is a JDK 6 Update 18-as verzióját felhasználva. Kiegészítve a specifikus NXT robotokra szánt Java csomaggal, amit LeJOS-nak hívnak. Projektemet, ennek a jelenlegi legfrissebb, 0.85-ös verziójával fejlesztettem. A LeJOS tartalmazza LEGO robotokhoz szükséges API specifiációt és csomagokat, amelyek a classes.jar fájlban vannak elhelyezve. Használatához le kell cserélni, az eredetileg az NXT-n futó LEGO Mindstorms firmware-t LeJOS-ra, így mostantól egy új operációs rendszer fog futni az eszközünkön, ami magában tartalmaz egy kis virtuális gépet, amely futtatni fogja a java nyelvű bájtkódjainkat. Ezen kívül tartalmaz néhány hasznos eszközrendszert, mint például a hibakereső, firmware frissítő, program feltöltő modul, valamint tartalmaz egy pár alapvető mintaprogramot.

Számos pozitív oldala van, ami megkönnyíti a fejlesztést:

- teljes Bluetooth támogatás
- több platformos támogatás: Windows, Linux, Mac OS X
- támogatja az Eclipse és a Netbeans IDE fejlesztői környezet néhány funkcióját
- eseményvezérlést
- többszálúságot
- pontos motorvezérlést
- könnyű LCD kezelést
- matematikai és más szabványos javabeli függvények és osztályok támogatása
- érzékelők támogatása
- adatfolyam kezelés szabványos Lego kommunikációs vonalakon
- és még számos más előny

Projektemet Windows XP SP3 operációs rendszer alatt fejlesztettem.

Ahhoz, hogy java nyelvű programot írjunk, fordítsunk és töltsünk az NXT-re, három dologra van szükségünk:

- Java JDK (Java Development Kit)
- LeJOS
- LibUsb driver, az USB kezeléséhez

Programíráshoz a Netbeans integrált fejlesztői környezetet használtam. Ez alaphelyzetben nem ismeri a specifikus osztályleírásokat, így először le kell tölteni a LeJOS plugint a Netbeans számára, majd meg kell adni a projekt beállításainál a LeJOS `'\lib'` könyvtárát. A Netbeanst azonban csak szövegszerkesztésre használtam, hogy jelezze a szintaktikai hibákat, fordításhoz már a Windows parancssorát használtam, mert a Netbeans csak a class fájlokat állítja elő, a robot számára futtatható `nxj` kiterjesztésű fájlt nem.

Fordításhoz elsősorban be kellett, hogy állítsam a környezeti változókat:

- JAVA_HOME: a JDK telepítési könyvtára
- NXJ_HOME: LeJOS telepítési könyvtára
- PATH könyvtárhoz hozzá kell adni a JDK és a LeJOS `'\bin'` könyvtárát

Amennyiben ezzel megvagyunk, gond nélkül mehet a fordítás. Ehhez először meg kell nyitnunk egy parancssort és a forrás fájlunk könyvtárába lépni. A következőkben ismertetek néhány fontos parancsot:

- `nxjc <java-fájl>.java:`
lefordítja és előállítja a class fájlt
- `nxjlink -v <class-fájl> -o <bináris>.nxj:`
a class fájlt összelinkeli a már említett `classes.jar` fájlal és előállítja az `nxj` kiterjesztésű állományt
- `nxjupload <bináris>.nxj:`
feltölti a robotra a programot. Paraméterként megadhatjuk, hogy kábelen vagy usb-n szeretnék feltölteni, valamint a cél címet is. Elsődlegesen USB kábelen próbálkozik, ha itt nem észlel kapcsolatot tovább próbálkozik Bluetooth-on is.
- `nxj <class-fájl>:`
egyesíti az `nxjlink` és `nxjupload` parancsokat, `-r` paraméterrel feltöltés után automatikusan futtatja a programot az NXT-n.

Eleinte az `nxjupload` parancsot használtam. Később más megoldást kerestem, mivel a program feltöltése elég időigényesnek bizonyult. Helyette az előbb említett `nxjbrowse java` kisalkalmazást használtam. Ennek megnyitása nálam nagyjából 3 percig tartott, de ezt csak egyszer kell meghívni és amíg fut, akárhányszor feltölthetünk és letölthetünk fájlokat a segítségével, ami 1-2 másodperc alatt végbemegy.[3][5]

5.3. A program bemutatása

Mielőtt elindítanánk a programot a pontosabb számítás érdekében ajánlott beállítani a következő paramétereket: „WHEEL” konstans értéke tárolja a kerék átmérőjét, a „TRACK” konstansban kerül letárolásra a bal kerék közepétől a jobb kerék közepéig mért tengelytávolság.

Működését tekintve a program először tesz egy kört szűk ívben, eközben beállítja az érzékelőket és elvégzőz olyan műveleteket, amelyek a további futáshoz szükségesek, valamint létrehoz egy fájlt. A továbbiakban megtesz még egy kört, ekkor kezdődik a mérési folyamat. Az eredményeket az előbb létrehozott „SensorData.log” állományban rögzíti. Mindeközben az LCD kijelzője mutatja az értékeket. A kör megtétele után egyenesen halad tovább. Az érzékelők olvasása és a robot mozgása az ESCAPE gomb megnyomásával ér véget.

Amennyiben a letárolt adatokat tartalmazó fájlt meg szeretnénk tekinteni, csatlakoztassuk az USB kábelt az NXT robot megfelelő portjába és a már említett nxjbrowse segítségével válasszuk ki a fájlnkat és töltsük le a számítógépünkre.

A fájlban lévő adatok azonos mérési eredményei egymástól szóközzel vannak elválasztva, a különböző mérések pedig új sorba kerülnek. Microsoft Office Excelbe importálva azonnal meg is kezdődhet az adataink feldolgozása.

A program rendszerkövetelménye:

- LEGO Mindstorms NXT csomag LeJOS operációs rendszerrel
- 1 darab giroszkóp, megfelelő portra csatlakoztatva
- 1 darab gyorsulásmérő, megfelelő portra csatlakoztatva
- 1 darab digitális iránytű, megfelelő portra csatlakoztatva
- Java SDK
- USB porttal rendelkező személyi számítógép

A projekttem három saját osztályból áll. Az egyik vezérli a robot mozgását (Movement), a másik végzi a szenzor műveleteket, háttérszámításokat, és az állomány létrehozását, a harmadik osztály a főosztályunk (Sensor), ami nem csinál semmi mást, csupán elindítja az előbbi két osztályunk 1-1 példányát megfelelő időben. Ezek a példányok külön-külön szálon futnak, ugyanis míg ez nem így volt, a robot vagy mérni, vagy menni tudott, egyszerre a kettőt nem, így viszont párhuzamosan képesek dolgozni.

Nézzük ezen osztályokat részletesen. Kezdjük a SensorMeasure osztállyal.

A giroszkópom GyroSensor (1. port), gyorsulásmérőm TiltSensor (2. port), az iránytűm CompassSensor (3. port) típusú objektum, neveik sorban gyroRead, accelRead, compassRead.

Mindhárom szenzorleíró osztály a lejos/nxt/addon könyvtárban található meg, valamint mindháromnak konstruktorként át kell adni, hogy melyik portba vannak bekötve.

A szál run metódusának legelső dolga, hogy létrehozza a fájlnkat. Amennyiben az már létezik, törli és újból létrehozza. A fájlba íráshoz a FileOutputStream-et használom. Az állomány egy sora 6 szóközzel elválasztott adatértékből áll, amelyek a következő sorrendben tárolódnak el: gyorsulásmérő x, y és z tengelyének értéke, giroszkóp és az iránytű aktuális értéke, valamint a rendszerből lekért idő milliszekundumokban megadva. A biztonság kedvéért az adatsorokat fejléccel is elláttam, így nem lehet összekeverni őket.

A szenzor adatai egységesen float típusúak, ezeket egy „str” nevű tömbbe átkonvertálom sztringgé a String.valueOf(érték vagy kifejezés) függvénnyel, ezeket folyamatosan kiírom az LCD kijelzőre, hogy azok értékeit folyamatosan figyelemmel lehessen kísérni. A különböző sztring változóból csinálom egy text sztringet a saját appendToString függvénynek, és ezt a text-et adom át az appendToFile metódusnak, ami meghívja a getBytes függvényemet, mely a karaktereket bájt típusú számra konvertálja, ugyanis a FileOutputStream osztály write metódusa bájtokkal tud dolgozni:

```
static private byte[] getBytes(String inputText){
    byte[] bytes = new byte[inputText.length()+1];
    for(int i=0;i<inputText.length();i++){
        bytes[i] = (byte) inputText.charAt(i);
    }
    bytes[inputText.length()] = 0;
    return bytes;
}
```

A roboton lévő kilépés gomb megnyomásával a fájlt le is zárom, a fájlműveleteknél szükséges kivételkezelést is teljesítem. Sajnos a fájl méretének vagy egy korlátja, mert az NXT viszonylag kis memóriával rendelkezik, azonban így is több ezer sornyi adatot képes letárolni. Az osztály végén lévő sleep függvény segítségével állíthatjuk be, mennyi ideig várjon a következő méréssel. Ha ez az időérték túl kicsi, kevesebb ideig tudunk mérni, mivel adott időegység alatt több adatot kapunk, ebből következik, hogy hamarabb megtelik a memória. Ellenkező esetben ritkábban mintavételezünk, így nem tudjuk reálisan követni a változásokat. Az állomány kezelését ezzel be is fejeztem.

Az egyik legfontosabb alkalmazott osztály, ami elég intelligensnek bizonyult és nagyban megkönnyítette a munkámat a CompassPilot. Konstruktorra sorrendben a következő: (iránytű, kerékátmérő, kocsi szélessége, bal motor, jobb motor, [fordítva]). A legelső paraméterben a létrehozott iránytű példányát kell átadni. A kerék átmérője rá van nyomtatva a robot kerekére. A kocsi szélessége a bal kerék közepétől a jobb kerék közepéig mért távolság. Mind a kerékátmérő, mind a kocsi szélességének mérőszáma tetszőleges, vagyis mindegy milyen mértékegységben adjuk meg, de a kettőnek meg kell egyeznie, mert ezeket csak arányszámok lesznek.

Az osztály a következő képlet alapján valósítja meg a robot forgását:

$$\frac{\text{kocsi szélessége}}{\text{kerékátmérő}} \cdot \text{szög}$$

Ebből a képletből látható, hogy tényleg érdektelen, hogy mm-ben, cm-ben, hüvelykben vagy valamilyen egészen más egységben van megadva. Azonban ezek megadása minél pontosabb értéket követel meg, mert már nagyon kicsi eltérés is látható hibát okozhat egy idő után. Még ha jól is adjuk meg a mérőszámokat, előfordulhat köszönhetően a robot felépítésnek, hogy a LEGO elemek kissé szétszúznak vagy a kerék a motorral összekötött építőeleméről kissé távolabb kerül az eredeti pozíciójától, természetesen ezen esetekben is csökken a hitelesség mértéke.

Visszatérve a konstruktorra a bal és a jobb motor értelemszerűen a bal oldali és jobb oldali motorok, ha felcserélve adjuk meg, akkor annyi történi, hogy a másik irányban fog elfordulni a robot. A fordítva mező opcionálisan megadható logikai érték, ha értéke igaz, akkor a robot abban az esetben megy előre, ha azt mondjuk a motoroknak, hogy hátra forduljon, tehát fordított logika fog érvényesülni.

A szenzorok által mért értékeket egyszerűen 1-1 beépített függvénnyel lekérhetem, azonban mint már említettem, hogy pontosak legyenek az adataink a szenzorokat kalibrálnunk kell. A gyorsulásmérő metódusai között találtam egy getData függvényt, ahol paraméterben meg kell adnunk a kezdő regisztert, a buffert és a buffer méretét. Ezek után követtem a gyáró oldalán található leírásokat. Giroszkóp beállítása esetén offszetet számolok. Ezt úgy teszem, hogy a program elején mérek az érzékelővel 50 értéket, ezt átlagolom és kerekítem, formálisan így néz ki:

```

public static void calculateOffset() {
    offset = 0;
    double offsetSum = 0;
    for (int i = 0; i < 50; i++) {
        offsetSum += (double) gyroRead.readValue();
        try {
            Thread.sleep(4);
        } catch (InterruptedException e) {
            LCD.drawString(e.getMessage(), 2, 0);
        }
    } offset = Math.ceil(offsetSum / 50);
}

```

A kapott számot levonom a giroszkóp értékéből és így a helyes értéket kell kapnom. Integrálásához két változóban tárolom az aktuális időt milliszekundumban, az első időt a kalibráció végén kérem le(sTime), de ez csak egyszer hajtódik végre. A másik időt ciklikusan minden mérés elején(eTime). Az sTime-nak minden mérés végén átadom az eTimeot, az idők különbségét átváltom másodpercre az 1000-rel való osztás segítségével. A kapott szám a két idő különbsége a mérések között eltelt idő. Az integrálás kódja:

```

if (first==true){
    sTime=System.currentTimeMillis();
    gyroIntegral=0;
    first=false;
}
else{
    eTime=System.currentTimeMillis();
    getGyroValue();
    gyroIntegral=gyroIntegral+(eTime-sTime)/1000*gyroValue;
    sTime=eTime;
}

```

A digitális iránytűt az osztály intelligens metódusa kalibrálja helyettem, csupán a calibrate() parancsot kell kiadnom, ekkor a robot megtesz egy szűk ívű kört lassan, hogy

megnézze milyen mágneses mező veszi körül. Ezzel a kalibrációnk befejeződött, amit a program a tuned változó igaz értékűre állításával jelzek.

A kijelzőre megpróbáltam minden munkafolyamatot kiírni (kalibrálás, mérési adatok), ez segítségemre is volt. Ha elakadtam, akkor a kiírásnak köszönhetően mindig tudtam hol történhetett hiba, illetve a mérési adatokat is tudtam ellenőrizni, ugyanis nagyjából lehet látni az értékeken, hogy mennyire reálisak. Az LCD-re írás rendkívül egyszerűen megy, csupán meghívtam az LCD.drawString(„sztring”, x koordináta érték, y koordináta érték) függvényt.

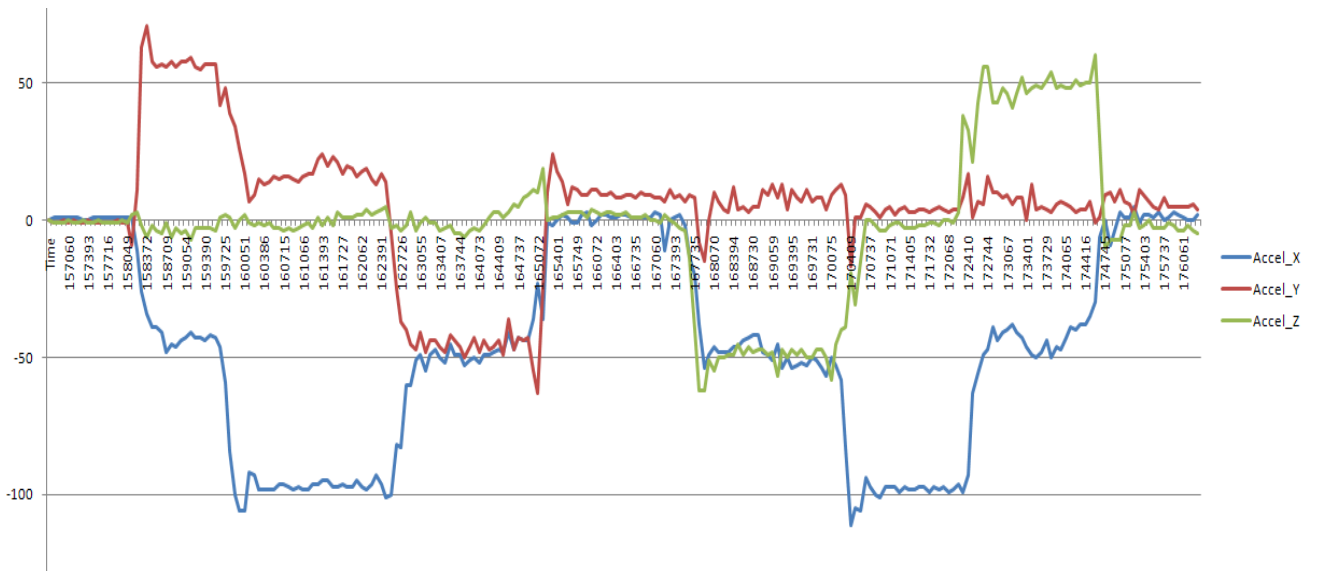
Movement osztály:

Ebben az osztályban valósítom meg a robot mozgását. A SensorMeasure osztályban alkalmazott CompassPilot-hoz hasonló TachoPilot beépített osztályt használom fel, annyi különbséggel, hogy itt nem kell és nem lehet megadni az iránytűt paraméterként. A többi paraméter megegyezik a CompassPilot-nál leírtakkal. A kerék átmérőjének és a tengelytávolság méreteit a másik osztályból kérem le, így az értékek ugyanazok lesznek, ezzel elkerülve a többszörös adattárolás és az ebből eredendő ellentmondást. A robot mozgását, csak akkor indítom el, ha a szenzorok kalibrálása megtörtént. Először elfordítom a robotot az osztály elején változóban megadott fokkal. Tesztjeimnél 360 fokot adtam meg, ami egy teljes fordulat. A kalibrálásnál is megtett már egy kört, következésképp egymás után két kört fog leírni a robot. A másodiknál már aktív a mérési folyamat. A giroszkóp integrálja az értékeit, amelyet folyamatosan összesít, így a kör megtétele után meg tudjuk mondani, hogy a megadott fok érték helyett mennyit fordult valóságban a robot. Miután a folyamat befejeződött nagyon rövid ideig leáll a motor, majd a robot elindul egyenesen, majd ismét megáll, addig folytatódik a folyamat, míg meg nem nyomjuk a robot kilépés gombját. Minden elindulásnál magasabb fordulatszámra vált a motor, egészen addig, míg el nem éri a 600-as sebességet, ekkor lecsökken 250-re. Majd ismét elkezd növekedni a sebesség. A maximális sebesség 900, de ezt csak teljesen feltöltött, 9V-os teleppel tudja produkálni. A sebességváltásra azért volt szükség, hogy össze tudjam hasonlítani a különböző mértékű gyorsulásokat.

A szenzor osztály a főosztályunk, ez tartalmazza a main metódust, és ezt kell nxj-be lefordítani és feltölteni a robotra.

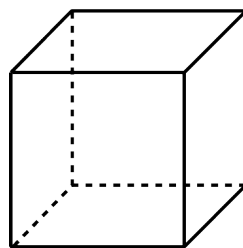
A méréseket elvégezve következtetéseket tudtam levonni ezek eredményeiből. Rájöttem, hogy a gyorsulásmérő rendkívül érzékeny, már apró rezdülésekre is kitérnek az értékei. A

következő diagramokon láthatjuk a gyorsulásmérők viselkedését. Az x tengely jelzi az előre, az y az oldalra és a z a függőleges irányú gyorsulást. A diagram vízszintes tengelyén feltüntetve látszik, mely időpillanatban történt a mérés, az időértékeket milliszekundumokban kérem le a `System.currentTimeMillis()` függvénnyel.



10. ábra - Gyorsulásmérő kalibrálása

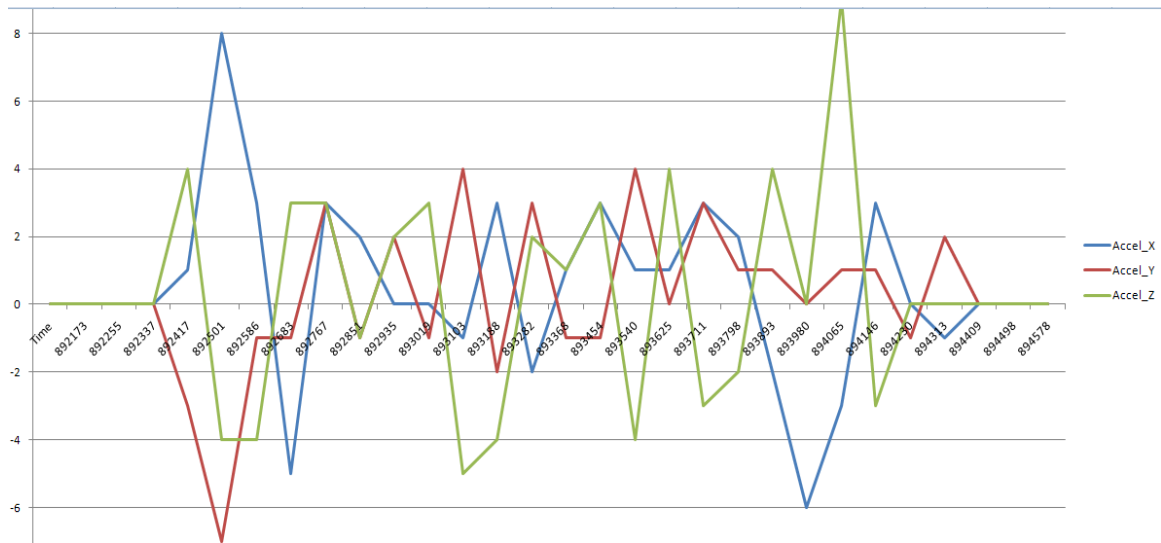
A gyorsulásmérő hitelesítésének legjobb módja, ha diagramon megjelenítjük a kapott adatokat. A kalibráció helyességét úgy ellenőrizhetjük, hogy a tér hat pozíciójába, tengelyenként két, egymással ellenkező irányba forgatjuk a szenzort. Ezek az irányok megfelelnek a kocka hat oldalának.



A grafikont megvizsgálva becsléssel is meghatározhatjuk a gyorsulás tényleges zéró értéket. Az alábbi táblázatban átlagolás alapján pontosan meghatároztam ezeket a számokat. Látható, hogy a tengelyeknek különbözik a nullpontjuk, függetlenek egymástól.

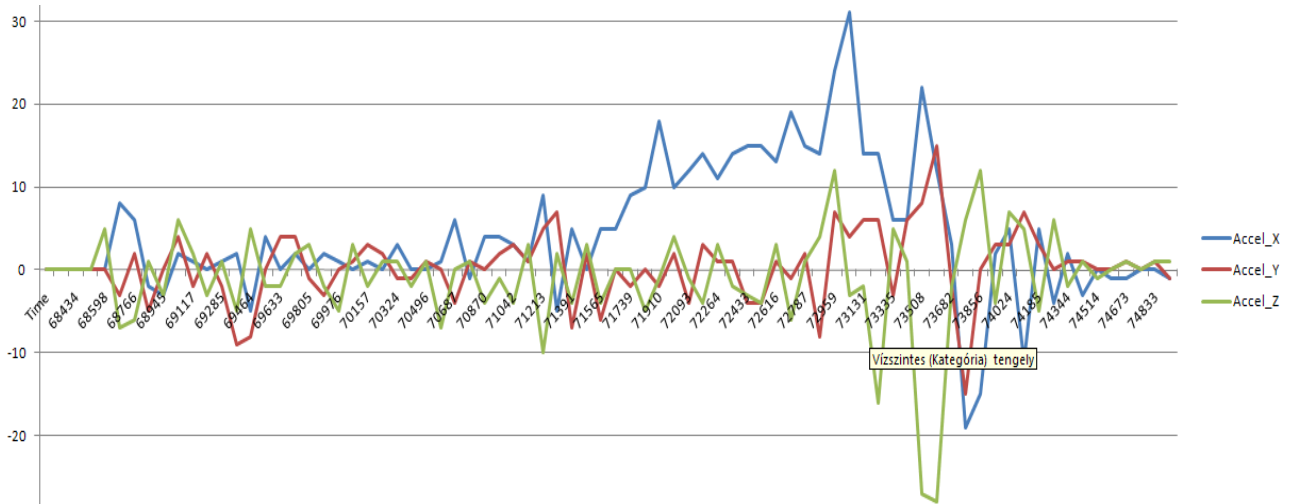
x	y	z
-45,6949	4,7288	-1,7331

A következő grafikon a robot gyorsulását szemlélteti.



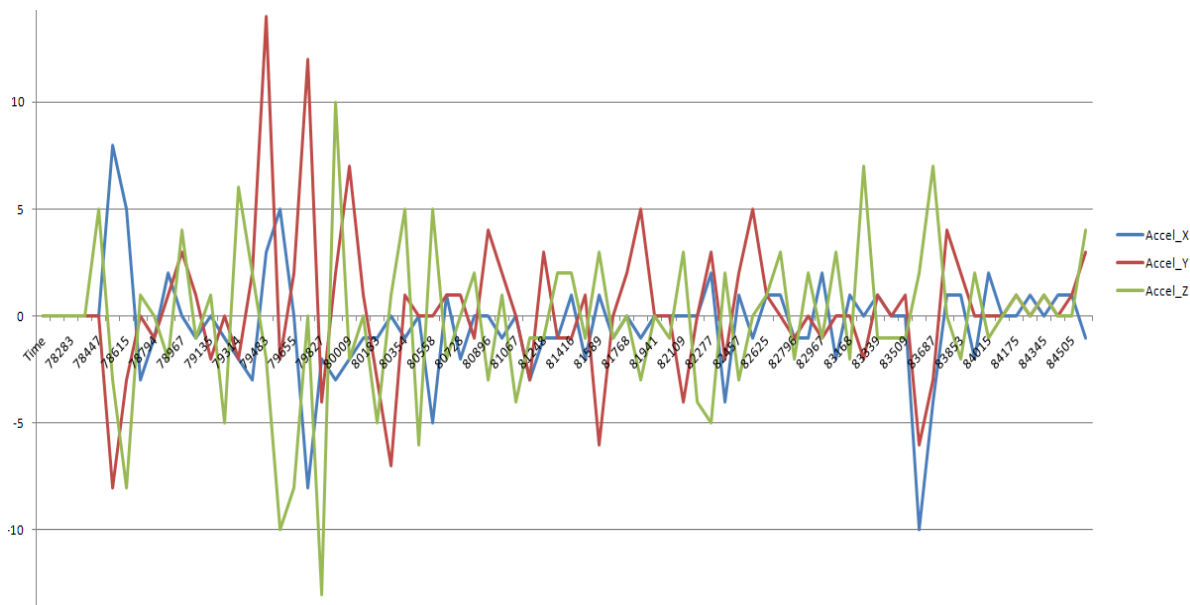
11. ábra - Egyenes vonalú gyorsulás grafikonja

Első ránézésre azt mondhatjuk, hogy össze-vissza kitérnek az értékek, de a sokadik mérés után tapasztalhattam, hogy a LEGO robot túl könnyű, így a motor sebességváltoztatásának hatására a robot megrezdült, illetve olyan irányokban is elmozdult kis mértékben, amely felé nem lett volna szabad. Leginkább a gyorsulási folyamatnál valamint megállásnál tapasztaltam, oly módon viselkedett a robot, mint egy autó, amikor erős gázt vagy féket használunk és érezzük, hogy belepréselődünk az ülésbe vagy előre dőlünk. Esetünkben ez a z tengelyre hat.



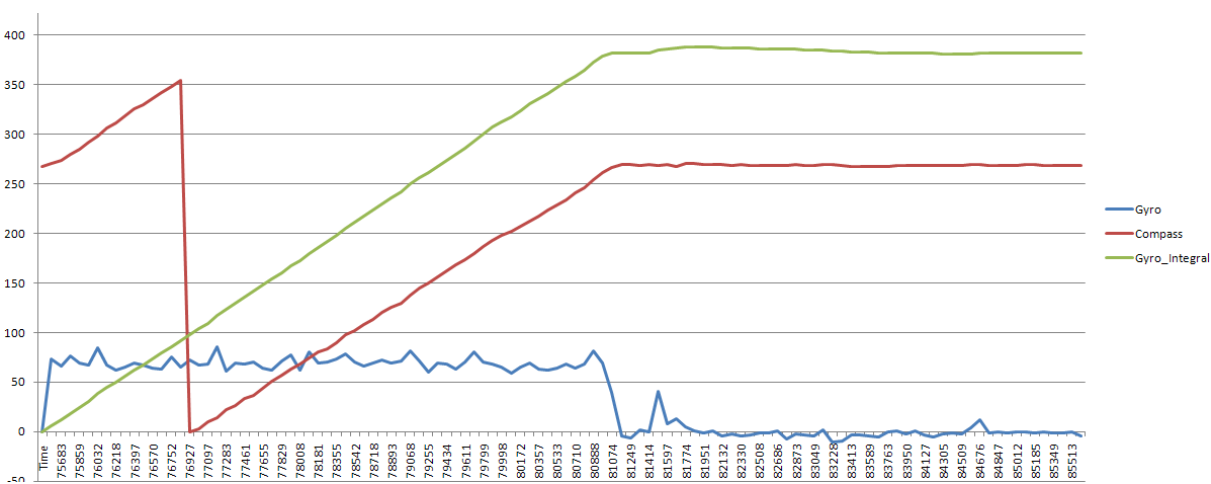
12. ábra - Grafikon a robot emelkedőre feljutásáról

Ez a mérés egy hozzávetőlegesen 16°-os emelkedőre felmenet készült. Minél inkább növeltem a szöveget, annál nagyobb nem várt kitérések mutatkoztak. Az emelkedő kartonpapírból készült, tetején a karton egy hajtásánál kissé meggyűrődött, egy „fekvőrendőr” alakult ki. Ez a grafikon végén lévő értékekből kiolvasható.



13. ábra - Grafikon a robot egyik kereke alatti akadályok szemléltetéséről

A robot egyik kereke alá tett akadályok (tollak) hatására valamennyi tengely mentén a gyorsulásmérő nagy kiugrásokat mutat.



14. ábra - Az iránytű, a giroszkóp és a giroszkóp integrál értékei

Az ábráról leolvasható, hogy a gyorsulásmérővel ellentétben az iránytű és a giroszkóp esetén nincsenek kiugró értékek. A grafikon első fele jelzi a fordulást, második fele, ahol az

értékek nem változnak lényegesen az egyenes vonalú gyorsulás mutatja. A giroszkóp értéke fejezi ki a másodpercenkénti fordulatszámot, ami szemmel láthatóan azonos a kör megtétele alatt. A giroszkóp integrálja azt számolja, hány fokot fordult a robot az eredeti állapothoz képest, az értékeket ábrázolva egyenest kapunk. Az iránytű szintén egyenes, annyi különbséggel, hogy 360-nál nullázódik.

Táblázatban összegyűjtöttem néhány mért forgási eredményt, amit különböző felületeken próbáltam ki különböző sebességek mellett, ugyanazon fok értékekkel és beállításokkal. Néhány értéktől eltekintve kimondható, hogy a sebesség növelésével egyre pontatlanabb az elfordulás mértéke. Valamint a felületek sem egyforma eredményt adnak. A kipróbáltak közül legjobbnak a szőnyeg ígérkezett, a többi mind fényes, csúszós felület. Azt is tapasztaltam, hogy ugyanolyan felületen, sebességgel, beállításokkal sosem kaptam két egyforma értéket egymás után, köszönhetően annak, hogy nem lehet kiszámolni előre, mennyivel fog megcsúszni a kerék. Azt is hozzá kell tennem, hogy az iránytű alapján az alábbi felsorolt értékektől különbözőbbeket kaptam. Ez azért lehetséges, mert a két szenzor közül az egyik pontatlanabb, de ami még valószínűbb a giroszkóp másodpercenkénti szögelfordulás idő szerinti integrálja több tizedesjegyből áll, és mindig az előző törtet számolja tovább, míg az iránytű csupán egész értékeket tud kezelni.

sebesség	felület				
	padló	fém	csempe	szőnyeg	fényes karton
100	366,8	365,6	366,4	352,5	369,2
200	372,9	370,5	383,2	371,4	374,4
300	378,2	377,7	380,6	372,2	375,6
400	375,9	377,6	375,3	363,2	375,6
500	382,4	368,8	383,1	370,5	378,1
600	379,3	377,9	383,2	369,8	375,8
700	381,4	377,0	385,2	369,8	379,4
800	384,7	384,0	384,8	370,5	382,2
900	382,8	381,0	387,7	371,2	382,4

6. Összefoglalás

Szakdolgozatom készítése közben megismerkedtem a giroszkóp, a gyorsulásmérő és a digitális iránytű működésével, azok használatával és ezekre támaszkodó programot alkottam. Véleményem szerint sikerült olyan programot írni, amely a belső érzékelőket felhasználva meg tudja mondani milyen irányban, milyen gyorsulással közlekedik a projektben résztvevő robot. Persze még lehetne rajta csiszolni és továbbfejleszteni is. Ezt kiegészítve egy GPS vevővel már komplett navigációs rendszert lehetne létrehozni, amely minden körülmény között megállná a helyét. Az eredmények pontatlansága miatt érdemes lenne kipróbálni egy stabilabb, precízebb robottal is.

Munkám során sok tapasztalatot gyűjtöttem. Remélem, a későbbiekben ezeket tudom majd kamatoztatni. Számos problémával találtam magam szemben, amik bebizonyították, hogy az elméletben tökéletesnek tűnő megoldások csak nagyon ritkán egyeznek meg a gyakorlattal. Ugyanakkor sokkal nagyszerűbb látni a végeredményt egy igazi, kézzel fogható eszközön, amely nem csupán egy szimulátorral szemlélteti az eredményt, hanem valós mozgása, reakciói vannak.

Valószínűleg a jövőben a navigáció elterjedése és fejlődése még inkább kihat majd a technológiákra, leginkább a robotokra. Véleményem szerint a robotok navigációja leginkább a mesterséges intelligenciával vegyített képfeldolgozásra fog alapulni.

Minden nap tudomást szerezhethetünk a legmeghökkenőbb hírekről. Talán az egyik ilyen, hogy elkészítettek egy robotot, mely legyekkel táplálkozik és ezeket alakítja át energiává. Azonban ezekből csak nagyon kevés erőforráshoz jut, így a mozgása is nagyon lassú.

Hová tart ez az iszonyú mértékű technikai fejlődés? Egyszer minden bizonnyal eljön az az idő, hogy a robotok navigációja tökéletesedik, meg tudják különböztetni a tárgyakat egymástól. A rohamos fejlődésnek köszönhetően néhány kérdés felmerült ezekkel a szerkezetekkel kapcsolatban. Ma már olyan dolgok tisztázásával is foglalkoznunk kell, mint a roboti jog. Mert mi történne akkor, ha egy robot bűncselekményt követne el? A robot a vétkes ilyenkor vagy az őt beprogramozó ember? A másik kérdés az ember-robot kapcsolat. Amerikában már felvetették, hogy miért nem lehetséges összeházasodni egy személynek és egy gépnek. Bármi is lesz a jövőben, vegyük figyelembe, hogy a robotok csak gépek, azért alkottuk őket, hogy biztonságosan megkönnyítsék és kiszolgálják az életünket, érzelmek

kifejezésére képtelenek, ha mégis rendelkeznek majd érzésvilággal, azt is csak előre betáplált módon tudják kifejezni.

7. Köszönetnyilvánítás

Ez úton szeretném megköszönni dr. Szabó Istvánnak, hogy rendelkezésemre bocsájtotta a felhasznált eszközöket, továbbá szakmai tanácsaival, ötleteivel és útmutatásával támogatta szakdolgozatom elkészülését.

8. Irodalomjegyzék

[1] Kucsera Péter: Zárt térben használható földi mobil robotok navigációs és akadályfelismerő szenzorrendszerei, a beltéri navigáció lehetőségei

members.chello.hu/kucsera.peter/gep.doc

Utolsó elérés ideje: 2010. május 4.

[2] Wikipédia: Érzékelő

<http://hu.wikipedia.org/wiki/Érzékelő>

Utolsó elérés ideje: 2010. május 3.

[3] LeJOS: Java for LEGO Mindstorms: LeJOS NXJ Tutorial

<http://lejos.sourceforge.net/nxt/nxj/tutorial/index.htm>

Utolsó elérés ideje: 2010. május 1.

[4] HiTechnic: Products

<http://www.hitechnic.com/products>

Utolsó elérés ideje: 2010. május 2.

[5] Bátfai Norbert: Operációs Rendszerek 2 – 6. és 7. előadás és labor

<http://www.inf.unideb.hu/~nbatfai/>

Utolsó elérés ideje: 2010. május 4.

[6] Fazekas Annamária: Robotok, fizikai ágensek

http://cs.ubbcluj.ro/~csatol/mestint/diak_pdf/2005/fazekasannamaria.pdf

Utolsó elérés ideje: 2010. május 4.

[7] Wikipédia: Mobile robot

http://en.wikipedia.org/wiki/Mobile_robot

Utolsó elérés ideje: 2010. május 3.

[8] Ipari robotok

<http://www.mestersegesintelligencia.hu/doc/ipari%20robotok.php>

Utolsó elérés ideje: 2010. május 2.

[9] Dr. Busics György: A műholdas helymeghatározás alapjai

http://www.geo.info.hu/portal2007/images/stories/bgy/nepszeru_gnss_1_resz_alapok.pdf

Utolsó elérés ideje: 2010. május 3.

[10] Műholdas helymeghatározás

http://rs1.sze.hu/~ricsi/telematika/muholdas_helymeghatarozas.doc

Utolsó elérés ideje: 2010. május 1.

További cikkek:

Az ipari robot

<http://www.villanyszaklap.hu/cikkek.php?id=397>

Utolsó elérés ideje: 2010. május 2.

Robot invázió

http://www.sztaki.hu/uploads/media/ITB_Kovacs_Gyorgy.pdf

Utolsó elérés ideje: 2010. május 4.

Mágneses iránytű

<http://xaknak.hrasko.com/20060408iranytu>

Utolsó elérés ideje: 2010. május 2.

Navigációról mindenkinek

http://www.pdacafe.hu/index.php?option=com_content&task=view&id=101&Itemid=42

Utolsó elérés ideje: 2010. május 1.

A költöző madarak látják a mágneses teret

<http://index.hu/tudomany/magnes0922/>

Utolsó elérés ideje: 2010. május 1.

Vass Gergő: Gépi látás

<http://www.vassg.hu/pdf/gepilatas1.pdf>

Utolsó elérés ideje: 2010. május 2.

Repülő autó kell az amerikai hadseregnek

http://index.hu/tech/2009/12/28/repulo_auto_kell_az_amerikai_hadseregnek/

Utolsó elérés ideje: 2010. május 2.

A sofőr nélküli Audi TT: robot vezeti a szupergépet

http://www.kisalfold.hu/auto/a_sofor_nelkuli_audi_tt_robot_vezeti_a_szupergepet_8211_vid_eoval/2134284/

Utolsó elérés ideje: 2010. május 2.

Robot a műtőben

http://egeszseg.origo.hu/cikk/0713/291112/20070328robot_a_muto_1.htm

Utolsó elérés ideje: 2010. május 1.

Nanorobotok

<http://www.kma-hu.com/vegsoidok/nanorobotok.htm>

Utolsó elérés ideje: 2010. május 1.

Felhasznált képek:

[I] Ipari robot az autógyártásban:

<http://www.mestersegesintelligencia.hu/doc/ipari%20robotok.php>

Utolsó elérés: 2010. április 28.

[II] Aibo robotkutya:

http://www.sony.net/SonyInfo/News/Press_Archive/199905/99-046

Utolsó elérés ideje: 2010. április 29.

[III] Robotfoci csapat 2009-ben:

<http://worldmustbecrazy.com/robo-soccer-world-cup-robocup-football-world-cup-2009>

Utolsó elérés ideje: 2010. május 5.

[IV] Gyorsulásmérő tengelyei:

<http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NAC1040>

Utolsó elérés ideje: 2010. május 3.

[V] GPS műholdak elhelyezkedése a Föld körül:

<http://astro.u-szeged.hu/szakdolg/vegiandras/felhasznalas/helymeghatározas.html>

Utolsó elérés ideje: 2010. április 20.