

<p>BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék</p>	<p>Robotirányítás rendszerteknikája előadás 6. fejezet</p>	<p>RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 1. oldal</p>
--	--	---

Robotirányítás

rendszerteknikája

Elektronikus jegyzet

6. fejezet

Készítette: Dr. Tevesz Gábor docens
BME Automatizálási és Alkalmazott Informatikai Tanszék
1117. Budapest, Magyar tudósok körùtja 2.
Q ép. B szárny II. em. B216.
Tel: 463-2881
Fax: 463-2871 (adm.)
Mail: tevesz@aut.bme.hu

Hallgatók: Villamosmérnöki és Informatikai Kar
Nappali tagozat
Villamosmérnöki mesterszak (MSc)
Számítógép alapú rendszerek szakirány hallgatói

<p>BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék</p>	<p>Robotirányítás rendszertechnikája előadás 6. fejezet</p>	<p>RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 2. oldal</p>
--	---	---

COPYRIGHT

Jelen dokumentum a BME Villamosmérnöki és Informatikai Kar hallgatói számára készített elektronikus jegyzet. A dokumentumot a Robotirányítás rendszertechnikája c. tantárgyat (BMEVIAUM255 ill. BMEVIAUMA10) felvevő hallgatók jogosultak használni, és saját céljukra 1 példányban kinyomtatni. A dokumentum módosítása, bármilyen eljárással részben vagy egészben történő másolása tilos, illetve csak a szerző előzetes engedélyével történhet.

Copyright © 2009-2017 / Dr. Tevesz Gábor

Köszönetemet fejezem ki Kiss Domokos tanársegédnek a 6.3 fejezet anyagának és a Függelék példáinak kidolgozásáért.

Copyright © 2009-2017 / Dr. Tevesz Gábor

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 3. oldal
--	---	---

6. MOBIL ROBOTOK	4
6.1 Mobil robotok osztályozása	4
6.1.1 Kerekeken mozgó robotok mobilitása	6
6.1.2 Járó robotok	16
6.1.3 Moduláris robotok	18
6.2 Mobil robotok tájékozódása	22
6.2.1 Odometria.....	24
6.2.2 Odometria optikai szenzorokkal.....	28
6.2.3 Inerciális helymeghatározás	29
6.2.4 Az ultrahangos távolságmérés elve	36
6.2.5 GPS és DGPS	44
6.3 Mobil robotok navigációja akadályok között	65
6.3.1 A konfigurációs tér	67
6.3.2 Mozgástervezési módszerek osztályozása	72
6.3.3 Pályatervezési módszerek	74
6.3.2.1 Láthatósági gráf.....	74
6.3.2.2 Cella-dekompozíció	77
6.3.2.2 Rapidly Exploring Dense Trees	79
6.3.4 Akadályelkerülési módszerek	82
6.3.4.1 Virtuális erőtér	82
6.3.4.2 Vektormező hisztogram	84
6.3.4.3 Dinamikus ablak módszer	90
FÜGGELÉK	95
F.1 feladat – Odometria hibaszámítás és UMBMark kalibráció.....	95
F.2 feladat – Ultrahangos szonár rendszer	100
F.3 feladat – Omnidirekcionális hajtás	106

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 4. oldal
--	---	---

6. MOBIL ROBOTOK

Ebben a fejezetben a mobil robotika világát tekintjük át röviden. A mobil robotika napjaink egyik legdinamikusabban fejlődő, legtöbb lehetőséget magában hordozó mérnöki területe. Ez a terület sok tekintetben megváltoztatja a robotikai fejlesztések irányát. A korábbi fejezetekben már önmagában is komplex irányítási feladatként bemutatott, adott esetben még különösebb „intelligenciával” sem rendelkező hagyományos ipari robotok mellett a hétköznapi használat szintjén (háztartásokban, egészségügyben, stb.) is egyre gyakrabban alkalmazzák az autonóm, döntési feladatokat is ellátni képes mobil robotokat.

Mivel a stabil (fixen telepített) robotok feladataikat korlátozott téren végzik, az ipari alkalmazásokban (hegesztés, összeszerelés, festékszórás, stb.) legjobban elterjedt robotkarokat viszonylag könnyű létrehozni, és megfelelően programozni. Ám, ha egy robotot „magára hagyunk” az őt körülvevő világban, a tervezőknek lényegesen több problémát kell megoldaniuk, melyek közül az adott terepre optimalizált (kereken, síneken vagy lábakon mozgó) helyváltoztató rendszer kidolgozása jelenti az egyik legfőbb kihívást. Ahhoz, hogy a gépek minél mozgékonyabbak legyenek, jól alkalmazkodjanak a folyamatosan változó környezethez, a környezet változásai ne zavarják meg a viselkedésüket (zaj- és hibatűrők legyenek), és a mindenkor környezeti helyzet függvényében, időben hozzák meg döntéseiket, egyre nagyobb önállósággal és intelligenciával kell bírniuk.

A mai mobil robotok általában nem eléggé megbízhatóak. Mint minden számítástechnikai rendszernél, így a robotikai fejlesztésekben is nagyobb szerephez kell jutniuk a speciális megbízhatóságot mérő módszereknek, melyekkel előre jelezhető a robot meghibásodása, az elvégzendő feladat sikeres végrehajtásának valószínűsége. A mobil robotok akkor váltják majd be a hozzájuk fűzött, napjainkban még néha túlzottnak tűnő reményeket és elvárásokat, ha feladataikat nemcsak laboratóriumi közegben, hanem a dinamikus, percről percre módosuló „külvilágban” is teljes biztonsággal képesek lesznek valós időben elvégezni. Beépített, önállóan kialakított modellel kell rendelkezniük környezetükön (legalábbis annak a rendelkezésre álló szenzorokkal folyamatosan meg nem figyelhető vonatkozásairól), érzékelniük és elemezniük kell ezeket a változásokat, meg kell találniuk benne a saját pontos pozíciójukat, majd meg kell tervezniük, végre kell hajtaniuk a mozgást. Összegezve: egyfajta intelligens cselekvést, cselekvéssort kell kivitelezniük. Csak akkor könnyítik meg az ember életét és állnak helyt a minden nap különböző területein – ilyenek pl. a takarítás, épületmonitorozás, szórakoztatás, bevásárlóközpontok, múzeumok, kórházak (belő, strukturált környezetek), erdők, bányák, tengerfenék, katasztrófa sújtotta és katonai terepek, világűr (külső, strukturálatlan környezetek) –, ha eleget tesznek ezeknek a feltételeknek.

6.1 Mobil robotok osztályozása

A robotok mozgásának módját alapvetően két szempont határozza meg: az elvégzendő munka és a munkavégzés környezete. Hárrom környezettípuszt különböztetünk meg: levegő/világűr, víz és szárazföld.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 5. oldal
--	---	---

- **Levegő/világűr**

Légi robotokat elsősorban az űrhajózásban (Cassini, stb.), valamint a hadászatban használnak. A legismertebb katonai alkalmazások az ember nélküli járművek (Unmanned Aerial Vehicle - UAV). Egyes rendszerek, például a drónok (pilóta nélküli célrepülőgépek, melyekkel felderítenek, amelyekre célzási gyakorlatokon lőnek) gyorsabban fejlődnek: könnyebb megépíteni és programozni őket, nem kell akadályok között navigálniuk.

- **Víz**

Az óceánokban, tengerekben tudományos és ipari célokkal egyaránt használnak alámerülő robotokat, automatizált víz alatti járműveket (Automated Underwater Vehicle, AUV). Biológia és számítástudomány integrációját jól szemlélteti, hogy egyes vízi robotokat élőlényekről mintáznak. Érdekes kísérlet a vízen és szárazföldön egyaránt funkcionáló, kételtű robotok fejlesztése.

- **Szárazföld**

Mivel az ember is a szárazföldön él, az ebben a környezetben működő mobil robotok a leggyakoribbak. A továbbiakban főként a tipikusan szárazföldi feladatokat végző mobil robotokat vizsgáljuk.

A helyhez nem kötött szárazföldi robotok helyváltoztatása két alapvető koncepció szerint valósul meg: az egyik csoportba a síneken/kerekeken mozgó, a másikba a járó robotok tartoznak. Léteznek más csoportosítások is, például a helyüket sínnel, illetve kerékkel változtatók külön kategóriába sorolása. A tudósok fejlesztései sem mindig igazodnak a fenti osztályozáshoz, valószínűleg hosszasan vitatkozhatnánk arról, hogy a Carnegie Mellon Egyetem (CMU – Pittsburgh, USA) egyetlen gömbszerű labdán járó Ballbotja egy újabb osztály előfutára, vagy a már létező kettő valamelyikébe tartozik.

Az első csoportot főként két-, három-, négy-, vagy hatkerekű robotok képviselik. A kereken mozgó robotok kötött pályás és kötetlenül mozgó alkategóriákba sorolhatók. Közös tulajdonságuk a lépegető társaikhoz képest jelentősen kisebb az energiaigényük. Legfőbb problémájuk azonban, hogy gyakran a legkisebb mértékben egyenetlen terepen is elakadhatnak (mint azt a Sojourner Mars-járó tette 1997-ben), és általában a legkisebb akadályokat is képtelenek átlépni – inkább kikerülök azokat (arról már nem is beszélve, hogy otthoni környezetünkben gyakoriak a lépcsők, ajtóküszöbök, stb.). Viszont – jelenleg még – gyorsabbak, mozgékonyabbak, mint a helyváltoztatást lábakkal megvalósító társaiik.



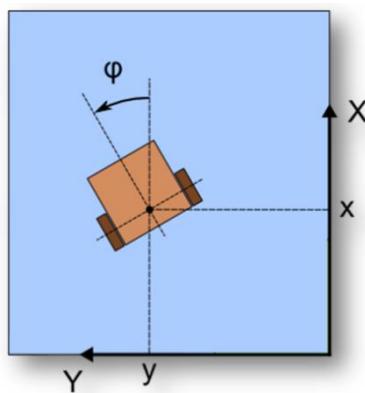
6.1.1 ábra Ballbot –
az „egykerekű” robot

6.1.1 Kerekeken mozgó robotok mobilitása

Kerekeken mozgó mobil robotok esetén a hajtott kerekek elhelyezkedése, kialakítása és szabadsági foka határozza meg a robot helyzetváltoztatási lehetőségeit. Mindegyik kerék konfigurációnak meg vannak az előnyei és hátrányai, ezért a robot tervezésénél nem árt alaposan meggondolnunk, melyiket választjuk. A robot elrendezéseket két nagy csoportba soroljuk: *holonom*, és *anholonom (nem holonom)*. Egy mechanikai rendszert akkor nevezünk anholonomnak, ha egy adott állapotában (konfigurációjában) létezik olyan elmozdulás/elfordulás irány, amely nem adható meg a lehetséges beavatkozások hatásainak algebrai kifejezéseként. Anholonom rendszerek legtipikusabb példája a személyautó, amely az aktuális orientációjára merőleges irányba nem tud elmozdulni.

Amennyiben sík terepen közlekedő robotról van szó, akkor a kívánt szabadsági fokok száma három:

- 1) a robot x irányú pozíciója
- 2) a robot y irányú pozíciója
- 3) a robot φ orientációja



6.1.1.1 ábra – Sík terepen közlekedő robot pozíciója és orientációja

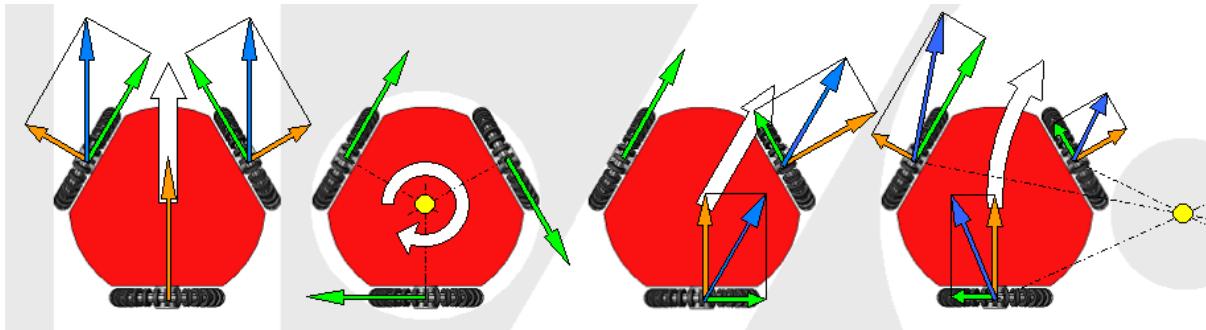
(A levegőben navigáló robotok fentiekkel szemben 6 szabadsági fokkal rendelkeznek: ezek a pozíciót meghatározó 3 Descartes-koordináta, valamint a három koordinátatengely mentén értelmezett szögfordulások).

a) holonomikus robotok

A gyakorlatban alkalmazott robothajtás-konfigurációk között viszonylag ritka a holonomikus rendszer. Ennek oka, hogy az ilyen meghajtású robotok általában vagy bonyolult és drága mechanikát követelnek meg, vagy a helymeghatározás során alkalmazott odometria (kerekek elfordulásából származtatott pozíciókövetési eljárás, ld. később) igen pontatlan, ezért egyéb helymeghatározási módszert igényelnek.

A holonomikus robotok közül talán az omnidirekcionális hajtású robot nevezhető a legegyeserűbbnek mechanikailag. Neve is utal rá (*omni* – minden, *direction* – irány), hogy képes bármilyen irányú sebesség elérésére, és orientációját is képes ettől függetlenül

változtatni. Háromkerekű változata már holonomikus, hiszen három függetlenül hajtott motorral rendelkezik.



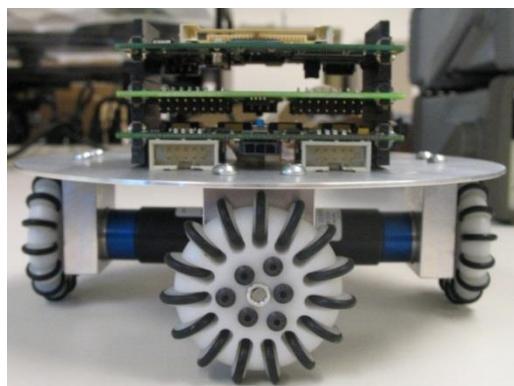
6.1.1.2 ábra – Háromkerekű omnidirekcionális robot sebességvektorai

Az ábrán az eredő sebességet a fehér nyíl jelöli. Ez a fajta meghajtás speciális kerekekkel igényel, hiszen az ábrán látható eredő (kék) sebességevektorok csak úgy állhatnak elő, ha a kerekek képesek a tengelyükkel párhuzamosan is elmozdulni. Az ehhez szükséges speciális kerekekre láthatunk két példát a következő ábrán.



6.1.1.3 ábra – Kerekek az omnidirekcionális hajtáshoz

Természetesen elhelyezhető egy rendszerben négy, illetve több kerék is, de ebben az esetben elképzelhető olyan eset is, hogy a kerekek szembe hajtják egymást, valamint figyelembe kell venni, hogy sík terepen a négy pontos alátámasztás nem garantálja az összes kerék megfelelő érintkezését.

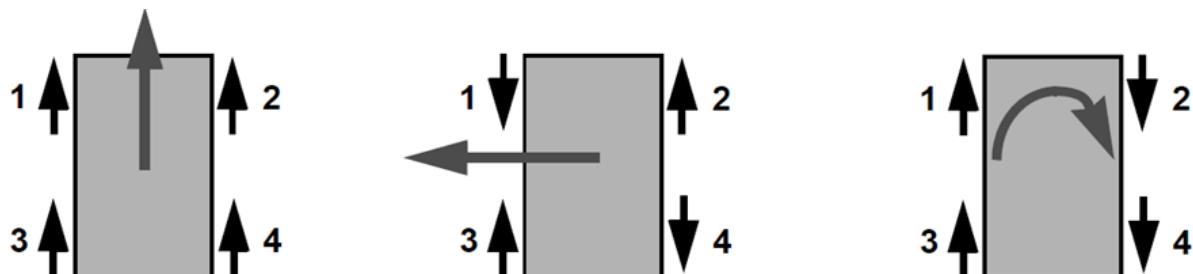


6.1.1.4 ábra – Négykerekű omnidirekcionális hajtás

Érdekes holonomikus irányítási lehetőséget kínálnak az ún. mecanum (svéd) kerekekkel felszerelt mobil robotok, melyeket a svédországi Mecanum AB nevű cég fejlesztett ki.



6.1.1.5 ábra – Mecanum kerekess holonom robot



6.1.1.6 ábra – Mecanum kerekkekkel minden három szabadsági fok irányába vezérelhető a robot

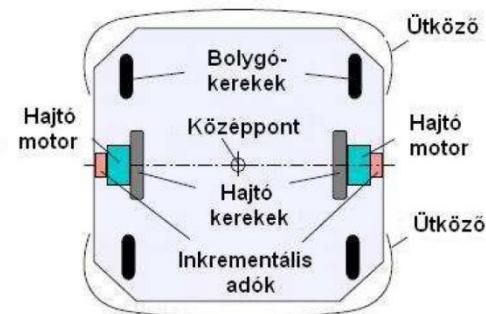
A megoldás hátránya, hogy a svédkerekkekkel fölszerelt robot oldalirányú stabilitása kisebb, mint a hagyományos társáé, ezért a négy motornak minél inkább egyező erővel és szinkronban kell dolgozniuk már az egyenes irány tartásához is.

Az omnidirekcionális megoldások egy további fontos hátránya, hogy a kerekkek kerületi kialakítása csak közelíti a kört, ezért az elfordulásukból leszármaztatott helymeghatározás (odometria, ld. később) számítása nagyon hamar pontatlannak lesz.

b) nem holonomikus robotok

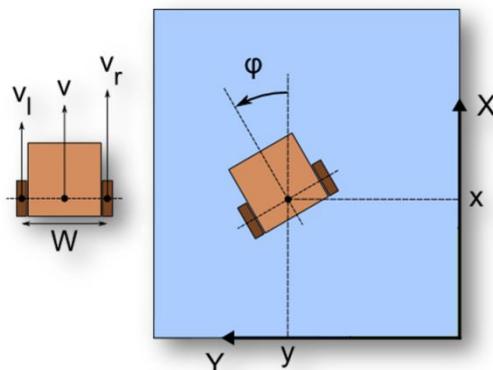
A gyakorlatban sokkal elterjedtebbek a nem holonomikus robotok, gondolunk csak egy automata kerekess székes alkalmazásra, robotautóra vagy lánctalpas eszközökre.

A beltéri mobil robotikában elterjedten alkalmazzák a differenciális hajtást, mivel egyszerűsége mellett manőverező képessége is megfelelő.



6.1.1.7 ábra – A differenciális hajtás elve

A differenciális hajtású robot kinematikáját az alábbi egyenletek írják le:



6.1.1.8 ábra – Differenciális hajtású robot jellemzői

$$v = \frac{v_l + v_r}{2} \quad (6.1.1)$$

$$\omega = \frac{v_r - v_l}{W} \quad (6.1.2)$$

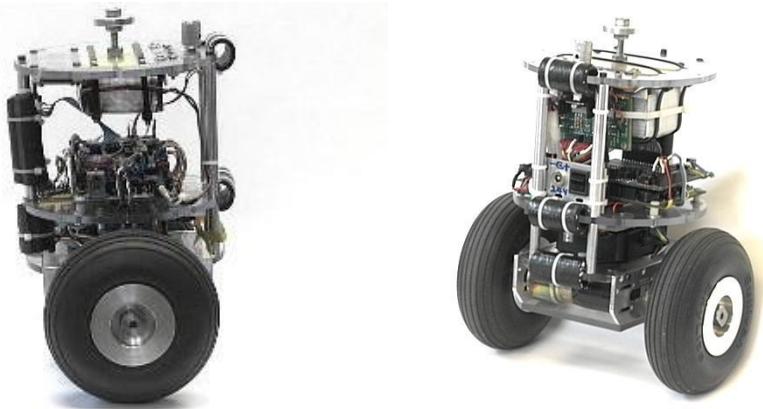
$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} v \cdot \cos \phi \\ v \cdot \sin \phi \\ \omega \end{pmatrix} \quad (6.1.3)$$

ahol v a robot pálya menti sebessége, ω a robot szögsebessége, v_r és v_l a jobb, illetve bal kerék kerületi sebessége, W a két kerék távolsága, x és y a robot referencia pontjának koordinátái, ϕ pedig a robot orientációja.

Fontos, hogy a két hajtott kerék tengelye egybeessen, hiszen csak így kerülhető el a kerekek csúszása. Az ilyen hajtással rendelkező robot 2 szabadsági fokkal rendelkezik, így csak bizonyos irányokban képes mozogni. A robot két, függetlenül hajtott kerekén kívül általában (minimum) egy bolygókerék – ún. Castor kerék - elhelyezése szükséges ahhoz, hogy a robot

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 10. oldal
--	---	--

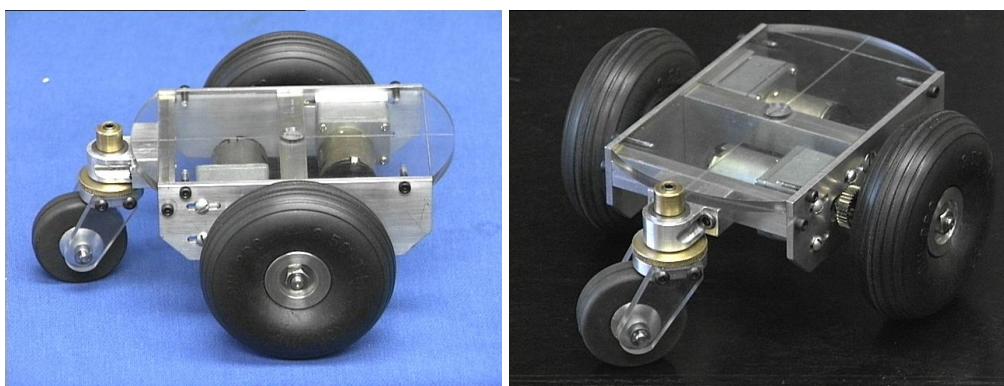
legalább három pontos alátámasztása garantált legyen, bár mint látni fogjuk, ez sem törvényszerű előírás.



6.1.1.9 ábra - A két keréken egyensúlyozó nBot

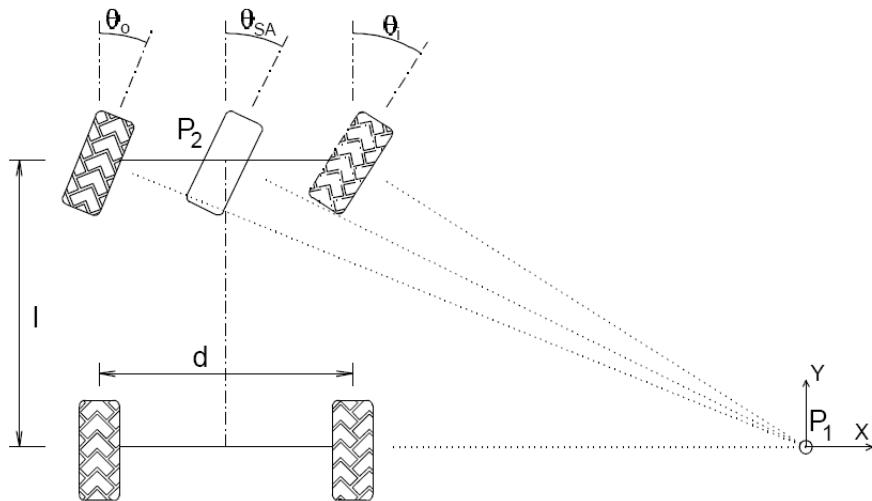
Mind a korábban már bemutatott Ballbot, mind az nBot és a további kétkerekű robotok esetében számolunk kell még egy az eddigiekben felüli szabályozási kényszerrel: az egyensúly megtartásával.

A három- és a négykerekű robotok esetében is kormánymű alkalmazása helyett előszeretettel alkalmazzák a differenciális hajtást, két kerék független motorokkal történő meghajtását. A harmadik és a negyedik kerék – általában bolygókerék formájában – csak támasztó szerepet lát el. A két hajtott kerék eltérő fordulatszámmal történő mozgatása automatikusan a robot kanyarodását váltja ki, ellentétes irányba történő forgatásuk a robot helyben történő megfordulását is lehetővé teszi.



6.1.1.10 ábra - 3 kerekű mobil robot mechanikai felépítése és a kerekek hajtása

Háromkerekű hajtás esetén elől található a hajtott kerék, amelynek kormányszögét (θ_{SA}) változtathatjuk, hátul pedig két egytengelyű passzív kerék adja az alátámasztást. Látható, hogy ebben az esetben csak két szabadsági fokunk van (a hajtott kerék szögsebessége és kormányszöge), tehát hasonlóan a differenciális hajtáshoz, itt sincs lehetőségünk bármely sebességvektor mentén mozogni.



6.1.1.11 ábra - Hárromkerekű hajtásra visszavezetett Ackerman hajtás

A négykerekű robotok egy jelenős része a járműveinkhez hasonló, Ackerman rendszerű kormányművel rendelkező ún. „autószerű” (vehicle-like) robot. Fontos, hogy a θ_0 és θ_1 kormányszögek nem egyezhetnek meg, mivel akkor elkerülhetetlen a kerekek folyamatos csúszása. A 6.1.1.11 ábra szemlélteti az Ackerman hajtás visszavezethetőségét háromkerekű hajtásra, amennyiben a két első kerék tengelyei a körülfordulási (P_1) pontban metszik egymást.

A járművekbe épített intelligencia mértékét jellemzi, hogy 2005-ben már történelmet írt egy sofőr nélküli, tisztán számítógépek által irányított Volkswagen Touareg terepjáró: a világban elsőként teljesítette a DARPA Grand Challenge-et, egy több mint 200 kilométeres versenyt a Mojave sivatagban. A Stanley névre keresztelt autót a Stanford egyetem mesterséges intelligenciával foglalkozó kutatólaboratóriuma készítette fel.



6.1.1.12 ábra - Stanley, a VW vezető nélküli terepjárója (DARPA Grand Challenge)

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 12. oldal
--	---	--

A Google vezető nélküli autója (*Google driverless car*) projekt célja a sofőr nélküli autók technológiájának kifejlesztése. Vezetője a Google mérnöke, Sebastian Thrun, a Stanford Artificial Intelligence Laboratory igazgatója, az ő csapata fejlesztette ki Stanley-t.

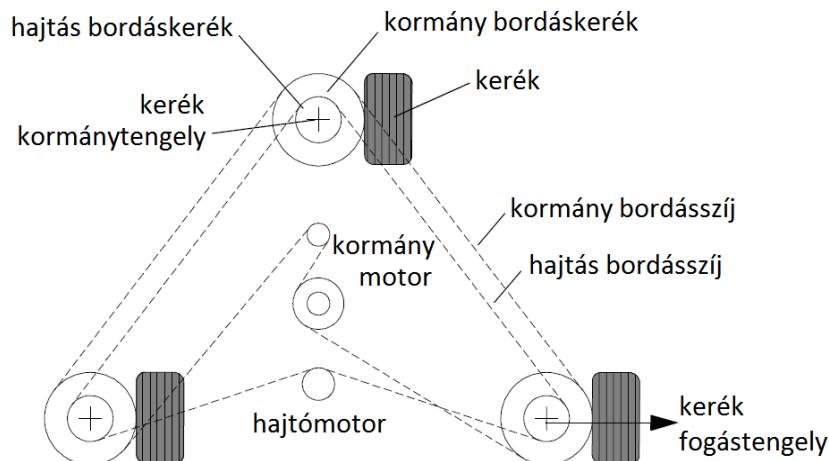


6.1.1.13 ábra – Egy Lexus RX450 vezető nélküli autó (*Google Car*)

Az Amerikai Egyesült Államokban Nevada állam 2011. június 29-én engedélyezte, hogy útjain vezető nélküli autók is részt vehessenek a közúti forgalomban. (Ez nem egy egyszerű döntés, a Google nagyon sok lobbija fekszik abban, hogy ez az engedély megszületett.) 2012 augusztusában Thrun fejlesztőcsapata bejelentette, hogy vezető nélküli autóik segítségével több mint 500 000 km-t tettek meg az utakon balesetmentesen, a mintegy egy tucatnyi autóból álló flottájuk segítségével

2014. december 22-én a Google bejelentette, hogy elkészült az önvezető autó prototípusával és 2015-től közúton teszteli azt San José-ban. A cég az autót a prototípus elkészültétől számított öt éven belül tervez kereskedelmi forgalomba hozni, sorozatgyártásához pedig autóipari partnereket keres. A vállalatnál bíznak abban, hogy a vezető nélküli autók iránti érdeklődésre reagálni fognak az autógyártók és akkor elkezdődhet a sorozatgyártás. A California Department of Motor Vehicles egyik ügyvédje szerint aggodalomra adhat okot, hogy a technológia számos esetben a törvényi szabályozás előtt jár. Az ügyvéd idézte az állam erre vonatkozó törvénycikkelyét, amely feltételezi, hogy az ilyen technológiával is működtethető járművekben minden lesz ember, aki szükség esetén beavatkozik és működteti a járművet. Megválaszolatlan kérdések sorát veti fel az önvezető autó megjelenére, melyekre egyenlőre közlekedési szakemberek, mérnökök és jogászok együtt keresik a válaszokat.

A szinkron hajtás egy olyan érdekes megoldás, amely mechanikai úton biztosítja a kerekek csúszásmentességét. Az ábrán látható megoldásban a hajtás fogaskerekek (áttételen keresztül) merev kapcsolatban vannak az egyes kerekekkel, a kormány bordáskerekek egyszerre fordítják azokat.

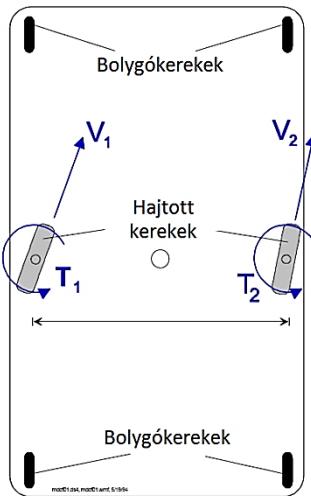


6.1.1.14 ábra – A szinkron hajtás elve

Hasonlóan az omnidirekcionális hajtáshoz, itt is három hajtott kerék van, viszont ezek nem speciális kerekek, csak a tengelyükre merőleges elmozdulásra képesek. A kerekek szögsebessége megegyezik, erről egy közös hajtás bordásszíj gondoskodik, valamint a kerekek folyamatos párhuzamos síkja is garantált, mivel a kormányzás is egy közös kormány bordásszíj segítségével történik. Ezen két tulajdonság biztosítja a szinkronizációt (egyenlő nagyságú és párhuzamos erővektorok a kerekek érintési pontjaiban), így a kerekek csúszása megfelelően pontos mechanika esetén nem fordulhat elő. A szinkronizált vezérlés nem csak bordásszíjjal oldható meg: bizonyos alkalmazásokban a kerekek kormányzásáról és hajtásáról független szervomotorok gondoskodnak, a szinkronizáció az elektronikus irányítórendszer feladata - de az ilyen alkalmazásoknál a szabályzási tranziensek eltérése csúszást okozhat.

A szinkron hajtás előnye a jó manőverezési képesség (a robot orientációjának változtatása nélkül álló helyzetből akármilyen irányú sebességvektor mentén indulhat – jóllehet ehhez a kerekeket álló helyzetben forgatni kell), illetve a pontos odometria számításának lehetősége, viszont látható, hogy ez a konstrukció is csak 2 szabadsági fokkal rendelkezik, így ellentétben pl. az omnidirekcionális hajtással, a robot sebessége és szögsebessége nem változhat függetlenül.

Léteznek olyan kerekess mobil robotok is, amelyek a szükségesnél több szabadsági fokkal rendelkeznek. A sok szabadságfokú (*MDOF* – *Multi-degree-of-freedom*) robotok manőverezési képessége jobb, mint egyszerűbb társaiké. Túlhatározott mivoltukból következően viszont bonyolult feladat a kerekek olyan irányítása, hogy azok csúszását elkerüljük (akár szembe is hajthatjuk őket). Az ilyen tulajdonságú robotoknál szükség van külső érzékelőkre ahhoz, hogy a robot pozíciója és orientációja a megfelelő pontossággal ismert legyen.



6.1.1.15 ábra – 4 szabadságfokú robot platform

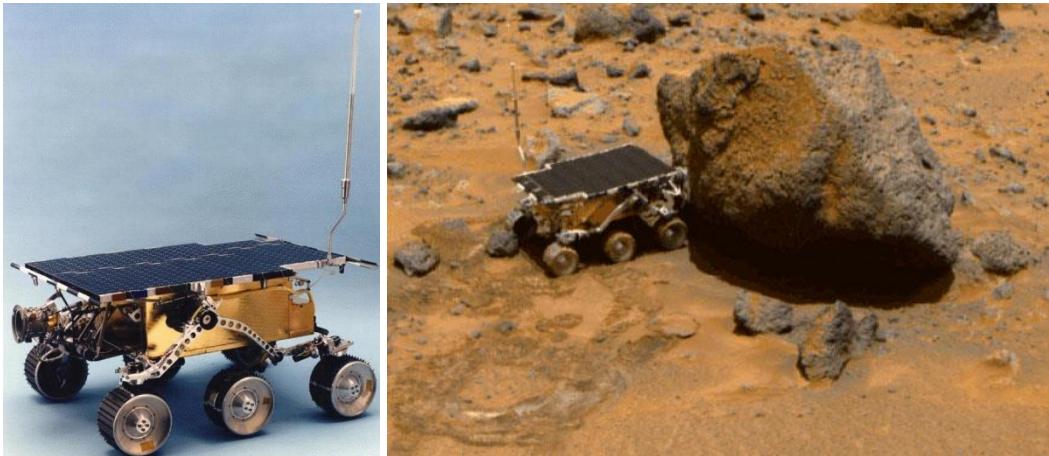
A sok szabadságfokú robotplatformok tipikus felhasználási területe a durva felszínen történő navigáció (katonai alkalmazások, holdjáró robotok). Ilyen alkalmazásoknál fel sem merül a kerekek elfordulásából származtatott pozíció meghatározás kizártlagos alkalmazása.

A 4 kerekű mobil robotok sem feltétlenül követik a megszokott járműkonstrukciót – a gyors, kis helyen történő manőverezési képesség egészen újszerű megoldásokban, akár minden a 4 kerék közvetlen meghajtásában és kormányozhatóságában is megnyilvánulhat, mint a Unique Mobility Inc. által katonai célokra létrehozott UNIQ robot.



6.1.1.16 ábra - UNIQ: Kerekenként kormányzott és hajtott (8 DOF) mobil robot

Egyenetlen terepen közlekedő járművek esetében a legjobban bevált megoldás a több, nagyon lágy felfüggesztésű, általában egyenként hajtott kerekekre épülő hajtás és kormányzás. (Sokszor ügyelni kell a sziklás egyenetlen terep, az átmenetileg levegőben maradó hajtókerék és a felborulás veszélyére is.) Az előre nem látható extrém környezeti akadályok azonban eredményezhetnek olyan helyzeteket, melyeket a robot fizikailag nem, vagy csak nehezen képes leküzdeni (Mars expedíció, Sojourner űrautó).



6.1.1.17 ábra - Sojourner (Mars Pathfinder expedíció, 1997 – NASA)

Katonai alkalmazásokra általában a katonai járműveknél már jól bevált többkerekű, vagy lánc- vagy hernyótalpas megoldásokat alkalmazzák (szeme előtt tartva a kerekeknek a harci környezetben való kisebb sebezhetőséget is)

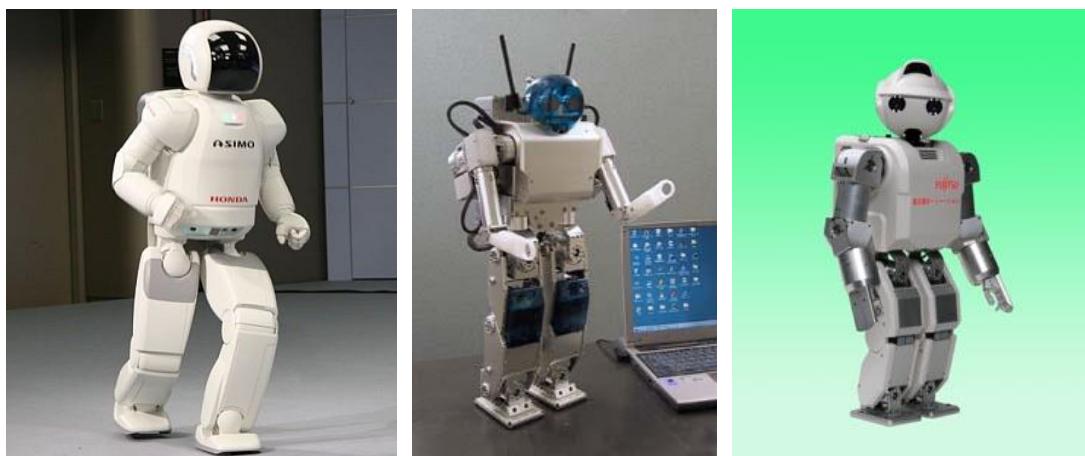


6.1.1.18 ábra - Katonai célú mobil robotok: az amerikai SWORDS és az orosz MRK-27BT

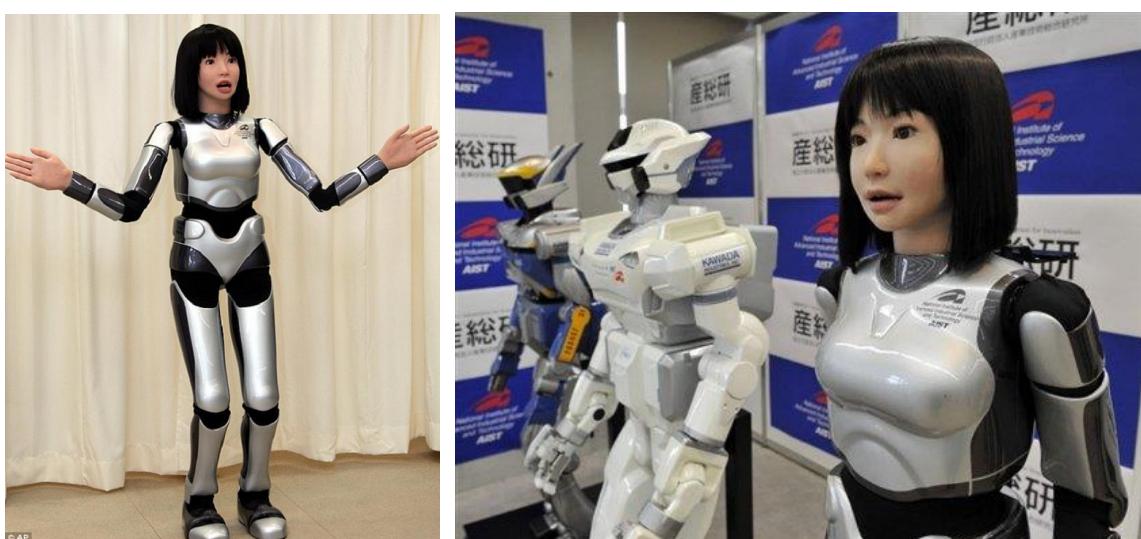
BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 16. oldal
--	---	--

6.1.2 Járó robotok

A járó robotok további két alcsoporra, a majdnem vagy teljesen vízszintes terepen működő „lépegetőkre” és a meredek falakon tevékenykedő „mászókra” oszthatók. Lábaikat elsősorban villanymotorok, illetve hidraulikus vagy pneumatikus dugattyúk mozgatják előre és hátra, de komoly kísérleteket folytatnak az ún. izomhuzal (muscle wire) megoldásokkal, melyekkel nagyon kis tömegű, nagy megbízhatóságú mozgató mechanizmusok készíthetők el. A dugattyúk ugyanúgy kapcsolódnak a láb különböző részeihez, mint az izmok a csontokhoz. A különböző dugattyúk összehangolása, összehangolt működésük megtervezése igen nehéz feladat. A tervezőnek ki kell találnia, és be kell programoznia a számítógépbe a helyváltoztatásban érintett dugattyúmozgások helyes kombinációját, máskülönben a robot nagyon hamar elesik. Sok mobil robot rendelkezik a számítógépnek mozgáskorrekcióra vonatkozó utasításokat adó beépített egyensúlyozó rendszerrel (például giroszkóppal).



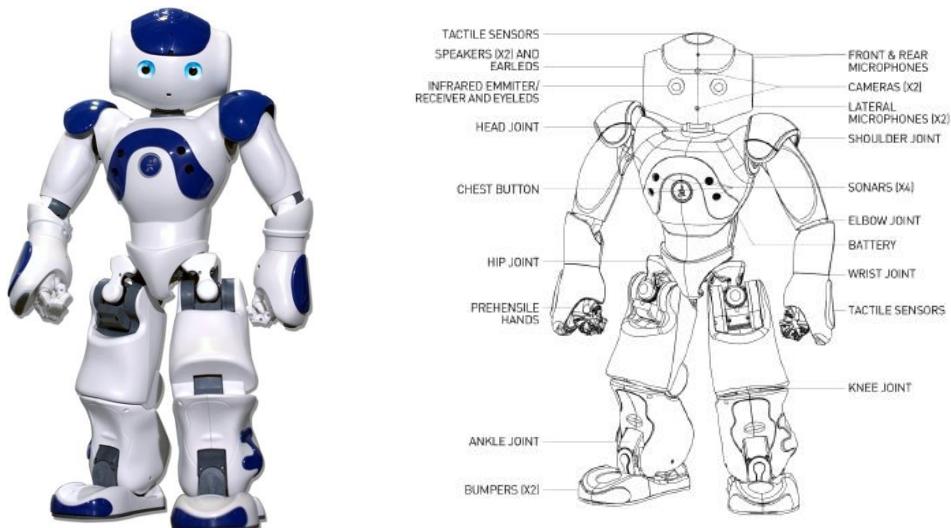
6.1.2.1 ábra - Humanoid járó robotok: Asimo (Honda), a HOAP-1 és a HOAP-3 (Fujitsu)



6.1.2.2 ábra - Egy „csinosabb” humanoid robot megoldás (HRP-4C)

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 17. oldal
--	---	--

Az Aldebaran Robotics nevű francia vállalat fejlesztette ki a Nao névre keresztelt humanoid, önállóan cselekvő, de programozható, Linux operációs rendszerrel működő robotokat. A robot 58 centiméter magas, 5 kg tömegű, és 25 szabadságfokkal rendelkezik, ami sokkal finomabb mozgást tesz számára lehetővé, mint amire a legtöbb jelenlegi kísérleti modell képes. Készítői elsősorban a robot kezét tökéletesítették, hogy az a lehető legjobban hasonlítsan az emberi kézre, így a Nao markolni is képes. A robot kamera segítségével lát, szintetizátor segítségével beszél. A modell programozásához az Aldebaran az URBI (Universal Real-time Behaviour Interface) nevű, nyílt forráskódú parancsnyelvet használja, amelyet mozgó alkatrészek és szenzorok vezérlésére fejlesztettek ki. A felhasználó ennek segítségével személyre szabhatja majd annak viselkedését, érzelmi reakcióit és képességeit. A robot fejében egy 1.6 GHz-es intel Atom processzor található, amelyen a Linux kernel fut az összetett feladatrendszer ellátására (további slave CPU is található még a testében).



6.1.2.3 ábra – Nao, az 58 cm magas 25 szabadságfokú humanoid robot

Az Aldebaran Robotics cég a robotokat négyfélé, 25, 21, 14 és 2 szabadságfokú kivitelben hozza forgalomba.



6.1.2.4 ábra – A 14 szabadságfokú Nao

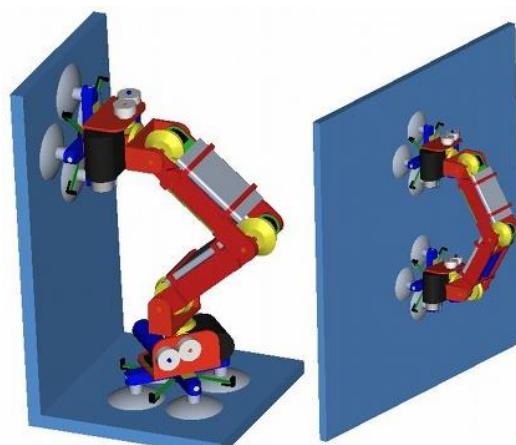
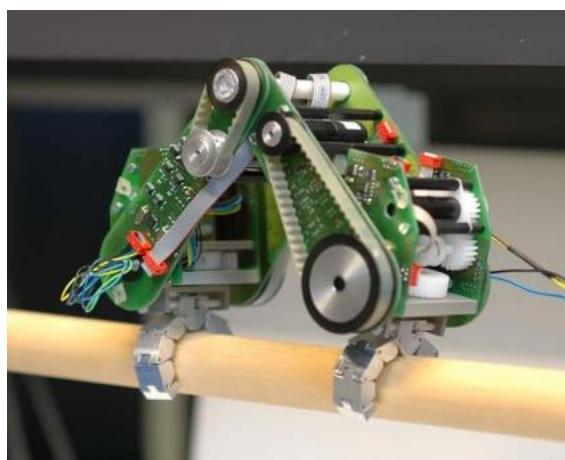
BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 18. oldal
--	---	--

Legtöbb gondot a két lábon történő (nem optimalizált, alapvetően bizonytalan) helyváltoztatás kivitelezése jelenti. A nagyobb stabilitás érdekében, a tervezők gyakran az állatvilágból lesnek el minták. A hatlábú rovarok például speciálisan jól egyensúlyoznak, és tökéletesen alkalmazzkodnak a legkülönbözőbb terepekhez.



6.1.2.5 ábra - *Genghis és a magyar Figaro, a rovarszerű mobil robotok*

Az egyedi igények egészen speciális robotkonstrukciókat is eredményezhetnek – mint a csövön, falon mászó robotok esetében.



6.1.2.6 ábra - *Speciális cső- és falmászó robotok*

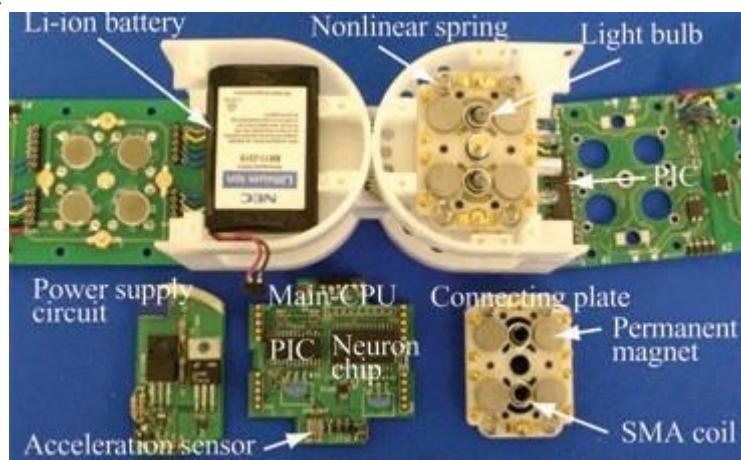
6.1.3 Moduláris robotok

Teljesen újszerű megoldásokat és ennek megfelelően a fenti osztályokba való besorolhatatlanságot képviselnek az önmagukat átépíteni (SR – Self Reconfigurable), alakjukat változtatni képes, helyváltoztatásukat akár saját elemeik szétválasztásával, majd újraegyesítésével megvalósító moduláris robotok. Egyik képviselőjük a japán AIST (National Institute of Advanced Industrial Science and Technology) mérnökei által kifejlesztett MTRAN I. – III. moduláris robotcsalád (MTRAN = Modular TRANsformer).



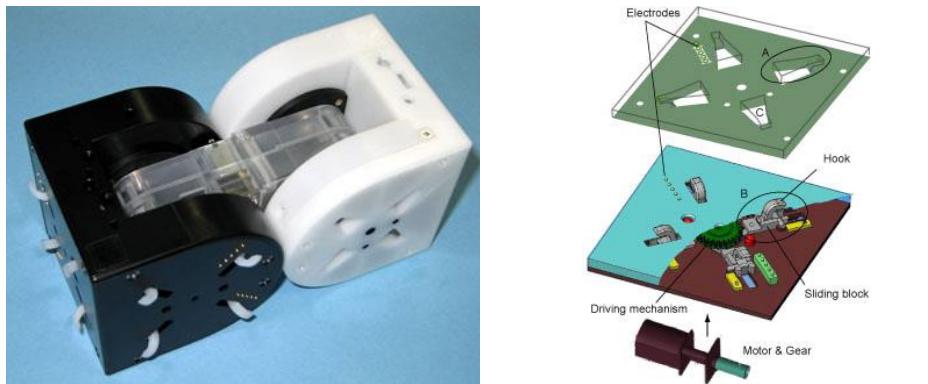
*6.1.3.1 ábra - MTRAN moduláris robot generációk
(MTRAN-I: 1998, MTRAN-II: 2002, MTRAN-III: 2005)*

Az MTRAN egymáshoz kapcsolt moduljai hasonlóan működnek, mint az élőlények sejtjei. Mechanikus funkciókkal bírnak, információt dolgoznak fel. Hárrom részből állnak: aktív és passzív tömbből, valamint a kettő közötti kapocsból. Mind az aktív, mind a passzív "kockák" az összekötő elem körüli száznyolcan fokos forgásra képesek. A modulok a rendszer többi moduljával kapcsolatot létesítő, ki- és bekapcsolt (on-off) állapotot váltogató mágneset is tartalmaznak. A kapcsolatteremtést, annak megszakítását nem külső erők, hanem maguk a mágnesek végzik.



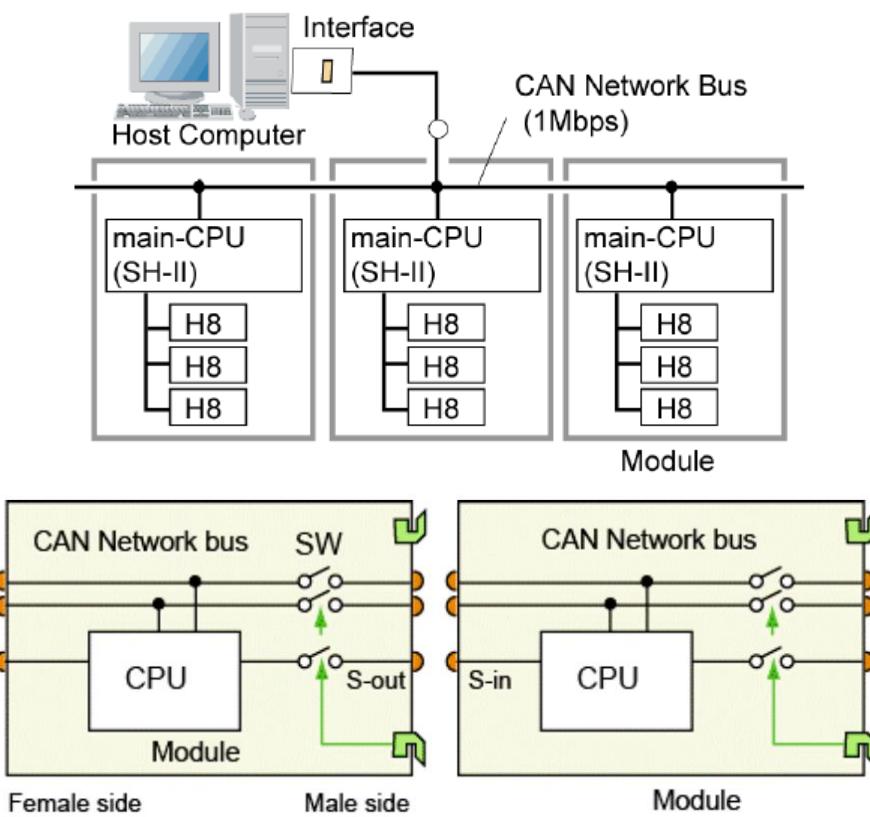
6.1.3.2 ábra - Az MTRAN-II belső felépítése

Az MTRAN III esetében energiatakarékosági okokból a kapcsolódás nagy sebességgel mozgatott mechanikai elemek segítségével történik.



6.1.3.3 ábra - MTRAN-III – a kapcsolódó építőelemek

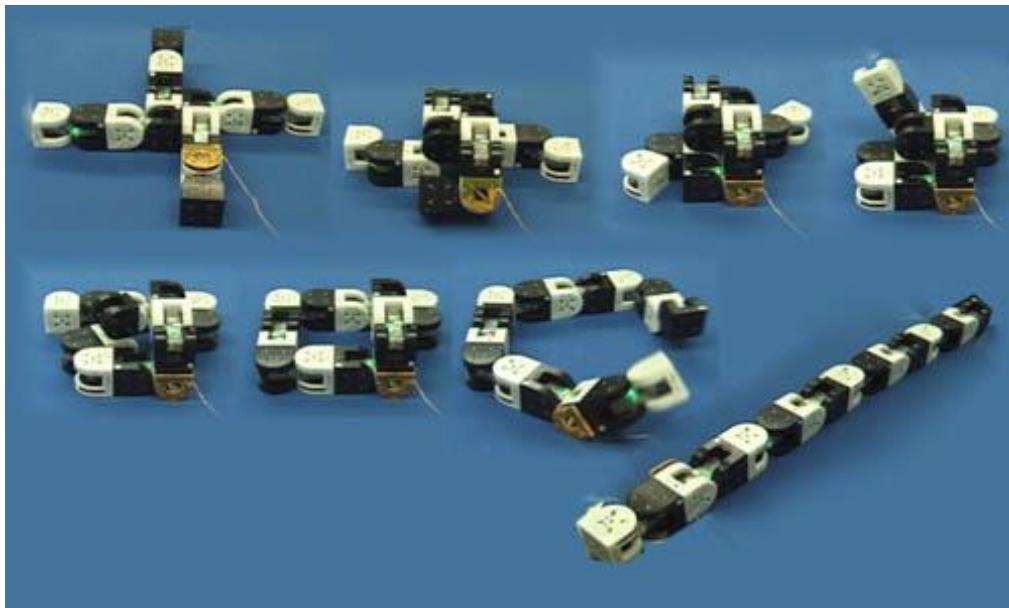
Minden egyes építőelemben egy fő- és három slave-processzor gondoskodik a részelemek vezérléséről, a modulok CAN-buszon keresztül kommunikálnak egymással.



6.1.3.4 ábra - MTRAN-III – a modulok kommunikációja CAN buszon zajlik

A moduláris robotok tervezésének két módja ismert: rács-, illetve láncalapú rendszerek. Előbbiek egységei Lego-kockák mintájára keverednek a 2D-s vagy 3D-s rács körül, utóbbiak a mindenkor szituáció függvényeként egymással összekapcsolódó, szétváló robotkarokra hasonlítanak. A rácsrendszerek tökéletesen alkalmasak az újrakonfigurálódásra, viszont a mozgásban kevésbé kiválóak. Másrészről, a láncrendszerek nagyszerűen generálnak mozgást,

ám az önmaguk által történő újrakonfigurálódás nem tartozik az erősségeik közé. A rácsrendszer fix koordinátái korlátozzák a mobilitást, ugyanakkor ideálisak új alak megformálásakor. A másik esetben a karok mozognak, tárgyakat fognak meg, löknek tovább, azaz dinamikus tevékenységekben jeleskednek. Ezzel szemben pozícióik koordinálása felettesebb nehéz.



6.1.3.5 ábra - MTRAN-III cellavariációk

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 22. oldal
--	---	--

6.2 Mobil robotok tájékozódása

John J. Leonard és Hugh F. Durrant-Whyte az ismert robottudósok a mobil robot navigáció problémakörét 1991-ben az alábbi 3 kérdésben foglalták össze:

- Hol vagyok ? (*Where am I ?*)
- Hova megyek ? (*Where am I going ?*)
- Hogyan juthatok el oda ? (*How should I get there ?*)

(Dr. Johann Borenstein, a Michigan Egyetem Mobil Robot Laboratóriumának neves professzora az első kérdést választotta híres könyve címéül is.)

A stabil robotokkal szemben a környezetében szabadon mozgó mobil robot egyik legnehezebb feladata térbeli helyzetének (pozíciójának, orientációjának) folyamatos meghatározása és mozgásának ennek megfelelő koordinálása.

A mobil egységek helyzetének meghatározási módszerei két fő csoportba oszthatók:

1. Relatív pozíció mérési módszerek

A robotika világában csak *dead-reckoning* (az angol *deduced reckoning* = levezetett, leszármaztatott helyzet-megállapítás) összefoglaló néven ismeretes módszer egy, a robot korábbi ismert pozíciójából kiinduló, sebessége és pályája ismeretében számítással meghatározott eljárást takar. A *dead-reckoning* módszer a relatív pozíciómérésen alapuló módszerek között ma már nem tekinthető egy kizárolagos navigációs eljárásnak, a modern inerciális navigációs eljárásokat is egyre elterjedtebben használják. Valamennyi módszer nagy hátránya, hogy mivel az új pozíció kizárolag az előző pozíció alapján kerül meghatározásra, az eljárás halmozott hibát hordoz magában, így a számítás pontatlansága az idővel arányosan nő.

a) Odometria

(Az *odometria* szó a görög *hodos* = utazás, kirándulás, és a *metron* = mérés szavak összetételéből származik)

Az odometria a robot pályája mentén történő elmozdulás mértékének meghatározása a fedélzeten elhelyezett „odométerek” jelei alapján. Az elmozdulást leggyakrabban inkrementális adók (enkóderek), vagy abszolút kódadók segítségével mérik, ismerve a kerekek szögelfordulását és a kormányzás orientációját ebből számítással levezethető a robot pozíciója a következő időpontban. Az odometria nagy előnye, hogy az említett adatok ismeretében mindenkor képes a robot új pozíciójának a becsülésére. Hátránya, hogy integrális jellege miatt a pozíciót számítás hibája minden határon túl nőhet, a hiba adott határon belül tartása adott időnként periódikusan független referencia segítségével megoldott pozícióhitelesítést követel meg.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 23. oldal
--	---	--

b) Inerciális helymeghatározás

Ez a módszer gyorsulásmérőket és giroszkópokat használ a mindenkorai gyorsulás és szögsebesség mérésére, majd ezek (kettős) integrálásával állítja elő az új pozíció és orientáció értékeit. A módszer előnye ismét az önálló, zárt alakban megadható számítási eljárás, a mindenkorai mérési zaj azonban az integráló számítások következtében időben korlátlanul növekvő mérési hibát eredményez. Az inerciális szenzorok csak korlátozottan alkalmazhatók pontos helymeghatározásra, ezen kívül hátrányt jelent viszonylag magas áruk is. A nagypontosságú, repülőgépeken használt mechanikus giroszkópok igen drága berendezések, viszont a száloptikát használó lézeres giroszkópok – melyek szintén igen pontosaknak tekinthetők – ára az utóbbi időben jelentősen csökkent, így kiváló megoldásként jelennek meg a mobil robotok helymeghatározó rendszereiben.

2. Abszolút pozíció mérési módszerek

a) Aktív jelzőegység (beacon)

A módszer a robot abszolút pozícióját 2, 3 vagy több aktív jelzőegység (beacon) jelének beérkezési irányából (szögéből), vagy valamelyen módszerrel megmért távolságából számítja ki. Az általában fény-, hang- vagy rádiójeleket kibocsátó adók a környezet jól ismert pontjaiban helyezkednek el.

b) Mesterséges tereptárgyak felismerése

A módszer jól elkülönülő, mesterséges jelzéseket (tereptárgyakat) helyez el a robot környezetében. A mesterséges jelzések nagy előnye, hogy ezek általában optimálisan érzékelhetők kedvezőtlen környezeti feltételek esetén is. A beaconokhoz hasonlóan általában itt is legalább 3 jelzőegység kell „látható legyen” a pozíció számításához. A tereptárgy felismerés előnye, hogy a módszerből adódóan a pozícióhiba korlátos lesz, viszont a tereptárgy nem mindenkorai és ebből adódóan a valós időben történő helymeghatározás nem mindenkorai lehetséges. A pontszerű beaconokkal szemben a mesterséges tereptárgyak adott formájuk vagy terület jellegük. Segédinformációk, mint pl. a távolság, általában lehetővé teszik a tereptárgy tulajdonságainak meghatározását, viszont ezek általában komoly számítási teljesítményt igényelnek és meglehetően pontatlannak számíthatók.

c) Természetes tereptárgyak felismerése

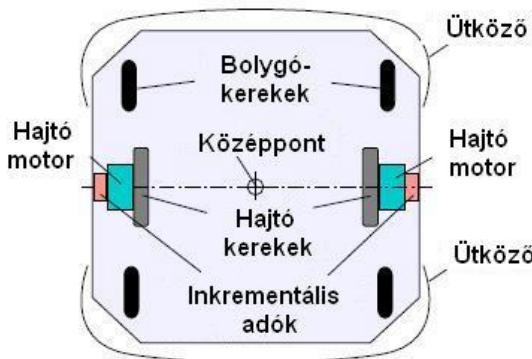
A tereptárgy ebben az esetben valamelyen megkülönböztetett jeggyel kell rendelkezzen a környezetben. Ennél a módszernél nincs szükség a környezet előkészítésére, viszont a környezet tulajdonságai előre ismertek kell legyenek. A módszer megbízhatósága kisebb, mint a mesterséges tereptárgyak esetében.

d) Modell-felismerés

Ennél a módszernél a robot saját szenzoraival információkat gyűjtünk a környezetről, majd azokat a környezet térképével vagy a modelljével összehasonlítjuk. Amennyiben a szenzorok által érzékeltek környezet és annak modellje egyeznek, a jármű abszolút pozíciója becsülhető.

6.2.1 Odometria

Amint már említettük, a mobil robotok esetében elterjedten alkalmazzák a differenciális hajtást, a két függetlenül hajtott kerék mellett harmadik, vagy negyedik kerékként alkalmazott bolygókerékek alapvetően csak támasztási célt szolgálnak. A korábban más bemutatott ábrának megfelelően az ilyen robottípusok jellegzetes kinematikai felépítése a következő:



6.2.1.1 ábra – Differenciális hajtómű mobil roboton

Odometria céljára inkrementális adókat (ritkábban abszolút kódadókat) alkalmazunk a kerekerek szögelfordulásának és a kormányzás orientációjának méréséhez, ebből számítással vezetjük le a robot pozícióját a következő időpontban. Ezekkel az érzékelő típusokkal részletesen foglalkoztunk a stabil robotok kapcsán is a 2.4 fejezetben, így itt további részletekre már nem térünk ki.

A középpont elmozdulásának számításához vezessük be a következő jelöléseket:

- k az inkrementális adó és a hajtókerék által megtett út közötti átszámítási faktor
- D a hajtókerék átmérője
- C_{enc} az enkóder által egy körülfordulás alatt szolgáltatott inkrementek száma
- $n_{áttétel}$ az enkóder és a hajtókerék közötti mechanikus áttétel arány

Fentiekkel az inkrementális adó számlálóegysége által érzékelt inkrementek száma (vigyázat: ez nem feltétlenül egyezik meg az impulzusok számával!) és a hajtókerék által megtett út közötti átszámítási faktor a következő:

$$k = D\pi / (n_{áttétel} \cdot C_{enc}) \quad (6.2.1)$$

Ennek segítségével a bal- és jobboldali kerék által megtett út a következőképpen számítható:

$$\Delta s_{1/r,i} = k \cdot N_{1/r,i} \quad (6.2.2)$$

A robot hajtásközéppontja (a hajtókerékek tengelyének felezőpontja) által megtett út:

$$\Delta s_i = (\Delta s_r + \Delta s_l) / 2 \quad (6.2.3)$$

A robot orientációjának megváltozása a bázis-koordinátarendszerben:

$$\Delta\varphi_i = (\Delta s_r - \Delta s_l) / W \quad (6.2.4)$$

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 25. oldal
--	---	--

ahol W a két hajtókerék talajjal való érintkezési pontjainak távolsága. A robot jelenlegi abszolút orientációja értelemszerűen a

$$\varphi_i = \varphi_{i-1} + \Delta\varphi_i \quad (6.2.5)$$

míg jelenlegi (abszolút) helyzete a

$$\begin{aligned} x_i &= x_{i-1} + \Delta s_i \cdot \cos \varphi_i \\ y_i &= y_{i-1} + \Delta s_i \cdot \sin \varphi_i \end{aligned} \quad (6.2.6)$$

egyenletek alapján számítható.

A fenti megoldás azt a közelítést alkalmazza, hogy az egyes időszeletek alatt a robot először egy helyben fordul, azután egyenesen halad. A gyakorlat szempontjából jobb megoldást jelent, ha egy másik közelítést alkalmazunk, nevezetesen a robot először tegye meg a kiszámított elfordulás felét, majd haladjon egyenesen, végül tegye meg az elfordulás másik felét is:

$$\begin{aligned} \varphi_i &= \varphi_{i-1} + \frac{\Delta\varphi_i}{2} \\ x_i &= x_{i-1} + \Delta s_i \cdot \cos \varphi_i \\ y_i &= y_{i-1} + \Delta s_i \cdot \sin \varphi_i \\ \varphi_i &= \varphi_i + \frac{\Delta\varphi_i}{2} \end{aligned} \quad (6.2.7)$$

A dead reckoning algoritmus egyszerűen levezethető egyenleteken alapul, a kapott algoritmus egyszerűen megvalósítható. A módszer azt feltételezi, hogy a kerekek elfordulása pontosan áttranszformálható kerekek által megtett úttá. Ez a feltételezés azonban csak korlátozottan érvényes, amelynek számos hiba az oka. A hibákat két csoportba, az úgynevezett szisztematikus, és nem szisztematikus hibák csoportjába sorolhatjuk.

Szisztematikus hibát okoz:

- a kerekek egymástól eltérő átmérője,
- a kerekek tényleges átmérőjének eltérése a névlegestől,
- a kerekek tényleges távolságának eltérése a névlegestől,
- a kerekek ideálistól eltérő elhelyezkedése,
- az enkóderek véges felbontása,
- az enkóderek mintavételezése.

Nem szisztematikus hibát okoz:

- a talaj egyenetlensége,
- a kerekek csúszása
 - a síkos talaj miatt,
 - a túlságosan nagy gyorsulás miatt,
 - külső hatások (pl. ütközés) miatt,
- a kerekek érintkezési pontjának bizonytalansága.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 26. oldal
--	---	--

Az, hogy a mely hibák okoznak nagyobb problémát, mindenkor az aktuális kialakítástól és a terep sajátosságaitól függ. Sima, egyenletes talajon haladó robot esetén általában a szisztematikus hibák jelentősebbek, míg egyenetlen terepen a nem szisztematikus hibák hatása kerül előtérbe. A robot által megtett út növekedésével a kiszámított pozíció és orientáció hihetősége egyre inkább csökken, és ebben mindenkor hibatípus szerepet játszik. A nem szisztematikus hibákkal ellentétben a szisztematikus hibák hatása nem csak megbecsülhető, de kalibrációs eljárások segítségével jelentős mértékben csökkenthető is.

A dead reckoning számítások alapjául szolgáló szögelfordulás-elmozdulás információkat sok esetben célszerű passzív (esetleg segéd-) kerek által szolgáltatott információkból levezetni. Komoly veszélyt jelent ugyanis a számítások pontosságára a hajtott kerekeknek induláskor és fékezéskor a talajon történő esetleges megcsúszása (pl. kipörgés, ütközés), melyek előbb relatív, majd a módszer kumulatív jellegéből adódóan abszolút pozíciószámítási hibában jelentkeznek.

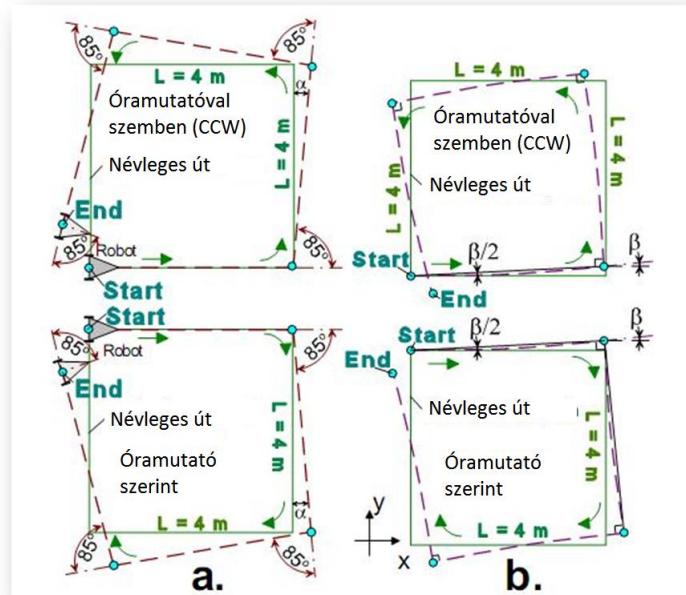
A michigani egyetemen kifejlesztett *UMBmark* (University of Michigan Benchmark) kalibrációs eljárás képes a kerekek eltérő átmérőjéből (D) és a nominálistól eltérő keréktávolságból (W) származó szisztematikus hibák (E_d , illetve E_w) jelentős mértékű csökkentésére. Az említett kétféle hibát a következő módon definiálhatjuk:

$$E_d = \frac{D_r}{D_l} \quad (6.2.8)$$

$$E_w = \frac{W_{act}}{W_{nom}} \quad (6.2.9)$$

Itt D_r és D_l a jobb, illetve bal kerék átmérőjét, W_{nom} a névleges, W_{act} pedig a tényleges keréktávolságot jelentik.

A kalibrációs eljárás során egy előre beprogramozott, adott oldalhosszúságú (minél hosszabb, úgy a kalibráció hatásfoka is jobb lesz) négyzet alakú pályán kell végighaladnia a robotnak, az óramutató járásával megegyező (CW), és azzal ellentétes irányban (CCW) egyaránt. A robotnak a saját dead-reckoning algoritmusa szerint kell egyenesen haladnia oldalhosszúságnyi távolságot, illetve saját mérése szerint kell fordulnia derékszögben, nem pedig a valós egyenes és derékszög szerint. A műveletet többször el kell végezni, és minden esetben fel kell jegyezni a kiindulási és a végső helyzet közötti eltérést. A névlegestől eltérő keréktávolság kanyarodáskor szöghibát (úgynevezett A típusú hibát) okoz, a kerekek eltérő méréte miatt pedig nem egyenes vonalú a mozgás a négyzet oldalain (B típusú hiba). Ezt a két esetet szemlélteti a következő ábra.



6.2.1.2 ábra – UMBMark – A és B típusú hibák

A két hiba ugyan egyszerre jelentkezik, de geometriai megfontolások alapján egymástól különválasztható a következő módon:

$$\begin{aligned}\alpha_x &= \frac{\bar{x}_{cw} + \bar{x}_{ccw}}{-4L} \\ \beta_x &= \frac{\bar{x}_{cw} - \bar{x}_{ccw}}{-4L}\end{aligned}\quad (6.2.10)$$

Itt α_x az A típusú hibák 90°-os forduláskor okozott szöghibája fokban, β_x a B típusú hibák miatt a négyzet oldalainak végén jelentkező orientáció hiba. Az \bar{x}_{cw} a kiindulási helyzethez képest mért átlagos x irányú eltérés az óramutató járásával megegyező, \bar{x}_{ccw} pedig az azzal ellentétes irányban, L a körüljárt négyzet oldalának hossza. A fenti mennyiségek az y irányú eltérések segítségével is meghatározhatók az előzőekhez hasonlóan:

$$\begin{aligned}\alpha_y &= \frac{\bar{y}_{cw} - \bar{y}_{ccw}}{-4L} \\ \beta_y &= \frac{\bar{y}_{cw} + \bar{y}_{ccw}}{-4L}\end{aligned}\quad (6.2.11)$$

A tényleges szöghibáknak az x és y irányú eltérésekből adódó értékek átlagát tekintjük:

$$\begin{aligned}\alpha &= \frac{\alpha_x + \alpha_y}{2} \\ \beta &= \frac{\beta_x + \beta_y}{2}\end{aligned}\quad (6.2.12)$$

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 28. oldal
--	---	--

Az eddigi eredményeket felhasználva meghatározhatjuk a B típusú hibák miatti íves pálya R sugarát:

$$R = \frac{L/2}{\sin(\beta/2)} \quad (6.2.13)$$

Ezzel pedig a keresett hibafaktorok meghatározhatók:

$$E_w = \frac{W_{act}}{W_{nom}} = \frac{\frac{\pi}{2}}{\frac{\pi}{2} - \alpha} \quad (6.2.14)$$

$$E_d = \frac{D_r}{D_l} = \frac{R + \frac{W_{nom}}{2}}{R - \frac{W_{nom}}{2}} \quad (6.2.15)$$

Így a fenti E_d illetve E_w hibatényezők, a $D_{nominal}$ névleges kerékátmérő, valamint a $W_{nominal}$ névleges keréktávolság felhasználásával a következőképpen kaphatjuk meg a keresett kalibrált értékeket:

$$D_r = \frac{2}{\frac{1}{E_d} + 1} \cdot D_{nom} \quad (6.2.16)$$

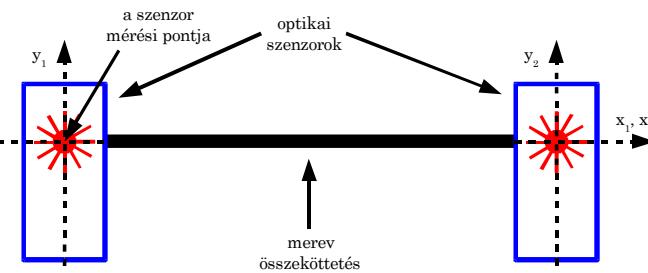
$$D_l = \frac{2}{E_d + 1} \cdot D_{nom} \quad (6.2.17)$$

$$W_{act} = W_{nom} \cdot E_w \quad (6.2.18)$$

Ezeket a lépésekkel elvégezve kompenzáljuk a kerekek átmérójének eltérést, és pontosítjuk a keréktávolságot, így a robotunk valóban egyenes úton fog közlekedni és megfelelő szögben fog elfordulni a négyzet bejárása során.

6.2.2 Odometria optikai szenzorokkal

Kerekeken guruló mobil robotok esetén általában alkalmazott módszer az előző fejezetben tárgyalt odometria, de adóhatnak olyan esetek is, amikor pl. a kerekek vastagsága, lánctalpas hajtás alkalmazása illetve nem megfelelően pontos mechanika esetén ez nem járható út. Ilyen esetekben felmerülhet optikai szenzorok alkalmazása, amelyekkel a talaj elmozdulása detektálható (pl. lézeres optikai egérben alkalmazott eszközök). Egy ilyen elrendezést mutat a következő ábra:



6.2.2.1 ábra – *Odometria optikai szenzorokkal*

A két optikai szenzor képes x és y irányban is érzékelni az elmozdulást. A szenzorok mérési pontjait differenciális meghajtású robot esetén a kerekek érintési pontja által definiált egyenesbe helyezve az y irányú elmozdulások segítségével egyszerűen megvalósítható a dead-reckoning algoritmus.

A fenti elrendezés természetesen bárhova elhelyezhető a robozon, hiszen képes x és y irányú elmozdulás mérésére is, így ha ismert a szenzorok egymástól való távolsága, és pozíciójuk és orientációjuk a robohoz rögzített koordinátarendszerben, akkor a mért x és y irányú elmozdulásokból a robot pozíciója számítható.

6.2.3 Inerciális helymeghatározás

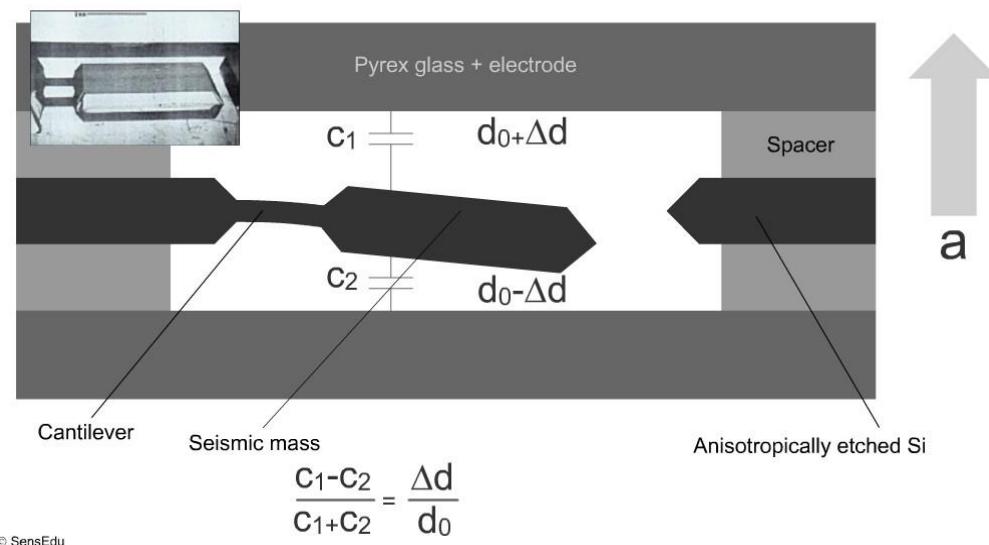
Az inerciális (tehetetlenségi) navigációs rendszerek működésének elve olyan fizikai törvényeken alapszik, melyek alkalmazásával külső segítség nélkül vagyunk képesek a hordozóeszköz helyzetének, haladási irányának és sebességének megállapítására. A rendszert a hordozóeszköz egy ismert helyzetében elindítva, a mozgás közben bekövetkező és mért gyorsulások és elfordulások mérése alapján számítjuk az aktuális helyzetet. A gyorsulásmérésen alapuló módszer 2 vagy 3 dimenziós gyorsulásmérő segítségével kíséri figyelemmel robotunk mozgását, ebből az információból kétszeres integrálással állíthatja elő a pozíció információt. A légi és a tengeri irányításban régóta alkalmazott giroszkóp segítségével egy adott irányhoz képesti szögelfordulás határozható meg, ez az orientáció pontosabb meghatározását teszi lehetővé. A MEMS (Micro Electro-Mechanical Systems) technológia elterjedésével napjainkban már elérhető áron beszerezhetők olyan szenzorok, amelyeket néhány évvel ezelőtt csak a repülőgépgyártásban, űrkutatásban voltak használatban magas áruk miatt. A giroszkópokat, gyorsulásérzékelőket ma már gyakran alkalmazzák az autóiparban, robotikában és egyéb technológiák mozgásának vezérlésének megvalósításában.

A gyorsulásérzékelőknek alapvetően két típusát különböztetjük meg:

a) Felületi mikromechanikai eljárással készült, kapacitív elvű gyorsulásszenzorok

Ezt a szenzortípust főleg a nagyobb gyorsulás illetve lassulásértékek (50...100 g) mérésére használják (pl. ütközésdetektálás, utas védelmi légszsák rendszerekben), de a kialakításuk és működési elvük miatt alkalmasak egészen kis gyorsulásértékek mérésére is. A mérési frekvencia 0 Hz-től akár több kHz-ig is terjedhet (azaz lehetőség van statikus mérésre is).

Ezen szenzorok sokkal kisebbek, mint a tömbi mikromechanikai eljárással készült szenzorok. A szükséges rugó-tömeg rendszer egy külön eljárással a szilíciumszelet felületén kerül kialakításra. A jelkondicionáló és a kiértékelő elektronika szinte kivétel nélkül ugyanazon a szilíciumlapkán, közvetlenül a szenzorelem mellett helyezkedik el. Mivel a szeizmikus tömeg rugókra van felfüggesztve, ezért egy egyenes vonalú, a mérési irányba történő gyorsulás megváltoztatja a távolságot a mozgó és az álló elektródák között, ami viszont kapacitásváltozást okoz a C1 és a C2 kondenzátorokban.



6.2.3.1 ábra – Kapacitív elven működő gyorsulásérzékelő (SensEdu)

Ez a kapacitásváltozás az elektromos jellemzők megváltozását idézi elő, amit a kiértékelő elektronika segítségével szűrnek, erősítenek, és esetlegesen átalakítanak. A nagyon kicsi (kb. 1 pF) kapacitás miatt a kiértékelő elektronikát a szenzorhoz a lehető legközelebb kell elhelyezni, ezért szokás azt közvetlenül ugyanarra a szilíciumszeletre integrálni (ún. „one chip design”).

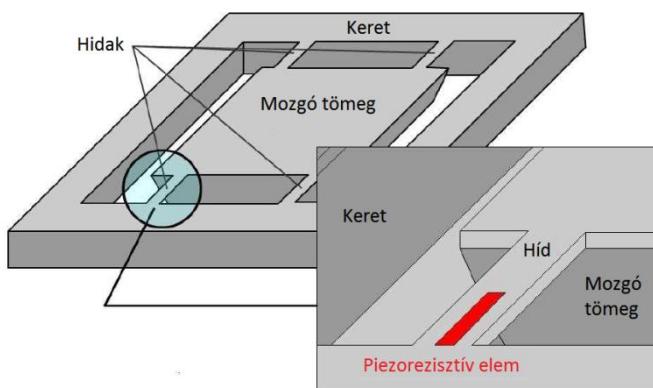
b) Piezoreiszstív elven működő szenzorok

A gyorsulás, illetve az abból következő erő mérésére régóta alkalmazott elv valamely rugalmas test alakváltozásának, illetve az ezzel összefüggésbe hozható valamely rész elmozdulásának mérése. Alapja a jól ismert Hooke-törvény, amely szerint a rugalmassági határig az erő és az általa okozott deformáció egymással lineáris kapcsolatban van. Ez az erzékelő első átalakítása. A második átalakítás, hogy ezt a deformációt valamely eszköz segítségével villamos jel képzésére használjuk.

Régóta ismert jelenség, hogy az ellenállások mértéke mechanikai feszültség vagy deformáció hatására megváltozik. Ez a megváltozás két részből áll össze: egyrészt megváltozik az ellenállás alakja (hossz-, illetve keresztmetszet-változás) másrészt a deformáció hatására a fajlagos ellenállás is megváltozik. Ez utóbbit piezoreiszstív effektusnak nevezzük. Itt is természetesen biztosított a statikus mérés lehetősége. A legjobb hatást úgy érhetjük el, ha négy darab piezoreiszstív elemet (mint nyúlásmérő bályeget), az ábra szerint helyezünk el. Ezek közül az egyik oldalon lévők nyújtott, míg a

másik oldalon elhelyezettek nyomott helyzetben vannak. Az elemeket hídkapcsolásba helyezzük, így a hatás már kényelmesen mérhető és a hídstruktúra miatt a hőfokfüggést is (részben) kompenzáltuk.

Ezt a szenzort úgy készítik, hogy egy szilíciumtömbből anizotróp maratás segítségével a fent látott struktúrát alakítják ki. A középen elhelyezkedő szeizmikus tömeget a kerettel négy híd köti össze, a tömeg tulajdonképpen ezeken a hidakon „lóg”. A hidak annyira vékonyak, hogy a tömeg a gyorsulás irányába (ami ebben az esetben a függőleges irány) kicsit elmozdulni képes. A piezorezisztív elemeket (minden hídra egyet, azaz összesen négy darabot) a hidakra helyezik el.

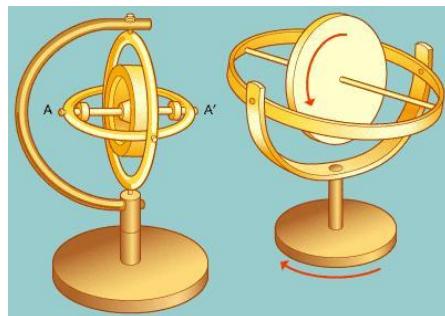


6.2.3.2 ábra – Piezorezisztív elven működő gyorsulásszenzor (SensEdu)

Giroszkópok

Az egyenes vonalú egyenletes mozgás tényének és a mozgás sebességének megállapítása egy teljesen zárt jármű belsejében semmiféle fizikai méréssel nem lehetséges. Az inerciális módszer szerinti eljárás is igényel bizonyos kapcsolatot a jármű műszerei és a környezet között. Ez a kapcsolat földi viszonyok között leggyakrabban a környezeti közeghez viszonyított sebesség mérése. A görbe vonalú mozgások szögsebességi összetevői és a jármű térfelületi elfordulásai azonban teljes mértékben meghatározhatók a külvilágtól független méréssel.

Az ilyen mérések egyik alapvető eszköze a pörgettyű. Mint az a fizikából ismeretes a pörgettyű lényeges sajátsága, hogy forgástengelyének térbeli irányát megindítása után igyekszik megtartani (perdület megmaradás törvénye). Amennyiben tömege és fordulatszáma elegendően nagy, felfüggesztése a tér bármely irányába könnyű elfordulását biztosítja, és súrlódási veszteségeit pótolva fordulatszámát stabilizáljuk, úgy a pörgettyű tartósan megőrzi forgástengelye eredeti irányát. A felfüggesztési követelményt az ún. 3 szabadságfokú erőmentes pörgettyűk kielégítik, ezeknél a forgó tömeg súlypontja egybeesik a három ortogonális elfordulási tengely metszéspontjával. A fordulatszám tartása a veszteségek ellenére leggyakrabban azért érhető el, mert a forgó tömeg egyben egy elektromotor forgórésze. Ez a megoldás igen precíz fordulatszám-tartást tesz lehetővé.



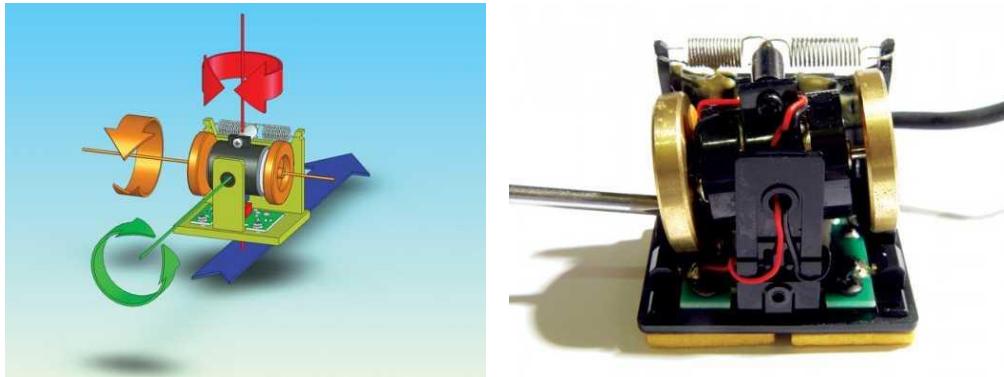
6.2.3.3 ábra – A mechanikus giroszkóp elve

Egy tetszőleges járműben felfüggesztett 3 szabadságfokú pörgettyű a járműnek környezetéhez képest bekövetkezett térfelváltásra jellemző hibajelet szolgáltat, alapállapotnak a pörgettyű megindításakor fennállott állapotot tekintve. Az ilyen pörgettyű egy kitüntetett térirányt őriz meg navigációs felhasználás esetén. A giroszkóp olyan komplett műszer, mely a pörgettyű fenti sajátosságát kihasználva a jármű kívánt térirányba állításához és annak megtartásához szükséges szabályozáshoz használható, iránykomponensek szerint bontott hibajelet szolgáltat. A pörgettyű üzemének kiszolgáló egységei is a giroszkóp részei. Nem beszélünk giroszkópról, ha a pörgettyű nem a kívánt pályával kapcsolatos kitüntetett térirányt őriz.

A giroszkóp (Foucault, 1852) annál pontosabban őrzi meg a kívánt kitüntetett térirányt, minél nagyobb a pörgettyű tömege és fordulatszáma. A forgó tömeg növelésének elsősorban a felfüggesztési nehézségek szabnák korlátot. A jármű gyorsulásai és az elkerülhetetlen rázó, ill. ütő igénybevételek a precíziós felfüggesztések számára komoly igénybevételt jelentenek. Pilóta nélküli repülő eszközökönél, (pl. a rakétaknál) nem ritka a 100 g-t is meghaladó rövididejű gyorsulás, 10 g nagyságrendű gyorsulások pedig akár huzamosabban, vagy ismétlődően előfordulhatnak. A tömeg növelésének határt kell szabni a gravitációs erőter változásaira való érzékenység megnövekedése miatt is. A Föld gravitációs erőtere nem homogén és ezt az interkontinentális ballisztikus rakéták, vagy lassú, de hosszan utazó hajók giroszkópjainál nem lehet figyelmen kívül hagyni.

A fordulatszám növelésével sajnos előtérbe kerülnek a súrlódási problémák, és a műszer élettartamának csökkenésével kell számolni. A ballisztikus rakéták giroszkópjai esetében a rövid élettartam nem okoz problémát, hiszen a rakéta repülési (és ezzel a giroszkóp működési) ideje néhány perc vagy néhány 10 perc. Ezeknek a rakétaknak a fedélzeti navigációs rendszerében igen nagy fordulatszámú pörgettyűket alkalmaznak (50-100 000 fordulat/ perc), emellett a rövid üzemi élettartam és a repülés végén való biztos megsemmisülés miatt a viszonylag olcsó konstrukció igénye indokolt. Más a helyzet a nagy hatósugarú repülőgépek és hajók navigációs műszereivel, így giroszkópjaikkal is. Ezek esetében a hosszú üzemi élettartam, a nagy pontosság és a nagy megbízhatóság igénye együttesen jelentkezik. Hasonló a helyzet egyes űrhajók és automata űrszondák esetében is. Az élettartam és a megbízhatóság megkívánt értékeinek biztosítását nagyobb tömegű és lassabban pörgetett giroszkópok teszik lehetővé. Egy nagy hatósugarú repülőgép fedélzeti giroszkópja a hozzá tartozó műszerrendszerrel együtt 50-100 000 \$ értékű, amely jól érzékelteti az ilyen műszerek speciális anyag- és munkaigényét.

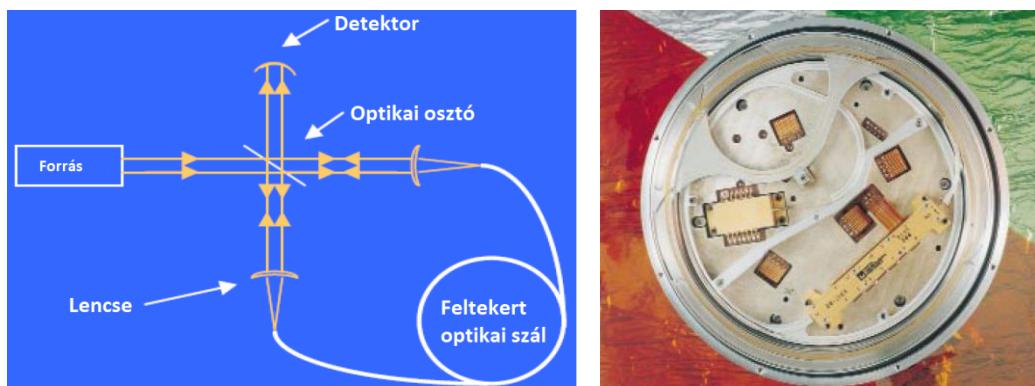
A pörgettyűs (mechanikus) giroszkóp pontossági jellemzőit a pörgettyű tömegének és fordulatszámának csökkentése alapvetően rontja. Ezért sokáig úgy tűnt, hogy ezek a giroszkópok „veleszületett” problémái, amelyek a magas árral együtt elkerülhetetlenek.



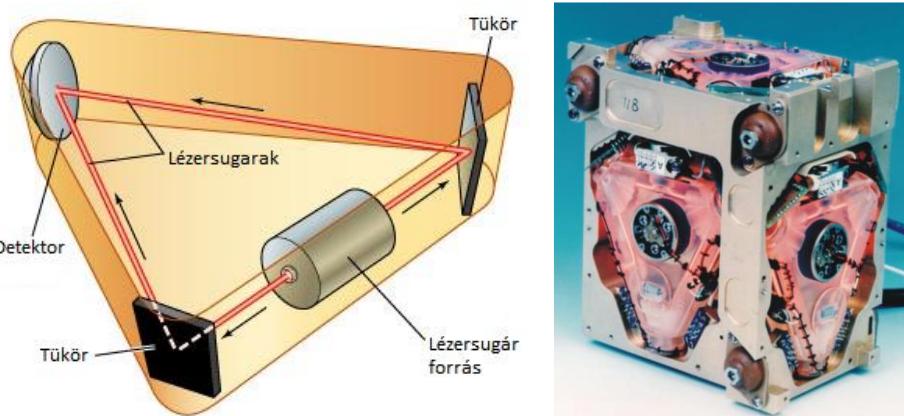
6.2.3.4 ábra – Elektromechanikus giroszkóp

A mechanikus giroszkópok helyett manapság már inkább a nagyobb pontosságú optikai (lézer) giroszkópokat használják. Az optikai giroszkópok működésének alapját a relativitáselmélet egyik alapkísérlete, a Sagnac-kísérlet (1913) képezi. Két fő változata az üvegszál optikát használó FOG (Fiber Optic Gyroscope) és a sokszög alakban elrendezett tükrökön alapuló gyűrűs RLG (Ring Laser Gyroscope).

Az üvegszálas megoldásnál az ellentétes irányban terjedő fényhullámok egy feltekercselt optikai szálban terjednek. A félvezető lézerből egy osztó segítségével jut a fény az optikai szál két végébe. Nyugvó helyzetben a nyalábok azonos fázisban érkeznek a szál ellentétes végeire. Forgás esetén azonban a forgás irányában haladó hullámnak hosszabb, a vele szemben haladónak rövidebb utat kell megtennie, aminek következtében fáziskülönbség adódik. Ez a fáziskülönbség arányos a szögsebességgel és az interferencia képből detektálható.



6.2.3.5 ábra – Az üvegszál optikás giroszkóp (FOG) elve és megvalósítása



6.2.3.6 ábra – A gyűrűs lézer giroszkóp (RLG) elve és megvalósítása

Az RLG giroszkópok mérési elve is a Sagnac effektuson alapul. A három- (vagy több-) szög alakú rezonátorban az óramutató járásával megegyező és ellentétes irányban két lézernyaláb kering. A háromszög sarkaiban tükrök vannak, kerülete pedig a lézer hullámhosszának egész számú többszöröse. Álló helyzetben a két nyaláb frekvenciája megegyezik. Ha azonban a rendszer forog, pl. az óramutató járásával megegyező irányban, az ugyanebben az irányban keringő fotonuktól a tükrök „távolodnak”, vagyis valamivel hosszabb utat kell megtenniük, mint az álló helyzetű rezonátorban - a hullámhosszuk tehát megnő. Az óramutató járásával ellentétesen keringő fotonok viszont „közeledő” tükrökkel találkoznak, vagyis rövidebb utat tesznek meg egy kör alatt, tehát a hullámhosszuk rövidebb lesz. Kimutatható, hogy a két fény sugár frekvenciák különbsége egyenesen arányos a szögsebességgel (ω) és fordítva arányos a fény hullámhosszával (λ):

$$\Delta f = \frac{4F\omega}{l\lambda} \quad (6.2.19)$$

ahol l a fény sugár által megtett út és F a fényút által körülzárt terület.

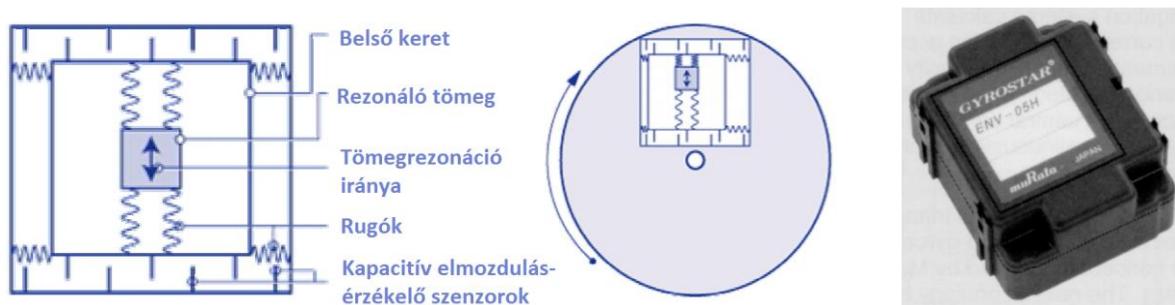


6.2.3.7 ábra – Optikai giroszkópok a valóságban

Az optikai megoldásokon alapuló giroszkópok óriási előnye, hogy frekvencia mérésével teszik lehetővé a szögsebesség meghatározását. A frekvenciamérés napjainkban a legnagyobb pontosságú mérések közé tartozik, mert időmérésre vezethető vissza. Az időmérés eljárásai kifinomultak, etalonjai nagy pontossággal előállíthatók. A két körüljárási irány

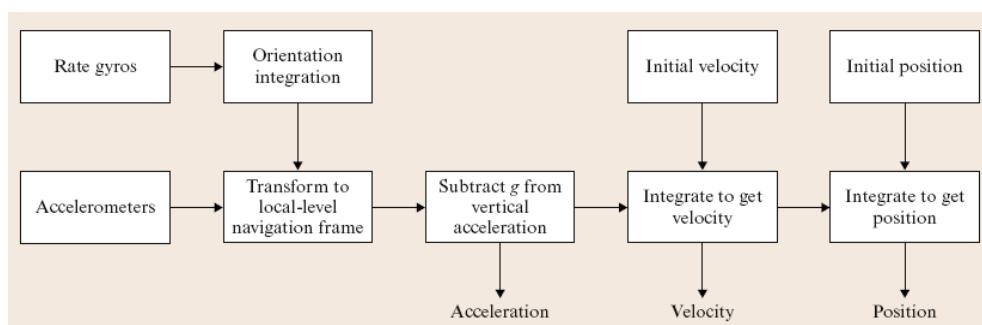
fénysugarainak frekvenciakülönbsége pl. álló fotokatódról a lebegési frekvencia mérésével meghatározható. Két fotokatóddal és logikai áramkörrel a frekvencia eltérés előjele is meghatározható. Gyakorlatilag a lebegési frekvenciát fotokatódon észlelhető jel maximumának alapján impulzusszámlálóval mérik. Ekkor a számláló közvetlenül az elfordulási szögre jellemző értéket mér. A zárt fényútba behelyezett lézerrel a szögsebesség, ill. szögelfordulás mérése a külvilágtól teljesen elzárva végzett időmérésre vezethető vissza.

A giroszkópek ára lényegesen lecsökkent a MEMS technológiá megjelenésével. Az erre a technológiára épülő vibrációs giroszkópek a Coriolis erő hatását érzékelik. Mint azt a fizikában tanultuk, a **Coriolis-erő** az inerciarendszerhez képest forgó (tehát egyben gyorsuló) vonatkoztatási rendszerben mozgó testre ható egyik tehetetlenségi erő. A mechanikus giroszkópokban alkalmazott, felpörgetett tehetetlen tömeg elfordításánál fellépő Coriolis erő érzékelhető rezgő (transzlációs) mozgást végző tömeg esetében is. Ezt az elmozdulást kapacitás változás mérésére vezetjük vissza. A MEMS giroszkóp esetében a mozgó test egy mikroelektronikai technológiával szilíciumhordozón kialakított rezgő tömeg. Amikor a hordozó és a hozzá rögzített rezgő mechanikai rendszer forog, arra a szögsebességgel arányos, a rezgés irányára merőleges irányú erő hat így a tömeg a rezgés irányára merőlegesen elmozdul. Ez az elmozdulás kapacitív úton mérhető.



6.2.3.8 ábra – MEMS technológiára épülő vibrációs giroszkópek

Az inerciális szenzorokból napjainkban már professzionális komplex mérőegységeket állítanak össze (IMU – Inertial Measurement Units), melyek segítségével folyamatos 6 szabadságfokú helyzet-meghatározás végezhető (SUMMIT, iMAR, MEMSense, InterSense).



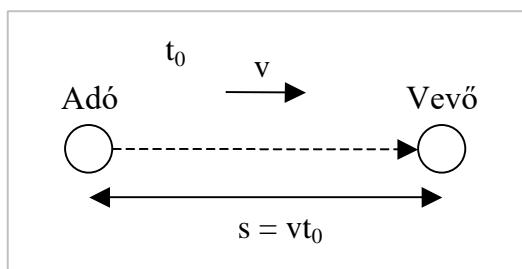
6.2.3.9 ábra – IMU blokkvázlata



6.2.3.10 ábra – 3 gyorsulásmérőből és 3 MEMS giroszkópból felépített IMU egység valóságos kialakítása

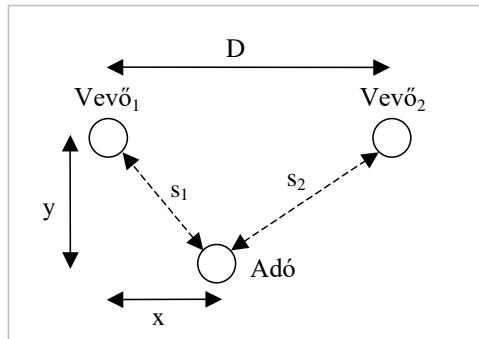
6.2.4 Az ultrahangos távolságmérés elve

Az ultrahangos távolságmérés és helymeghatározás során a távolságmérést időmérésre vezetik vissza. Az adó a $t = 0$ időpillanatban hanghullámot bocsát ki, amit a vevő - a hang véges terjedési sebességből adódóan - t_0 idő eltelté után érzékel (ez levegőben körülbelül 340 m/s). Az adó és vevő egymáshoz képest távolsága az $s = vt_0$ képlet alapján számítható.



6.2.4.1 ábra – Távolságmérés a hang terjedési idejéből

Amennyiben a két egység (adó és vevő) közötti távolságot mérní tudunk, egy további egység beiktatásával (ez lehet akár adó vagy vevő is) síkbeli pozíciót is meg tudunk határozni. A következő ábrán látható elrendezésben két egymáshoz képest rögzített helyzetű vevőegységet használunk fel az ismeretlen helyzetű adóegység helyzetének (x , y koordinátáinak) meghatározására. Ezt a módszert a szaknyelvben háromszögelésnek (a háromszög oldalhosszainak ismeretén alapuló helymeghatározásnak, **trilateráció**nak) nevezik. (A háromszögelésnek létezik egy másik változata is, amely az adó-vevő egységek egymáshoz képesti iránymeghatározásán, azaz a háromszög szögeinek ismeretén alapul – ezt a helymeghatározási módszert **trianguláció**nak nevezzük.)



6.2.4.2 ábra – A trilateráció elve

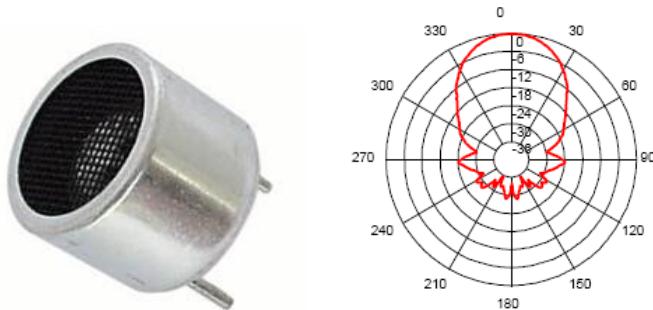
Az ábra jelöléseivel

$$\begin{aligned} x^2 + y^2 &= s_1^2 \\ (D - x)^2 + y^2 &= s_2^2 \end{aligned} \quad (6.2.20)$$

amiből levezethető, hogy

$$\begin{aligned} x &= \frac{D^2 + s_1^2 - s_2^2}{2D} \\ y &= \sqrt{s_1^2 - x^2} \end{aligned} \quad (6.2.21)$$

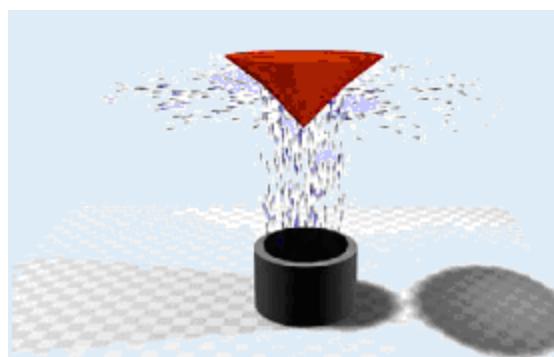
Ultrahangos távolságmérésnél általában 40-180 kHz közötti hanghullámot sugároznak ki, egy szelektív vevővel pedig érzékelik a beérkező hanghullámokat. A célra általában piezoelektronikus elven működő ún. szonár adó- és vevőkapszulákat használnak, melyek jellegzetessége, hogy egy szűk tértartományban ($\text{kb. } \pm 30^\circ$) képesek a jelek kiadására és érzékelésére. Az elemek erősen frekvencia-szelektívek, karakterisztikájuk egy-egy konkrét adó- és vevőtípus esetében csak egy szűk frekvenciatartományban érvényes (pl. $40 \pm 1 \text{ kHz}$).



6.2.4.3 ábra – Szonár adó- és a vevőkapszula, és ezek iránykarakterisztikája

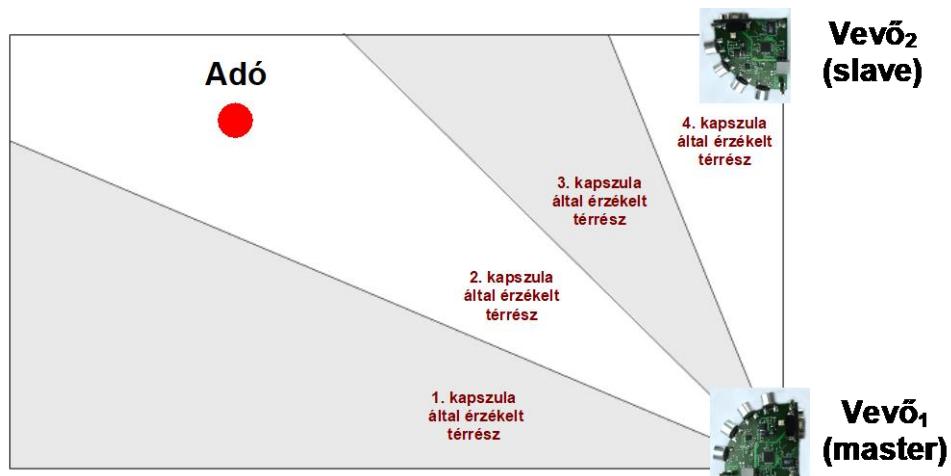
Amennyiben egy adott sík térrészben (pl. egy helyiségen) mozgó mobil robot pozíóját kívánjuk meghatározni, a feladat egy lehetséges megoldása a következő:

Legyen a két darab, egymással vezetékes, vagy rádiókapcsolatban álló vevő (továbbiakban *master* és *slave*) fixen telepítve a helyiségben, közülük a *master* rádiókapcsolatban áll a mozgó roboton elhelyezkedő adóegységgel is. Az elrendezést azért célszerű így kialakítani, mert az adó egység egyszerűen kialakítható körsugárzó karakteristikájúvá, míg a vevők általában irány érzékenyek, tehát jobban illeszkedik hozzájuk a fix telepítés. A *master* egység rádiókapcsolaton keresztül utasítja az adót adásra (a rádióhullámok terjedési ideje elhanyagolhatóan kicsi a hang terjedési idejéhez képest) a *slave*-egységet pedig az időmérés megkezdésére – ezt követően minden vevő egység méri a hanghullám megérkezésig eltelt időt. A *slave* elküldi a mért távolságadatot a *masternek*, az pedig kiszámolja a pozíciót a két távolságadat alapján. A kiszámolt pozíciót ezek után rádiókapcsolaton keresztül kapja meg a mobil robot irányító egysége.



6.2.4.4 ábra – Az irányfüggő adókapszula körsugárzóvá tétele

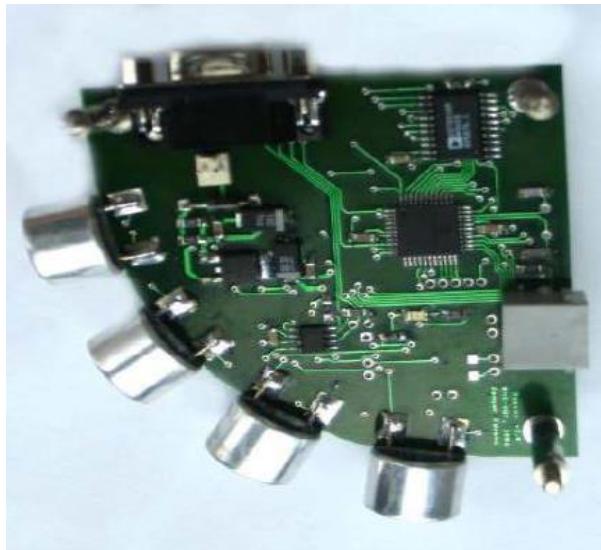
Mint már említettük, az adókapszula is erősen irány érzékeny, ennek kiküszöbölésére a robot tetején az adókapszulát úgy helyezzük el, hogy az az ultrahangot függőlegesen felfelé sugározza ki, majd közvetlenül fölé (az adókapszula tengelyébe) egy terelőkúpot helyezünk el, amely a ráérkező ultrahang hullámokat egyenletesen szétteríti a síkban. A módszerrel jó körszimmetria érhető el, viszont ez jelentős intenzitáscsökkenéssel is együtt jár.



6.2.4.5 ábra – Az irány érzékeny vevőkapszulák által felügyelt térrészek

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 39. oldal
--	---	--

Az adóhoz hasonlóan a vevők is irány érzékenyek. Ezért a fixen telepített vevőegységet több érzékelővel is felszereljük úgy, hogy az egyes vevőegységek által lefedett („látott”) térrészeken enyhén átfedjék egymást. Természetesen valamennyi vevőkapszula figyelése szükséges, az érzékelt jelek függvényében dönthetjük el, melyik térszben található éppen az adóegyége.



6.2.4.6 ábra – A 90°-os térnegyedet felügyelő vevőegység (aktiv beacon)

A mérési elv egyszerűnek tűnik, de azt meg kell említenünk, hogy a hangterjedés számunkra két legfontosabb tulajdonsága: a hangsebesség hőmérsékletfüggése és csillapítása a megtett út függvényében – amik számunkra a mérés során zavaró tényezők – mennyiben befolyásolhatja a mérési eljárás pontosságát, illetve hogyan tudjuk ezek zavaró hatását kiküszöbölni. Ezen kívül ismernünk kell a hangtér gerjesztésére használt adó által kibocsátott, valamint az érzékelésre használt vevőkapszula által érzékelt vett jel alakját, és ezek alapján egy viszonylag pontos távolságmérési módszert tudunk kialakítani.

A hangsebesség hőmérsékletfüggése

A hanghullám terjedési sebessége levegőben kizárolag a hőmérséklet függvénye. Az összefüggést a következő összefüggés adja meg:

$$c = 331.45 \cdot \sqrt{\frac{T[\text{°K}]}{273.15}} [\text{m/s}] \quad (6.2.22)$$

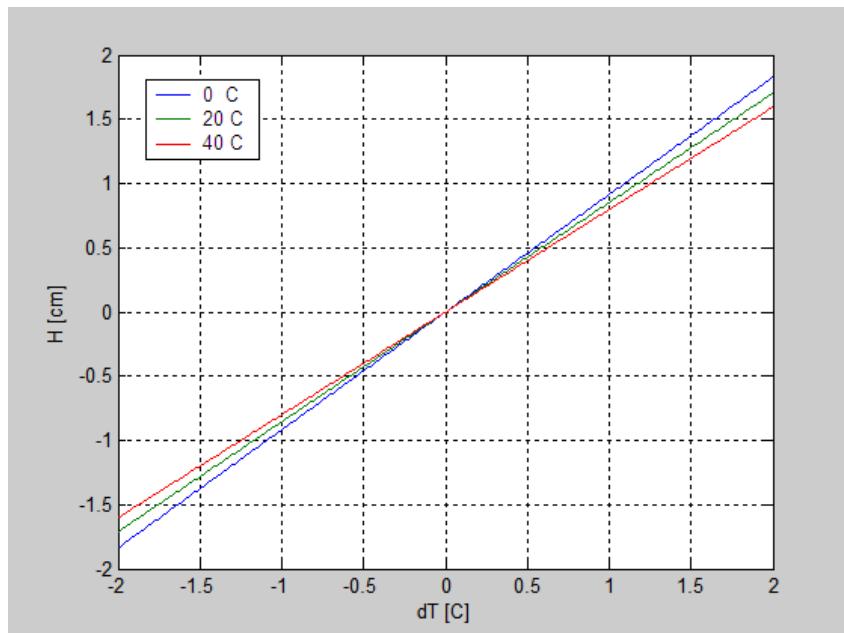
ahol T az abszolút skálán mért hőmérséklet. Vizsgáljuk meg, hogy mennyi hibát okozhat mérésünkben a hőmérsékletfüggés ! Legyen a valódi hőmérséklet T_0 , a valódi és a mért hőmérséklet közötti különbség ΔT , a mért idő t , a mért és a helyes távolság s_m, s_0 ill. a számítás során használt és a tényleges sebesség v_m, v_0 . Az abszolút mérési hiba (H) a mért és a helyes távolság különbsége:

$$H = s_m - s_0 = v_m \cdot t - v_0 \cdot t \quad (6.2.23)$$

ahol feltettük, hogy időmérés pontatlanságából adódó hiba elhanyagolhatóan kicsi a hangsebesség változása által okozott hibához képest. Behelyettesítve ebbe az eredeti összefüggést a következőt kapjuk:

$$H = c_1 \sqrt{\frac{T_0 + \Delta T}{c_2}} \cdot t - s_0 = c_1 \sqrt{\frac{T_0}{c_2}} \cdot t \cdot \sqrt{1 + \frac{\Delta T}{T_0}} - s_0 = s_0 \left(\sqrt{1 + \frac{\Delta T}{T_0}} - 1 \right) \quad (6.2.24)$$

Az eredményben látható, hogy a hiba egyenesen arányos a mérni kívánt távolsággal. Mérésünket pl. 5 méteres távolságra méretezve a mérési hiba 5 méteres távolságban ábrázolható ΔT függvényében (T_0 -val paraméterezve).

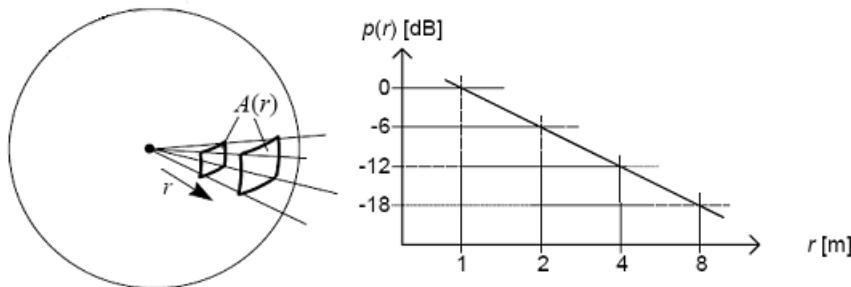


6.2.4.7 ábra – A mérés hibája a hőmérséklet eltérés függvényében

Az ábrán jól látható, hogy a választott munkapont közvetlen közelében a hiba arányosan változik a hőmérséklet különbséggel, ± 1 °C-os hőmérsékletmérési eltérés esetén körülbelül ± 1 cm-es pontossággal tudjuk mérni az objektum helyzetét (a környezeti hőmérséklettől függően). mindenépp szükség van tehát egy megfelelő pontosságú hőmérőre is a mérés során, a számított távolságoknak a tényleges hőmérséklethez tartozó hangterjedési sebesség alapján való számítására.

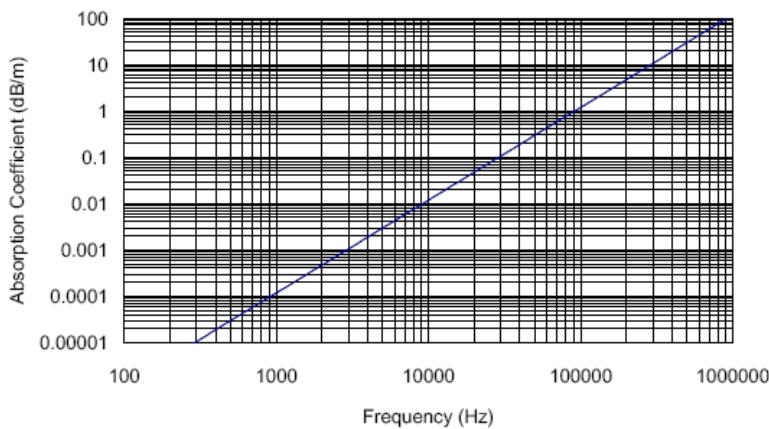
Csillapítás a levegőben

Csillapításnak nevezik a hanghullám intenzitásának terjedés közbeni csökkenését. A jelenség két okra vezethető vissza. Egyrészt a hanghullám szabad térfelület terjedésekor egyre növekvő felület mentén oszlik szét – ez a hangintenzitás $1/r^2$, a hangnyomás $1/r$ szerinti csökkenését eredményezi, ahol r a forrástól való távolság. Az energia megmaradásának elvából könnyen belátható, hogy ha ugyanaz az energia oszlik szét egy növekvő gömbfelület-részen, a hangintenzitásnak négyzetesen kell csökennie.



6.2.4.8 ábra – A hang intenzitásának csökkenése a távolság függvényében

Önmagában az ún. $1/r^2$ -es sugárzási törvény -20 dB/dekádos érzékelt jelszint csökkenést okoz a távolság függvényében. Ezen felül még abszorpciós folyamatok is gyengítik a jelet. Az abszorpciós veszteség elsősorban a frekvenciától, valamint a közeg összetételétől (por- és páratartalom) függ. Az ebből származó veszteséget dB/m egységen adják meg, de a közegösszetételektől való erős függés miatt a különböző forrásokban fellehető eredmények elég változatosak. Példaképpen az alábbi ábrán egy ilyen mért jelleggörbe látható.



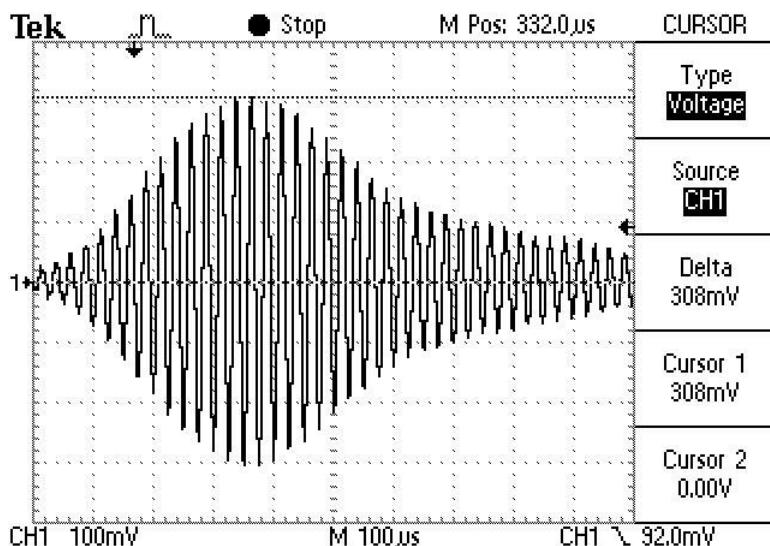
6.2.4.9 ábra – Az abszorpciós együttható a frekvencia függvényében

A fenti ábra alapján az abszorpciós együttható értéke 40 kHz-en (ultrahangos adónk rezgési frekvenciáján) 0.2 dB/m. Ez 5 méteres távolságban 1 dB csillapítást okoz, tehát a hangnyomás amplitúdója a kezdeti érték 90%-ára csökken.

Összességében elmondhatjuk, hogy csillapítás jelentős hányadát az $1/r^2$ -es sugárzási törvény okozza. Szemléletesen azt mondhatjuk, hogy ha egy hangforrástól 5 cm-re elhelyezkedő érzékelőt 5 méteres távolságra helyezünk át, az érzékelt jelszint a századrészére csökken. Ebből következik, hogy a vevőkapszulák jeleit az adó és a vevő egymástól mért távolságától függő mértékben kell erősíteni. A méréshez tehát várhatóan a mért jelnek adaptációs szabályozására is szükség lehet.

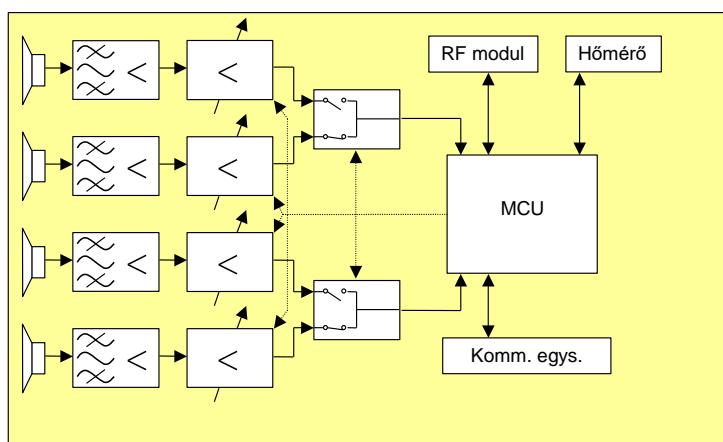
A vevőegység tulajdonságai

Minden méréshez az adóegység egy impulzuscsomagot (az alábbi példák esetében 10 teljes periódusnyi szinusz hullámot bocsát ki, ami az éppen aktuális vevőkapszula kimenetén a következő jelalak formájában jelenik meg:



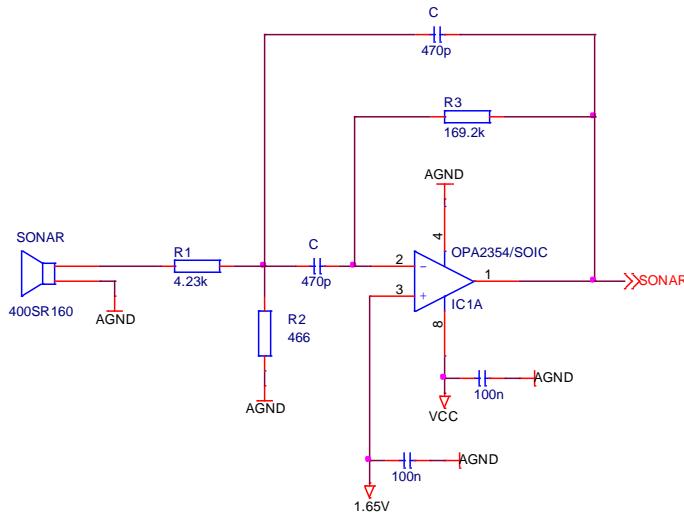
6.2.4.10 ábra – Jelalak a vevőkapszula kimenetén

Mint már említettük, a vevőegység az irányfüggés kiküszöbölésére 4 vevőegységet is tartalmaz, mindegyik vevő után sávszűrő található a hasznos frekvenciakomponens kiválasztására, majd erősítő és multiplexer fokozaton keresztül illesztjük a jeleket a feldolgozó mikrokontrollerhez.



6.2.4.11 ábra – A vevő egység blokkvázlata

A szonár vevőkapszula jelét aktív műveleti erősítős kapcsolás fogadja, amely egyrészt erősíti a vett jelet, másrészt megvalósítja a zajok kiszűrését biztosító másodfokú sávszűrő kapcsolást is.



6.2.4.12 ábra – A vevő fogadó egységének kapcsolása

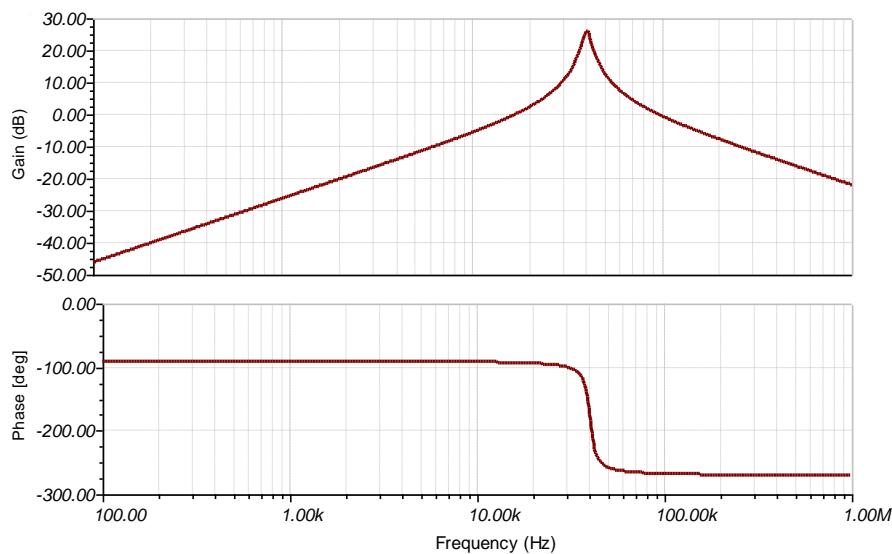
A kapcsolás egy másodfokú sávszűrőt valósít meg, melynek átviteli függvénye a következő egyenletekkel és karakterisztikákkal jellemezhető:

$$H(s) = \frac{T_0 \frac{s}{Q_p \omega_p}}{1 + \frac{s}{Q_p \omega_p} + \frac{s^2}{\omega_p^2}}, \quad \text{ahol}$$

$$\omega_p = \frac{1}{C \cdot \sqrt{(R_1 \times R_2) \cdot R_3}}$$

$$Q_p = \frac{1}{2} \cdot \sqrt{\frac{R_3}{R_1 \times R_2}}$$

$$T_0 = \frac{1}{2} \frac{R_3}{R_1}$$
(6.2.25)



6.2.4.13 ábra – A vevő fogadó egységének szűrőkarakterisztikája

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 44. oldal
--	---	--

A mikrokontroller a szűrőkapcsolás kimenetén megjelenő jelet mintavételezi, méghozzá a jel alakhű visszaállításához legalább 10-szeres túlmintavételezéssel (oversampling). Ily módon az elvárt analóg-digitális átalakítás sebessége $f_{A/D} = 4 \times 10 \times 40 \text{ kHz} = 1.6 \text{ MHz}$ (!) Kisebb mikrokontrollerek alkalmazása esetén a vett jelek valós időben történő feldolgozása nem elvárható.

A módszerrel viszonylag nagy pontossággal határozható meg robotunk helye. Ügyeljünk arra, hogy egyetlen vett hangperiódus „elvétése” (pl. magas alapzaj miatt a beérkező jelcsomag 1 periódussal később történő észlelése) kb.

$$\Delta s = 1/40 \text{ kHz} \times 340 \text{ m/s} = 8.5 \text{ mm} \quad (6.2.26)$$

pontatlanságot jelent az objektum helyzetének meghatározásában. A háromszögelés egy ismert hátránya, hogy a helymeghatározás pontossága nem azonos a mozgástér valamennyi részében, a háromszög „ellaposodása” (a mérendő objektum valamelyik vevőhöz közel a sarokban való elhelyezkedése) esetén a mérés hibája nagymértékben megnő.

Összefoglalva az eljárás fontosabb előnyei és hátrányai a következők:

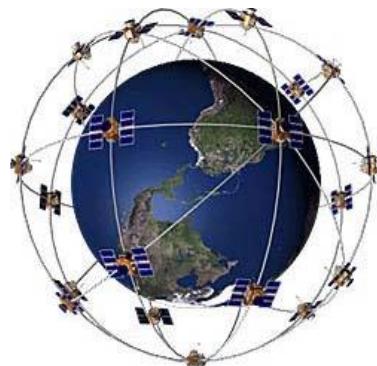
- **előnyök**
 - viszonylag pontos helymeghatározás (kb. 1-2 cm)
 - egyszerű elektronikus kapcsolás
 - korlátozott energiaigény
- **hátrányok**
 - korlátozott távolságok esetén használható (5-10 m)
 - a helymeghatározás pontossága romlik a vevők közelében
 - a módszer ebben az egyszerű kivitelében nem szolgáltat orientációs információt
 - azonos térben működő több egység esetén jelkódolást kell alkalmazni.

6.2.5 GPS és DGPS

A GPS rendszert az Amerikai Védelmi Minisztérium fejlesztette ki a 1970-es években annak érdekében, hogy járművek minden meg tudják határozni pontos pozíójukat, valamint a katonai vezetés is tudjon helyzetükről. A rendszer teljes neve NAVSTAR GPS (Navigation Satellite Tracking And Ranging Global Positioning System). Elődje az ún. NNSS (Navy Navigational Satellite System) vagy más néven TRANSIT rendszer, ami szintén egy műholdas navigációs rendszer volt, de elsősorban a hajózásban alkalmazták, mivel pontossága mindössze 100m körül volt. A régi rendszer működése a Doppler-effektuson alapuló mérésre épült, ami gyors hordozóeszközök (repülőgépek) számára nem tette lehetővé az alkalmazását. A szélesebb körben való alkalmazhatóság érdekében fejlesztették ki a GPS rendszert, a program 1978-ban indult és jelenleg is tart. Az amerikai hadseregnak e rendszer segítségével sikerült megnyernie a Perzsa öbölháborút 1991-ben, mivel így éjszaka tudtak vonulni a sivatagban anélkül, hogy vizuális tárgyak alapján kellett volna tájékozódniuk.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 45. oldal
--	---	--

Egyelőre az Egyesült Államok **NAVSTAR GPS** rendszere az egyetlen olyan rendszer, amely teljesen mértékben funkcionál napjainkban, ám ennek használatára is 16 évet kellett várni, mivel ekkorra épült ki a 24 műholdból álló teljes értékű rendszer. A GPS szolgáltatásai 1994 óta a civilszféra számára is elérhetők, korlátozott pontossággal, aminek feloldására még további éveket kellett várni, kb. 22 évvel az első műhold fellövését követően valósult meg az, hogy bárki korlátozás nélkül foghatja a helymeghatározáshoz szükséges jeleket.



6.2.5.1 ábra - A GPS rendszer műholdjai

A rendszert összesen 24 darab műhold alkotja, melyek 6 – egymáshoz képest 60°-kal elforgatott - pályasíkon keringenek a Föld felszínétől 20 180 km-es távolságban az Egyenlítőhöz képest 55°-os inklinációs szög alatt. A műholdak helyzetét úgy határozták meg, hogy egy adott földi pozícióból egyszerre több is látható. Az égbolton sík terepről egyszerre 7-12 műhold látható, melyből a helymeghatározáshoz 3, a tengerszint feletti magasság meghatározásához pedig további egy hold szükséges. A műholdak keringési ideje 11 óra 58 perc, és emiatt minden műhold kb. 5 órán keresztül ugyanazon a nyomvonalon látható (ezek a műholdak nem tévesztendők össze a geostacionárius műholdakkal, amelyeket a megfigyelő minden ugyanazon a helyen lát). Azt, hogy több műholdat is lássunk a feltétlenül szükségesnél azért előnyös, mert ekkor a mérés minden a legkedvezőbb pozícióban történő műholdakkal történhet.

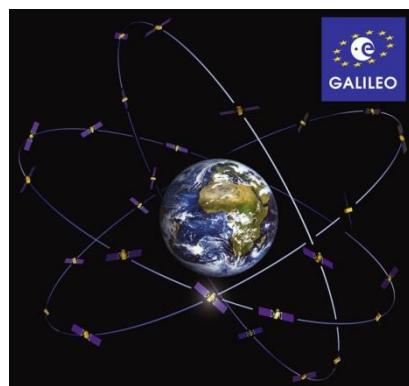


6.2.5.2 ábra - GPS műhold a szerelőüzemben és a világűrben (fantáziakép)

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 46. oldal
--	---	--

Az orosz **GLONASS** (a Globális Navigációs Műholdrendszer, azaz a GNSS orosz megfelelője) rendszer tervezése a 70-es évek második felében kezdődött el. Az első műholdakat 1982-ben bocsátották fel, a használatbavételhez szükséges 12 darabot 1991-ben sikerült elérni – ezek két pályán keringenek a Föld felszínétől 19 130 kilométeres magasságban. A rendszer 1995-ben érte el teljes funkcionálisitását három pályán, összesen 24 darab műholddal. Ezt követően – az orosz gazdasági hanyatlás következtében – a műholdak karbantartása elmaradt, és a rendszer állapota jelentősen romlott. 2001-ben a programot újból elővették, de újbóli rendszerbe állításáról ellentmondásos híreket hallani. A GLONASS esetében a három pálya inklinációja 64.8° , amely felülmúlja az amerikai GPS rendszer 55° -os értékét, tehát elvben jobb lefedettséget biztosít a sarkkörökön. A műholdak elrendezése következtében egy időben a Föld bármely pontján legalább öt GLONASS műhold látható (eggyel kevesebb, mint a GPS esetében).

Európa sem akar lemaradni Amerika mögött. 2002 márciusában az EU elfogadta egy sok (kb. 20) milliárd eurós navigációs rendszer kiépítését, ami az autósoktól a hajósokig bezáróan bárkinek segítséget nyújt a tájékozódásban a világ bármely pontján. A rendszert az olasz csillagászról, Galileo Galileiről **Galileo**-nak nevezték el, és nem titkolt célja az amerikai GPS uralmát megtörni - bár a két rendszer kompatibilis lesz egymással. A Galileo rendszer 27 működő és 3 tartalék műholdból fog állni, amelyek a Föld körül kb. 23222 kilométeres magasságban keringenek majd. A rendszer fejlesztési és tesztelési szakasza 2011-ig tartott, 2014-ben kezdte meg a működését, teljes kiépítését 2020-ig tervezik. Mivel a műholdak a legmodernebb technikával fognak készülni, a pontosságuk mintegy tízszer jobb lesz, mint a GPS rendszeré.

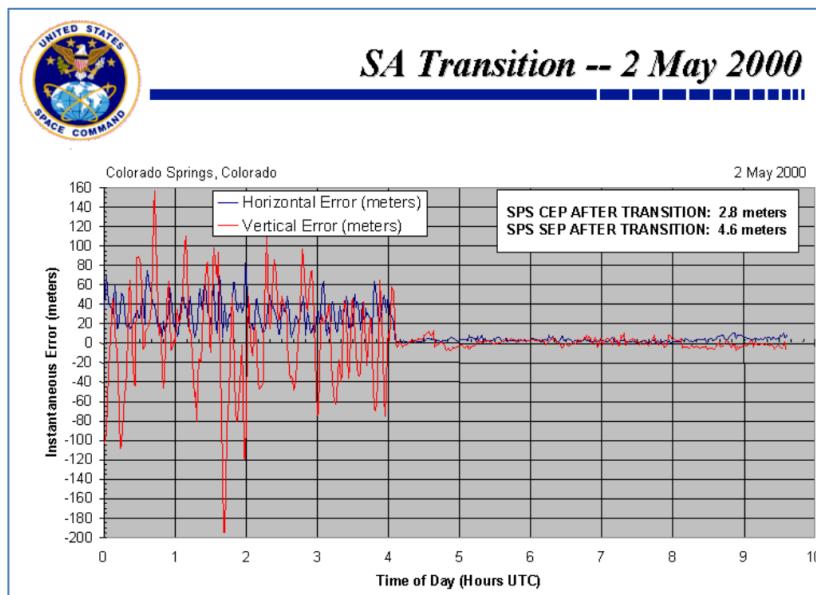


6.2.5.3 ábra - A Galileo rendszer műholdjai

Az amerikai NAVSTAR GPS és orosz GLONASS rendszerek katonai felügyelet alatt állnak, ami több okból kifolyólag is hátrányt jelentett (és jelent) a helymeghatározó rendszereket használók számára. Az orosz GLONASS kezdettől fogva megengedi a civil felhasználók számára is az Y kód használatát, de a vevők nem terjedtek el, elsősorban magas áruk és korlátozott előállításuk miatt. Ezzel szemben a GPS esetében bármikor visszavonhatják a hozzáférést a civil szféra számára. A Galileo rendszer ezekkel ellentétben minden fegyveres testülettől független lesz. Ez, és még az a tény, hogy Kína is segítséget nyújt a rendszer kiépítésében, szálka lett az Egyesült Államok szemében. Ők ugyanis a GPS polgári hozzáférhetőségét egy vélt, vagy valós konfliktus esetén csökkenthetik, vagy akár meg is

szüntethetik, míg a Galileo tőlük függetlenül tovább üzemelne - így azt az ellenfél is felhasználhatná navigációs célokra. Az amerikai monopóliumtól való félelem volt az egyik fő oka a Galileo projekt megszületésének: a Pentagon ugyanis nem vállalt semmiféle garanciát a műhold-szolgáltatás biztosítására. A GPS rendszer egy esetleges leállítása viszont katasztrófákhoz vezethet, gondolunk akár egy tankhajó zátonyra futására.

A NAVSTAR GPS rendszer tartalmaz egy ún. „szelektív hozzáférhetőség” (selective availability = SA) kapcsolót. Az SA lényegében a műhold órajel-frekvenciájának mesterséges elrontását és a pályaadatok szándékos torzítását jelenti (a technikai részleteket ld. később). Ennek célja az, hogy a rendszer pontatlanságát mesterségesen megnöveljék kb. ± 100 m-re, amivel gyakorlatilag meg lehet akadályozni, hogy a rendszert civil célokra használni lehessen. Az amerikai elnöknek évente kell döntenи abban a kérdésben, hogy a SA továbbra is kikapcsolva maradjon-e. 2000. május 2-ig a GLONASS volt az egyedüli pontos helymeghatározó rendszer. A civil felhasználók számára elérhető pontosság akkor 20 méter körüli értéket ért el, szemben a GPS 100 méteres pontosságával. 2000. május 2. hajnali 4 órától kezdődően azonban az amerikai kormány határozata értelmében kikapcsolták az addig használt SA zavarást, amely jelentősen megnövelte a GPS pontosságát: az a korábbi kb. 100 méterről 10-20 méter körüli értékre csökkent.



6.2.5.4 ábra – Az SA kikapcsolása a GPS rendszerben (www.gps.gov)

A Galileo rendszer az EGNOS szolgáltatás (ld. később) integrálásával sokkal nagyobb pontosságot érhet majd el. A Galileo ingyenes szolgáltatás keretében elérhető horizontális pontossága 4 méter, vertikális pontossága 7 méter lesz. Ezeket a követelményeket a Föld területének legalább 95%-án képes lesz teljesíteni. Természetesen lehetőség lesz pontosabb szolgáltatások igénybevételére is költségtérítés ellenében. Ezek a szolgáltatások elérhetik majd a centiméteren belüli pontosságot, ám ennek eléréséhez természetesen szükség van kiépített aktív GPS hálózatra is (korrekciós számítások elvégzéséhez).

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszer technikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 48. oldal
--	--	--



6.2.5.5 ábra – A GIOVE-A tesztműhold
2006. január 12. óta továbbít jeleket a Galileo rendszerben



6.2.5.6 ábra – A Galileo program kísérleti fázisának első két műholdja
(IOV=In-Orbit Validation)

A Galileo program első tesztműholdja, a GIOVE-A 2005. december 28-án indult Bajkonurból. Segítségével az ESA (European Space Agency) szakemberei sikeresen tesztelték a fedélzeti műszereket és a földi vevőberendezéseket. A második műhold, a GIOVE-B 2008. április 27-én indult. Fedélzetén volt a leg pontosabb atomóra (hidrogén mézer), amely addig is a világűrbe került. A program kipróbálási (IOV = In-Orbit Validation) fázisa a 2011. október 21-én indított első két műhold fellövésével kezdődött, majd két hasonló műhold indítására került sor, 2012. október 12-én. A négy IOV hold szerepe az, hogy az újfajta navigációs űreszközök fedélzeti berendezéseit, a jelsugárzást, a földi követésre és irányításra kiépített hálózatot és központokat tesztelje. Ezek a műholdak már a végleges rendszer részei lesznek. (Pontosabban közülük 3: az IOV-4 műhold tápellátása meghibásodott, aminek következtében az E1 navigációs jel sugárzása megszűnt.)

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 49. oldal
--	---	--

2013. május 24-én érkezett meg az európai műholdas navigációs rendszer véglegesnek szánt műholdsorozatának első tagja az ESA hollandiai technológiai központjába. Noordwijk mellett, az ESTEC (European Space Research and Technology Centre) létesítményeiben végezték el azokat a vizsgálatokat, amelyek sikeres végrehajtása nélkül nem indulhatott az űrbe a Galileo műholdrendszer teljes kiépítési fázisának (Full Operational Capability, FOC) első üreszköze. Ez is bekövetkezett: 2014. augusztus 22-én elindult az első két végleges (FOC-1, FOC-2) műhold. Sajnos minden műhold félresikerült pályára állt (hibás tervezés miatt megfagyott az üzemanyag egy része, és a hordozórakéta nem tudta elvégezni a szükséges pályamanővereket). 2014 decemberében (FOC-1) és 2015 márciusában (FOC-1) használható pályára sikerült navigálni őket, teljes használhatóságuk még mindig kérdéses. A program azóta is folytatódik, jelenleg 18 műhold kering a rendszer részeként földkörüli pályán.

Nr.	Műhold	Név	Gyereknév	Fellövési dátum	Hordozórakéta	Küldetési kód	Státusz
1	Galileo-IOV PFM	GSAT0101	Thijs	2011-10-21	Sojuz-2-1b Fregat-MT	VS01	Működő
2	Galileo-IOV FM2	GSAT0102	Natalia	2011-10-21	Sojuz-2-1b Fregat-MT	VS01	Működő
3	Galileo-IOV FM3	GSAT0103	Dawid	2012-10-12	Sojuz-2-1b Fregat-MT	VS03	Működő
4	Galileo-IOV FM4	GSAT0104	Sif	2012-10-12	Sojuz-2-1b Fregat-MT	VS03	Meghibásodott
5	Galileo-FOC FM1	GSAT0201	Doresa	2014-08-22	Sojuz-2-1b Fregat-MT	VS09	Bejáratás alatt
6	Galileo-FOC FM2	GSAT0202	Milena	2014-08-22	Sojuz-2-1b Fregat-MT	VS09	Bejáratás alatt
7	Galileo-FOC FM3	GSAT0203	Adam	2015-03-27	Sojuz-2-1b Fregat-MT	VS11	Működő
8	Galileo-FOC FM4	GSAT0204	Anastasia	2015-03-27	Sojuz-2-1b Fregat-MT	VS11	Működő
9	Galileo-FOC FM5	GSAT0205	Alba	2015-09-11	Sojuz-2-1b Fregat-MT	VS12	Működő
10	Galileo-FOC FM6	GSAT0206	Oriana	2015-09-11	Sojuz-2-1b Fregat-MT	VS12	Működő
11	Galileo-FOC FM8	GSAT0208	Andriana	2015-12-17	Sojuz-2-1b Fregat-MT	VS13	Működő
12	Galileo-FOC FM9	GSAT0209	Liene	2015-12-17	Sojuz-2-1b Fregat-MT	VS13	Működő
13	Galileo-FOC FM10	GSAT0210	Danielé	2016-05-24	Sojuz-2-1b Fregat-MT	VS15	Bejáratás alatt
14	Galileo-FOC FM11	GSAT0211	Alizée	2016-05-24	Sojuz-2-1b Fregat-MT	VS15	Bejáratás alatt
15	Galileo-FOC FM7	GSAT0207	Antonianna	2016-11-17	Ariane 5 ES	VA233	Bejáratás alatt
16	Galileo-FOC FM12	GSAT0212	Lisa ^[8]	2016-11-17	Ariane 5 ES	VA233	Bejáratás alatt
17	Galileo-FOC FM13	GSAT0213	Kimberley	2016-11-17	Ariane 5 ES	VA233	Bejáratás alatt
18	Galileo-FOC FM14	GSAT0214	Tijmen	2016-11-17	Ariane 5 ES	VA233	Bejáratás alatt

6.2.5.7 táblázat – Az eddig pályára bocsátott Galileo műholdak

Folyamatosan zajlik tehát a majd végül 30 egyszerre keringő (27 működő és 3 tartalék) holdból álló rendszer kiépítése. A kb. 23,5 ezer km-es felszín feletti magasságú, 56°-os hajlásszögű körpályákon keringő műholdak, három különböző pályasíkba lesznek rendezve. A majdani teljes konstelláció a Föld legnagyobb részén, minden időben elegendő műholdat biztosít a horizont felett ahhoz, hogy a rádiójeleket venni és feldolgozni képes berendezések ezek alapján meghatározzák a vevő pontos helyét és sebességét.

(A programhoz kapcsolódóan az Európai Bizottság gyermeknek rajzverseny-pályázatot írt ki, és ezek nyerteseiről – minden tagországból egy-egy gyerekről – nevezik el a műholdakat. "Magyarország műholdjának" neve a nyertes magyar-holland származású budapesti kislány neve után Lisa lesz.)

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszer technika előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 50. oldal
--	--	--

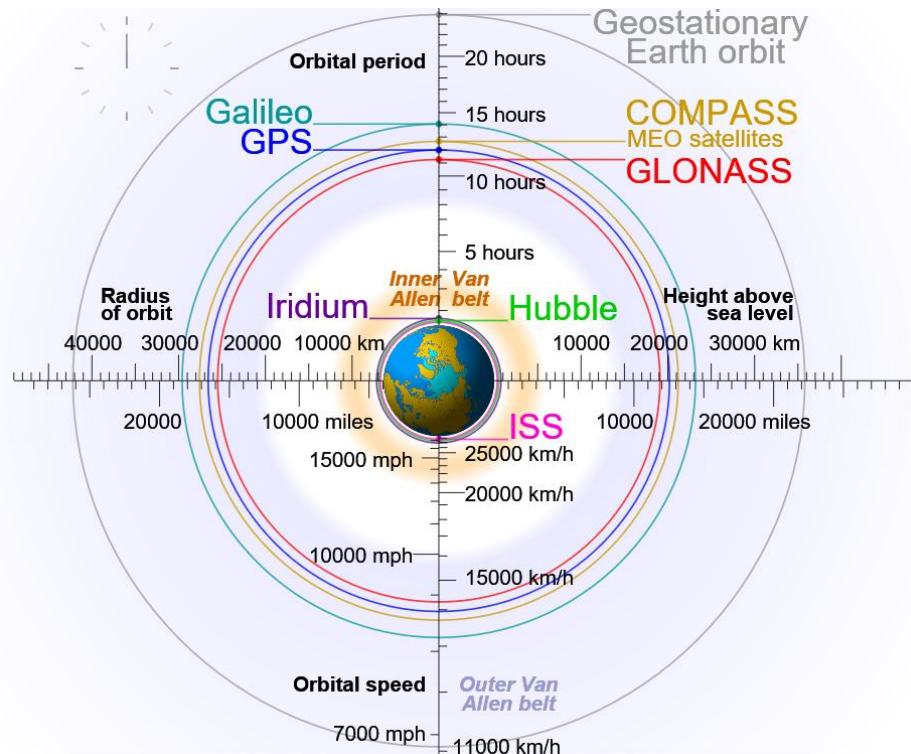


*6.2.5.8 ábra – Az első két Galileo műhold indítása 2011. október 21-én
Francia Guyana, Kourou űrközpontról*



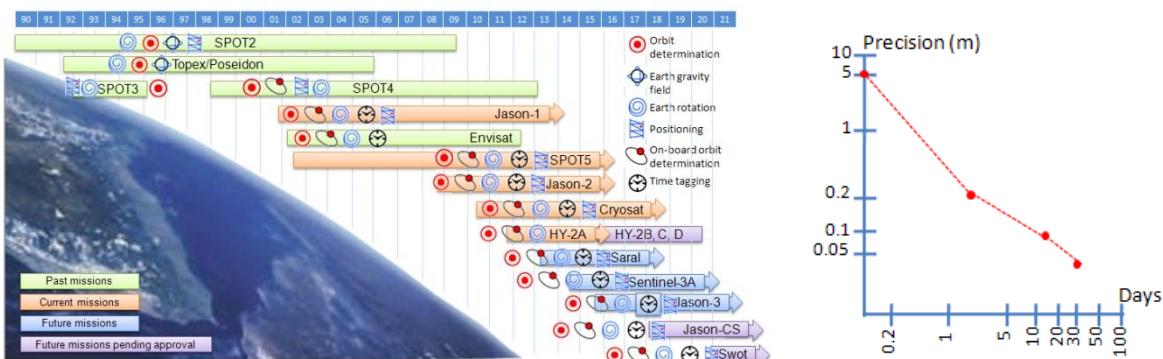
6.2.5.9 ábra – A FOC-5 és a FOC-6 műholdak a hordozórakétában

Az eddig felsorolt műholdas helymeghatározó rendszerek műholdjainak elhelyezkedését jól szemlélteti a 6.2.5.10 ábra. A skálákon leolvashatók az orbitális pályák Föld középpontjától és a tengerszinttől mért magasságai, a keringési idők és a műholdak sebességei is. A belső Van Allen sugárzási övön belül helyezkedik el a nemzetközi űrállomás (ISS) pályája, ezt követik a Hubble űrtávcső, majd az Iridium műholdak (telefonszolgáltatás). Ekkor még csak pár száz (360-780) kilométer magasan vagyunk a Föld felett, a navigációs rendszerek műholdjai ennél lényegesen távolabb, a közepes magasságú földkörüli pályákon (MEO), azaz kb. 2000 km-től a geoszinkron pálya magasságáig (35786 km) keringenek. Keringési magasság szerint növekvő sorban: Glonass [19100 km], Navstar GPS [20200 km], Compass [21000 km] – ld. később, Galileo [23200 km].



6.2.5.10 ábra – Műholdrendszer a Föld körül (by Geo Swan)

A műholdas pozíció-meghatározó rendszerek fenti felsorolása nem teljes. Megemlítjük, hogy Kína BeiDou (BDS, angol névén Compass) néven hozott létre navigációs rendszert, amely 2000-ben kezdte meg működését Kína és a vele szomszédos országok számára. 2010-ben jelentette be India saját navigációs rendszerének fejlesztését (IRNSS), melynek működtetését 2015-től tervezik. Ez a rendszer is csak a gazda ország kb. 1500 km-es körzetében fog szolgáltatást nyújtani. Quasi-Zenith Satellite System (QZSS) néven Japán készített saját magának rendszert, amely a GPS rendszer kiegészítése pontosabb pozícióadatok nyerése érdekében. Franciaország Doris néven hozott létre egy olyan Doppler-effektuson alapuló rendszert, amellyel (hosszúidejű átlagolással és számításokkal) geodéziai mérésekhez néhány cm pontosságú pozíció-meghatározás érhető el.

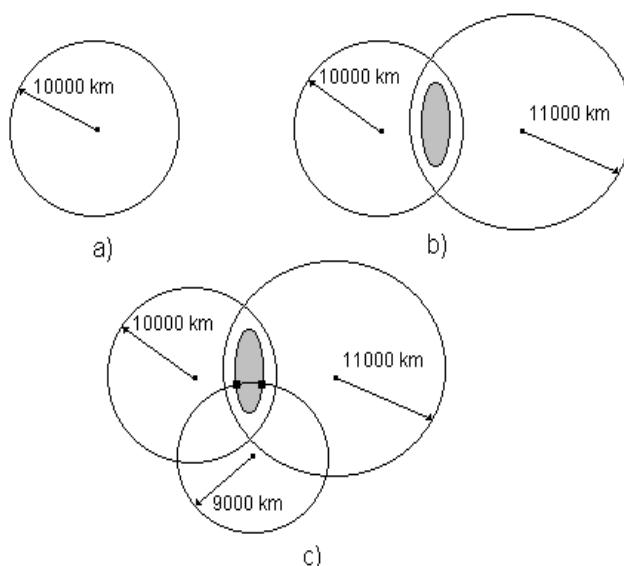


6.2.5.11 ábra – A francia Doris rendszer pontossága és kiépítettsége

Hogyan működik ?

A GPS rendszer működése a műholdak távolságának megmérésén alapul. Ez azt jelenti, hogy helyzetünket a Földön annak segítségével határozzuk meg, hogy lemjük a távolságunkat az űrben lévő egyes műholdaktól, vagyis a műholdak jelentik számunkra a viszonyítási pontokat. Felvetődhet a kérdés, hogy egyszerűen hogyan tudjuk pontosan megmérni egy műholdtól mért távolságunkat, másrészt hogyan tudjuk pontosan meghatározni, hol tartózkodik egy mozgásban lévő műhold ? Első körben fogadjuk el, hogy vannak olyan módszerek, melyek segítségével pontosan ki lehet számítani, hol van egy műhold az űrben, valamint azt is, hogy milyen távolságra vagyunk ettől a műholdtól.

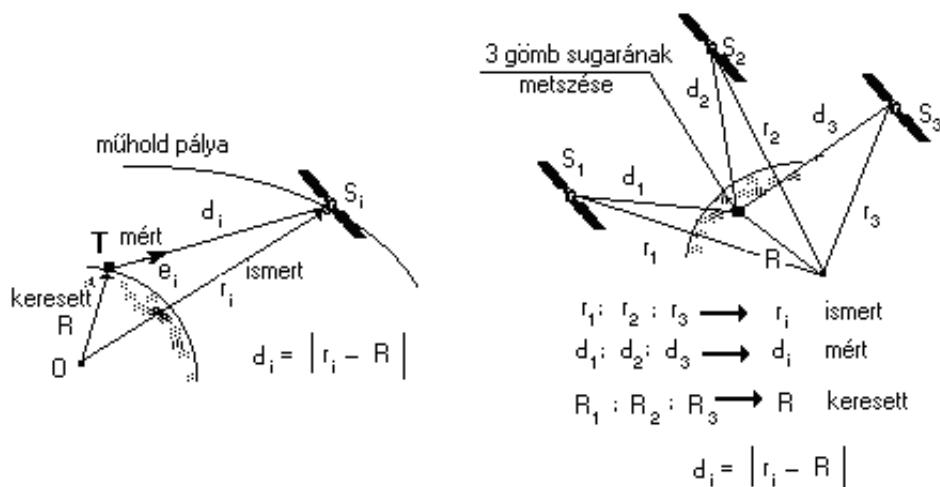
A GPS rendszer alapelve rendkívül egyszerű, csak némi térszemléletet igényel: ennek bemutatását a 6.2.5.12 ábra alapján végezzük. Ha tudjuk például, hogy az „A” jelű műholdtól 10 000 km távolságra vagyunk, akkor lényegesen leszük az a tér, ahol az egész világegyetemen belül tartózkodhatunk. Ez azt jelenti ugyanis, hogy valahol egy olyan gömb felületén lehetünk csak, amelynek középpontja az "A" műhold, és sugara 10 000 km.



6.2.5.12 ábra – Helymeghatározás 3 műholdtól mért távolság alapján

Amennyiben egyidejűleg még azt is tudjuk, hogy egy "B" jelzésű műholdtól 11 000 km távolságra vagyunk, ez tovább szűkíti azt a helyet, ahol tartózkodhatunk. Az a hely ugyanis, ahol 10 000 km távolságra vagyunk az "A" és 11 000 km távolságra a "B" műholdtól, azon a körvonalon fekszik, ahol a két gömb metszi egymást. Ha ezután egy harmadik műholdtól is elvégezzük a távolságmérést, akkor már lényegesen meghatározhatjuk a pontos helyünket. Amennyiben ugyanis tudjuk azt, hogy ugyanebben az időben a "C" műholdtól pl. 9 000 km a távolságunk, akkor már csak két pont lehet, ahol tartózkodhatunk. Ez a két pont ott található, ahol a 9 000 km sugarú gömb metszi a körvonalaat, ami a 10 000 km sugarú és a 11 000 km sugarú gömb metszésvonalával volt. A három műholdtól végrehajtott távolságmérésessel tehát leszűkítjük a térben lehetséges tartózkodási helyünket két pontra. Később látni fogjuk, hogy egy további műszaki ok miatt szükségünk lesz még egy távolságmérés végrehajtására, most azonban elméletileg elegendő számunkra ez a három távolság.

Hogyan tudjuk eldönteni, hogy a két pont közül melyik az igazi? A válasz rendkívül egyszerű: általában a két pont közül az egyik eleve fizikailag lehetetlen pozíciót eredményezne. Vagy túlságosan messze van a földtől, vagy lehetetlenül nagyra adódna a távolságunk előző helyzetünkhez képest. A GPS vevőkben levő számítógépek egyszerű hihetőség-vizsgálattal meg tudják különböztetni a helyes megoldást a fizikailag lehetetlenről. Amennyiben biztosan ismerjük a helyzetünk földfelszín feletti magasságát (pl. a tengerészek mindenig tudják, hogy a helyzetük magasságkoordinátája maga a tengerszint), úgy kiküszöbölnéjük a három közül az egyik műholdtól való távolságmérést. Ebben az esetben ugyanis a rajzokon ábrázolt gömbök egyikét helyettesíthetjük egy olyan gömbbel, melynek középpontja a földgömb közepe, s melynek sugara megegyezik a földgömb középpontjától vett távolságunkkal. A GPS rendszeren alapuló helymeghatározás alapelve tehát az, hogy műholdakat alkalmazunk referencia-pontokként, és ezeket saját helyzetünknek valahol a földfelszínen háromszögeléssel történő meghatározására használjuk.



6.2.5.13 ábra – Távolságmérés 1, majd 3 műhold esetén

Jelöljük a 6.2.5.13 ábra alapján a földfelszín tetszőleges T pontjába mutató helyzetvektort \bar{R} -rel, a tetszőleges pályán a Föld körül keringő mesterséges hold pillanatnyi tartózkodási helyét jelentő S_i pontba mutató helyzetvektort pedig \bar{r}_i -vel. Jelöljük továbbá a földi pontból a mesterséges hold pillanatnyi helyére mutató vektort $\bar{e}_i \cdot d_i$ -vel, ahol \bar{e}_i egységvektor, d_i pedig a T és S_i pontok távolsága. Felírható a következő egyszerű összefüggés:

$$\bar{r}_i = \bar{d}_i + \bar{R} \quad (6.2.27)$$

A mesterséges holdak navigációs célú felhasználásakor általában d_i -t mérjük. Ne feledjük azonban, hogy mindegyik vektor három összetevőt tartalmaz. A módszerrel a földi pont és a mesterséges hold pillanatnyi helyzete közötti vektornak csak a nagysága (a vektor hossza) határozható meg, a vektor iránya nem:

$$d_i = |\bar{r}_i - \bar{R}| \quad (6.2.28)$$

ahol $\bar{R}(x, y, z)$ a vevő ismeretlen, $\bar{r}_i(x_i, y_i, z_i)$ pedig a műhold ismert helyvektorai.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 54. oldal
--	---	--

Mivel a gyakorlatban a vevőkben egy kevésbé „pontos” óra található, amelynek egy δ -val jelölt órahibája van a mesterséges holdak órához képest (time offset), a futási idő mérésének ezen δ hibája következtében a mért távolság is egy $\Delta d = c \cdot \delta$ értékkal eltér a tényleges értéktől. A vevő órahibája miatt mért közelítő távolságot a valóságos távolsággal szemben pszeudotávolságnak nevezzük (pseudorange, jelölése D_i):

$$D_i = d_i + \Delta d \quad (6.2.29)$$

Ezekkel most már felírhatjuk a következő egyenletrendszert:

$$\begin{aligned} D_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} + c \cdot \delta \\ D_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} + c \cdot \delta \\ D_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} + c \cdot \delta \\ D_4 &= \sqrt{(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2} + c \cdot \delta \end{aligned} \quad (6.2.30)$$

Itt az (x_i, y_i, z_i) koordináták az $i = 1, 2, 3, 4$ mesterséges holdak ismert térbeli derékszögű koordinátái a kérdéses időpontban. A négy egyenlet négy ismeretlenet tartalmaz: a vevőantenna (x, y, z) koordinátáit és a δ órahibát. Ez egy nemlineáris egyenletrendszer, melyet először linearizálni kell, majd ezután iterációval oldható meg.

Mivel a GPS rendszer a műholdaktól való távolságunk ismeretén alapul, szükségünk van tehát egy olyan mérési módszerre, amellyel nagy pontossággal meghatározható, hogy milyen messze vagyunk a műholdaktól. A távolságmérés alapelve itt is a hullámok terjedési idejének mérése – szemben a korábban megismert ultrahangos távolságméréssel azonban itt jóval nagyobb sebességű rádióhullámok terjedési idejét kell megmérnünk. A GPS vevő úgy működik, hogy megméri azt az időt, ami alatt elér bennünket a műholdról kibocsátott rádiójel, majd ezt követően kiszámítja ebből az időből a távolságunkat az adott műholdtól. A rádióhullámok a fény sebességével terjednek ($c \approx 3 \cdot 10^8$ m/s). Ha pontosan meg tudjuk határozni, hogy mikor indította el felénk a műhold a rádióüzenetét, és mi mikor vettük azt, akkor már tudjuk, hogy mennyi idő alatt ért el bennünket ez az üzenet. (Ne feledjük: 4 különböző műholdtól 4 távolságmérés szükséges, hogy ki tudjuk számítani a helykoordinátáinkat). Természetesen az időmérés pontossága szabja meg a távolságmérés pontosságát.

A elektromágneses hullámok terjedési idejének meghatározásához először is pontosan meg kell határozni azt a pillanatot, amikor a jel elhagyta a műholdat. Ez a következő módon lehetséges: szinkronizálni kell a műholdat és a vevőket úgy, hogy ugyanazt a kódjelet pontosan ugyanabban az időben állítsák elő. Ez után már csak venni kell az egyes műholdaktól érkező kódjeleket és meg kell vizsgálnunk, hogy a saját vevőnk mennyi idővel korábban állította elő ugyanazt a kódjelet. Az időkülönbség az idő, amely alatt a jel eljutott hozzánk a műholdtól.

A helymeghatározás legkritikusabb eleme tehát az idő, a műholdról kibocsátott rádiójelek terjedési idejének a lehető legfontosabb meghatározása. Minél pontosabb az óránk, értelemszerűen annál pontosabb eredmény érhető el. Ennek érdekében a rendszerben atomórát

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 55. oldal
--	---	--

használnak, amely elvileg több százezer év alatt is csak egyetlen másodperc körüli késést produkál - persze az már más kérdés, hogy egy atomóra élettartama anyagválasztástól függően körülbelül 5-10 év. Mi is az atomóra ? Fizikában tanultuk, hogy a másodperc definíciója atomfizikai tulajdonságon nyugszik:

„A másodperc az alapállapotú cézium-133 atom két hiperfinom energiaszintje közötti átmenetnek megfelelő sugárzás 9 192 631 770 periódusának időtartama”.

A cézium mellett számos más anyagot használnak az atomórákban, például hidrogént vagy rubídiumot. A Galileo műholdakat időmérés szempontjából pl. többszörösen túlbiztosított, ráadásul rendkívül pontos, hidrogénmézerrel működő órákkal szerelik fel (ez az óra hárommillió év alatt sietne vagy késne egy másodpercret, azaz egy nap kevesebb, mint egy nanoszekundum eltérés lehetséges), hogy a rendszer akár méteres mérési pontosságot tegyen majd lehetővé. Elméletileg léteznek a Galileoénál pontosabb atomórák is, de ezek jóval magasabb árkategóriát is képviselnek.



6.2.5.14 ábra – A Galileo műholdak passzív hidrogénmézerrel működő atomórája
(maser = microwave amplification by stimulated emission of radiation)

Minden Galileo műhold fedélzetén négy atomóra működik (rubídium és hidrogénmézeres), mivel az időmérés pontossága alapvető fontosságú a navigációs rendszerek működésében. Némi riadalomra adott okot, hogy 2017. januárjában 10 atomóra (3 rubídium, 7 hidrogén) megállt. Természetesen ez nem jelenti a méregdrága rendszerek végleges kiesését – szakértők nyomozták a hiba okát, amit megtaláltak, az órákat 2017. júliusban újraindították, és azok azóta hibátlanul működnek.

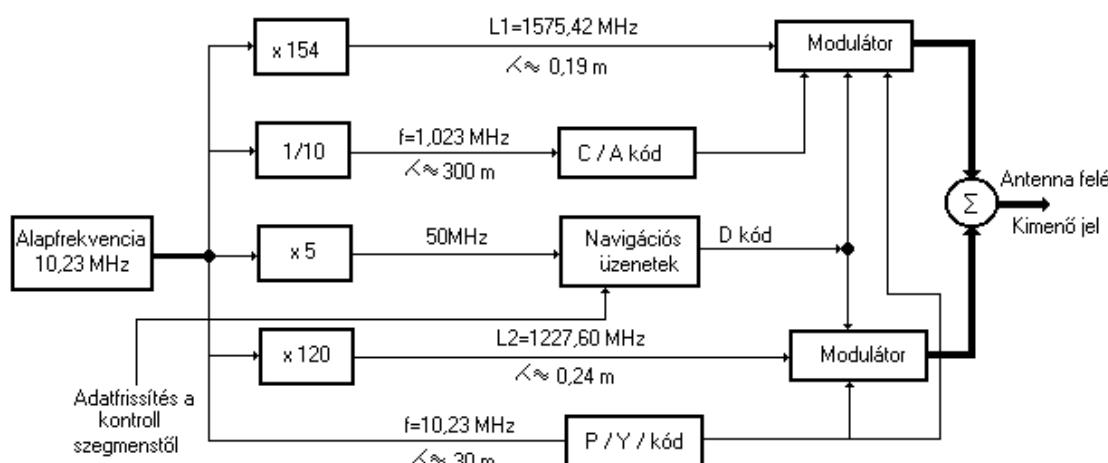
A technikai részletek...

Térjünk vissza a régóta működő Navstar GPS rendszerhez. A műholdak folyamatosan küldik hely és időadataikat a földfelszínre, amik alapján aztán a kliens oldali hardver háromszögelés segítségével meghatározza a koordinátákat, illetve a tengerszint feletti magasságot. A sugárzás két frekvencián történik:

$$L_1 = 154 \cdot f_0 = 1575.42 \text{ MHz} (\lambda = 19.05 \text{ cm.})$$

$$L_2 = 120 \cdot f_0 = 1227.60 \text{ MHz} (\lambda = 24.45 \text{ cm.})$$

ahol $f_0 = 10,23 \text{ MHz}$ a cézium, vagy rubídium atomóra alaposzcillátorának frekvenciája.



6.2.5.15 ábra - GPS műhold működési blokkvázlat

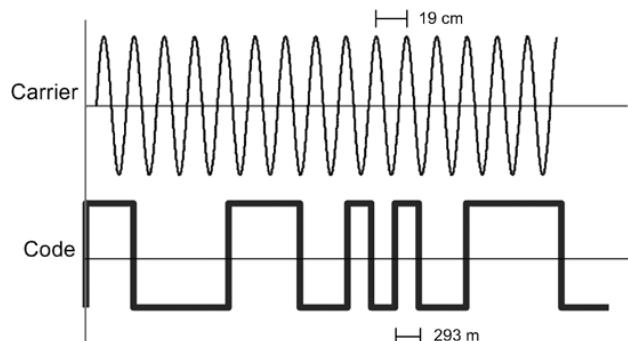
Az L1 és az L2 jelek mindegyike kódmodulált jel. A moduláció egyik célja a vivőhullámon egy időpont „megjelölése”, mivel majd a műhold és a vevő jelének összehasonlítása lesz az időmérésünk alapja. A moduláció másik célja a műhold speciális üzeneteinek a továbbítása. A két vivőhullám modulálásával kapjuk a modulált kódjeleket. A kódok a következők:

- a P (Precision = pontos) kódjel az L1 és L2 frekvencián,
- a C/A (Coarse/Acquisition = durva) kódjel az L1 frekvencián
- egy harmadik kód, a D- (data = adat) kód minden vivőfrekvencián navigációs üzeneteket, az ún. GPS adatokat tartalmazza.

A műholdak ezeket a kódjeleket továbbítják a vevőállomások felé. A C/A kód (vagy más néven civil kód) – amely bár meghatározott szabályok szerint készül mégis úgy néz ki mintha zaj lenne – pontos felépítése ismert, és mindenki számára hozzáférhető, ez a polgári felhasználók igényeit van hivatva kielégíteni. A szolgáltatását SPS-nek (Standard Positioning Service) nevezik. Az SA (=Selective Availability) zavarás 2000 májusában bekövetkezett kikapcsolása azonban nem jelentette a GPS minden felhasználható tulajdonságának felszabadítását a civil felhasználások számára. 1993-tól Amerika a P kódot titkosította (AS = AntiSpoofing). Az Y-nal jelölt titkos kódot kizárolag katonai vevőkészülékekbe építik be, csak ezek a készülékek képesek dekódolni azt. Ezek felhasználását Amerika mind a mai napig korlátozza. A P kód szolgáltatását PPS-nek (Precise Positioning Service) nevezik.

A vett jelek feldolgozása

A vevőegységek a vett jeleket alapvetően kétféle módszer szerint dolgozzák fel. Az egyik módszer szerint a műholdról érkező, C/A kóddal modulált jelet a vevő összehasonlítja a saját maga által előállított, hasonló kódú jelével (ez a kód ismert!), és addig csúsztatja a két jelet egymáson, míg azok fedésbe nem kerülnek egymással. Ezt a módszert **kódfázis-mérésnek** nevezzük, ezt alkalmazza jelenleg a hétköznapi felhasználásban lévő eszközök többsége. A kódfázis-mérés pontossága kisebb, több 10 m-es nagyságrendű (ld. később). A másik módszer szerint a vivőhullám fázisát hasonlítjuk össze a vevőben előállított referenciajel fázisával, ezt **vivőfázis-mérésnek** nevezzük. A vivőjel frekvenciája $L_1 = 1575,42$ MHz (hullámhossza 19,05 cm), azaz több, mint 1500-szorosa a C/A kód továbbítási frekvenciájának. Ezzel a módszerrel (ideális esetben) akár mm-es pontossággal végezhető el a jelek kiértékelése – lényegesen nagyobb pontosságú mérések nem kevés számítást igénylő kiértékelésével. A valós pontosságot elsősorban az időszinkronizáció és a rádióhullámok terjedési sebességének az ideálistól való eltérése befolyásolja.



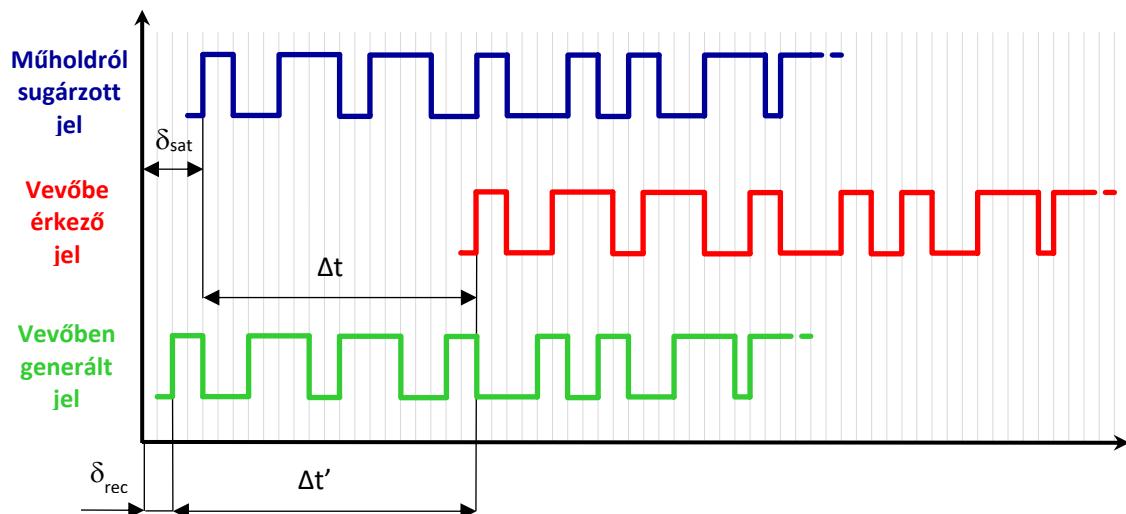
6.2.5.16 ábra – A kódfázis-mérés és a vivőfázis-mérés összehasonlítása

Kódfázis-méréses módszer esetén a vevőegység által vett kódok (demodulációt követően) egy bináris jelsorozatot alkotnak, azaz a 0 és az 1 értékek meghatározott sorozatának váltakozásából tevődnek össze. A NAVSTAR GPS rendszerben minden egyik műholdnak saját kódja van, vagyis minden egyik műhold ugyanazt a vivőfrekvenciát a saját speciális kódjeleivel modulálja, majd sugározza a vevőberendezések felé. (Ezzel szemben a GLONASS GPS rendszerben a vivőfrekvencia holdenként változik, a kód viszont azonos).

A távolságméréshez szükséges pontos időmérés kód-összehasonlításos technikával történik. Ennek során a vevő előállítja az ismert kód (a C/A kód) mását, majd összehasonlítja azt a műholdról vett bitsorozattal. Az összehasonlítás időkésletetéssel történik. Az összehasonlítás alapját a műholdról vett jel és a vevőben előállított referenciajel bitjeinek szorzata képezi. Egy bit hossza $\tau = 1/1.023\text{MHz} = 977.52$ nanoszakundum, amihez vákuumban mért fénysebesség mellett $\lambda = 293.052$ m-es hullámhossz tartozik – nem nehéz belátni, hogy ez a kvantum (± 1 bitidő, ún. chip-rate) egyben a távolságmérésünk kvantuma, azaz pontossága is. Ha a két kód egyezik („fedi egymást”), a szoratósszeg a lehető legnagyobb, azaz megegyezik a kód hosszával, 1023-mal.

A C/A kód hossza 1023 bit és valamennyi polgári célú vevőberendezés számára hozzáférhető, ismert (ún. pszeudovéletlen kód). Ezt a kódot nem csak a műholdak generálják, hanem a vevők is minden milliszekundum kezdetén. A műhold ezt a kódolt jelsorozatot sugározza ki, amelyet a vevő azonosítani tud, és meg tudja mérni az időkülönbséget a vett és a saját kód megfelelő pontjainak megjelenése között. Ha az adó órája és a vevő órája pontosan ugyanúgy járna, akkor a vett jel és a vevő jele közti időeltolódás (Δt) kizárolag a műholdról kibocsátott jel terjedési idejétől függne (ld. 6.2.5.17 ábra). A valóságban mért $\Delta t'$ terjedési idő a GPS időskálán mért valódi terjedési idő és a kölcsönös óraállás (δ) összege.

$$\Delta t' = \Delta t + \delta = \Delta t + (\delta_{sat} - \delta_{rec}) \quad (6.2.31)$$



6.2.5.17 ábra – Kódfázis-mérés a C/A vagy a P kódok vételekor

Megjegyezzük, hogy a műhold a saját órahibát modellező paramétereket a GPS-adatok keretében sugározza (a D-kódban, ld. később), így a fedélzeti óra hibája nem számottevő. A mért $\Delta t'$ terjedési idő és a c fénysebesség szorzata a D_i pszeudótávolság:

$$D_i = c \cdot \Delta t' = d_i + c \cdot \delta = d_i + c(\delta_{sat} - \delta_{rec}) \quad (6.2.32)$$

$$D_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + c(\delta_{sat} - \delta_{rec}) \quad (6.2.33)$$

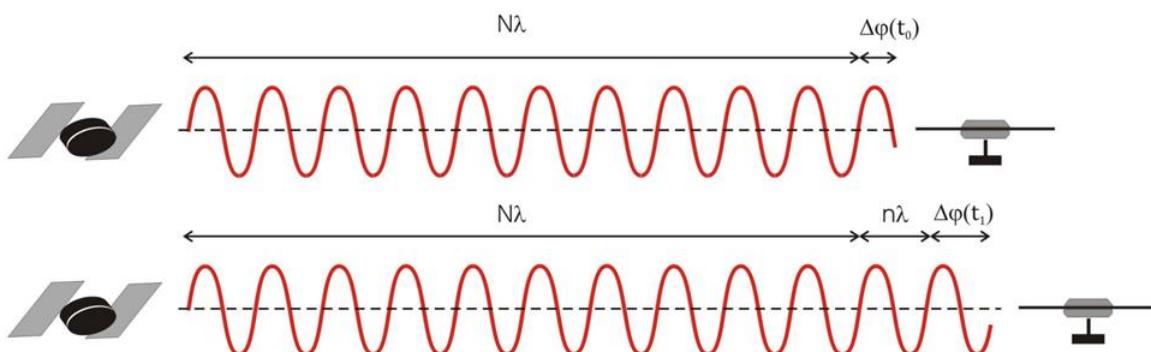
Ez a kódfázis-mérés alapegyenlete. Itt a d_i a valódi geometriai távolságot jelöli, vagyis a műhold és a vevő közötti távolságot azon időpontban (epochában), amikor a műhold órája δ_{sat} -ot, a vevő órája δ_{rec} -et jelez a GPS időskála szerint. A d_i valódi geometriai távolságban benne foglaltatik a vevő három ismeretlen térbeli koordinátája (x, y, z) és az i -edik műhold három ismert koordinátája (x_i, y_i, z_i).

A figyelmes olvasónak feltűnhet, hogy egy 1023 bites kódossal $1023 \cdot 293.052 \text{ m} = 299.792 \text{ km}$ -es távolságon belüli értékek határozhatók meg ismétlődés nélkül. Hogyan határozható akkor meg ezzel a műhold közelítőleg 20200 km-es távolsága, amibe közel 80 „kódszónyi” fénysebességgel megtett út fér bele? Bár erre vonatkozóan egyértelmű előírás a GPS

irodalomban nemigen található, a szoftver megoldásokból kiderül, hogy a számításhoz figyelembe veszik a mérendő objektum megközelítő helyzetét (ezt általában az előző méréskor kapott helyzettel veszik azonosnak), és kiszámolják, hogy ebbe a távolságba hányszor fér bele a 299 792.458 km ismétlődési kvantum. Végül ezt az értéket veszik alapul az új távolság meghatározásánál is.

A C/A kód 1023 bitjével szemben a P kód egy közel 2^{48} bit hosszúságú kódsorozat (a hossz egészen pontosan két szám szorzata: $15\ 345\ 000 * 15\ 345\ 037 = 2,3547 \cdot 10^{14}$), amely részsorozatokra van felosztva, természetesen szigorúan titkos és értékét változtatják. Az amerikai hadiipar nagypontosságú eszközei ezt a kódsorozatot ismerik, és így a P kód 10-szer nagyobb frekvenciából ($\lambda = 29.305$ m) következően mintegy 10-szer pontosabban tudják meghatározni a pszeudotávolságot. A teljes kódsorozat terjedési (és ezzel ismétlődési) ideje 23 017 555,5 sec $\approx 6\ 393,77$ óra $\approx 266,4$ nap ≈ 38 hét.

A vivőfázis-méréses módszer a műhold által kibocsátott vivőjel és a vevő által előállított referencia-jel fázisának összehasonlításából (fáziskülönbség-mérésből) áll. A vivőfázis-mérés eredménye egy olyan egész és törtrészt tartalmazó érték, amelynek egész része a vivőjel periódusok (ciklusok) számát jelenti, tört része pedig a fáziskülönbséget, ciklus-egységen kifejezve. A ciklusok számlálásának kezdetét nem ismerjük. Ha meghatározott időközönként (ún. epochánként) rögzítjük a fázisértékeket, akkor ezzel lényegében a műhold-vevő távolság változását mérjük a két időpont között ciklus-egységen. Az elsőként rögzített fázismérés eredményében csak a tört rész a „valódi” mérési eredmény, az egész rész számlálásának kezdete ismeretlen. Ennek az egész értéknek a meghatározása további modellezéssel lehetséges, ez a ciklus-többértelműség feloldásának problémája. A ciklus-többértelműség tehát a mérés kezdő időpontjában a mért távolságban benne foglalt egész periódusok számát jelenti. Ez az ismeretlen egész szám számítással (becsléssel) határozandó meg.



6.2.5.18 ábra – A ciklus-többértelműség problémája

A távolság meghatározását biztosító kódjel-sorozaton kívül a műholdak minden frekvencián olyan kiegészítő üzeneteket is sugároznak, amelyek a műholdak helyzetére és a mérésekkel terhelő hibákra vonatkoznak. Az adatok a földi követő állomásokról származnak és rendszeresen frissítésre (pontosításra, módosításra) kerülnek. Az ún. navigációs üzenetek továbbítása a vevőberendezések felé a vivőhullám kódmodulációjával történik (D kód). Az átviteli sebesség 50 bit másodpercenként (50 bps). Összesen 5 navigációs részüzenet (frame)

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 60. oldal
--	---	--

kerül továbbításra, ezek mindegyike 300 bit hosszúságú. Egy navigációs üzenet hossza tehát 1500 bit, így a teljes adatátvitelhez

$$T = 5 \cdot 300 / 50 = 30 \text{ s}$$

szükséges. Az üzenetrészek a következők:

- **A műholdak óra-pontosítási paraméterei**

A GPS időrendszerét, az ún. GPS időt a földi vezérlő állomásokon elhelyezett nagypontosságú frekvenciaetalonok (cézium és maser órák) valósítják meg. minden egyes műholdnak saját, független órája (cézium + rubídium óra), és ezáltal egy saját időrendszerre is van, amit műhold-időnek nevezünk. A földi vezérlő központ egyik feladata az egyes műholdak óráinak összehasonlítása a földi GPS idővel, azaz a műholdak órajavítási paramétereinek a meghatározása. Az órajavítási paraméterek egy hatványsor együtthatói (összesen három paraméter van), amelyeket óraállásnak és órajárásnak nevezünk. Ezeket az órakorrekciónak paramétereit továbbítják az egyes műholdak a vevők felé, mégpedig 30 másodpercenként, minden egyes üzenet első részében. Ezen órakorrekciónak jóvoltából a műhold órahibáját ismertnek tekintjük.

- **Durva pályaadatok (almanach)**

A műholdpálya Kepler-féle pályaelemeit értjük ez alatt, amelyekkel az a célunk, hogy bármely időpontban előre jelezhetünk a műhold helyzetét. A pályaadatok hibája ennél a megadásnál még nagy (néhány km-es), viszont ezek hosszabb időtartamra érvényes adatok. Az almanach tehát a hosszú távú (néhány napos, hetes, esetleg hónapos) előrejelzést, a mérés tervezését szolgálja.

- **Fedélzeti pályaadatok (broadcast ephemeris)**

A Kepler-pálya pontos elemei (a hat pályaelem továbbá a gyors és lassú lefolyású pálya menti korrekciós adatok), amelyek a perturbáció, a nehézségi erő változása, a Nap és Hold hatása miatti korrekciókat is tartalmazzák, és ez által lehetővé teszik a műhold helyzetének néhány (5-10) méter pontosságú meghatározását. Egy-egy műhold csak a saját fedélzeti pályaadatait sugározza 30 másodpercenként. Ezeknek az adatoknak az érvényessége rövidtávra - néhány órára - szól, mert időbeli változásuk számottevő.

- **Ionoszféricus modell paraméterei**

A légkör (más szóval atmoszféra) különböző rétegekre osztható, ezek elnevezése a Földtől távolodva: troposzféra, sztratoszféra, mezoszféra, ionoszféra. A navigációs műholdak mintegy 20 ezer kilométeres távolságban keringenek a Föld körül, az elektromágneses jelek útjuk egy részét vákuumban teszik meg, de végül a teljes légkörön áthaladva jutnak el a vevőhöz. A jelterjedés szempontjából két légrétegnek van kiemelt jelentősége: a Földközeli, mintegy 12 km-es vastagságú troposzfrának (ahol a légkör tömegének döntő része van), és a Földtől kb. 70 km – 1000 km távolsági tartományt kitöltő ionoszfrának (amelynek részecskéi a Nap ultraibolya sugárzása következtében ionizált állapotba kerülnek). A GPS-jelek frekvenciája 1,5 GHz körüli érték, ami kb. 20 cm-es hullámhossznak felel meg. Az elektromágneses hullámterjedés sebességét pontosan ismerjük ($c=299\ 792\ 458 \text{ m/s}$), de ez csak vákuumban tekinthető állandónak. A molekulák

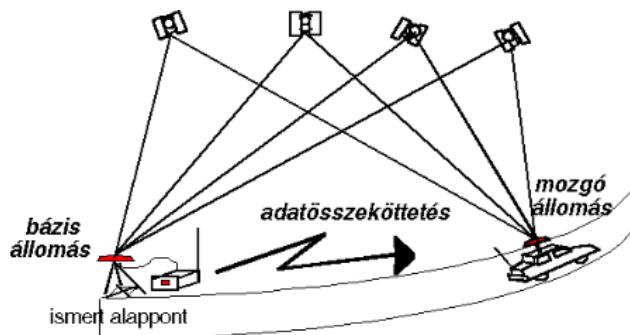
(és ionok) jelenléte a troposzférában, de különösen a szabad elektronok jelenléte az ionoszférában jelentős mértékben befolyásolja a rádióhullámok sebességét. Az elektronsűrűség (Total Electron Content = TEC) függvényében a jel késleltetést szorít, következményként tehát nagyobb távolságot mérünk, mint a valós geometriai érték. Az ionoszférikus késleltetés mértéke több tíz méter is lehet. Az ionoszférikus hatás erősen napszak-függő (a legnagyobb helyi idő szerint 14 órakor), de az évszakokhoz és a 11 éves ciklusú napkitörésekhez igazodó hosszú periódusú függése is van. Az ionoszférikus hatás csökkentésére globális modell (a vezérlő alrendszer mérései) alapján a műholdak korrekciós modell-paramétereit sugároznak.

- Műholdállapot**

Ez az üzenet a műhold működőképességről ad felvilágosítást. A GPS vevők kijelzőjén megjelenő elnevezések: „healthy/unhealthy” vagy „OK/BAD” is innen származtathatók.

Differenciális GPS (DGPS)

A GPS rendszer pontosságát növelő DGPS rendszer alapötlete az, hogy néhány száz kilométer távolságon belül kb. ugyanazok a zavaró tényezők hatnak a GPS mérőállomásokra. Ezért ha egy bázis- (referencia-) állomásról - aminek a koordinátái nagy pontossággal ismertek, és így a ráható zavaró tényezők is számíthatók - rádiójelekkel a korrigáló tényezőket folyamatosan a mérőállomásra küldjük, akkor ezzel a zavaró tényezőket részben kiküszöbölik.



6.2.5.19 ábra – A DGPS működési elve

Az álló vevőt egy ismert helyzetű pontra telepítik, melynek koordinátáit betáplálják a vevő memoriájába. A vevő meghatározza helyzetét és az ismert adatuktól való eltérést (a mért és a számított pszeudotávolságok különbségeit, a differenciát) valamilyen távközlési csatornán keresztül közli a mozgó vevővel, amelyik ezt a korrekciót a kijelzésnél figyelembe veszi. Differenciális GPS módszeren tehát általánosságban valós idejű, kódfázis-mérésen alapuló, relatív eljárást értünk, a DGPS rövidítést (Differential GPS) erre használjuk. A módszert kezdetben az óceánon, a parttól több száz km-re mozgó hajók pontosabb (néhány méteres) helyzet-meghatározására használták, amelyhez speciális, erre a célra szolgáló rádióadót és GPS referenciaállomást telepítettek és a vételhez is DGPS típusú, a rádiójelek vételére is alkalmas készülékre volt szükség. A kódfázis-mérésen alapuló DGPS pontossága néhány méter. A DGPS technika előnye az összes eddigi módszerhez képest a valós idejű (real-time) helymeghatározás.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 62. oldal
--	---	--

A DGPS rendszer tehát azt célozza, hogy a lehetőségeknek megfelelően növeljük meg az abszolút mérés pontosságát, semlegesítsük az SA hatását, de ne nekünk kelljen a referenciai-vevőt telepíteni, üzemeltetni, hanem annak adatait szolgáltatásként kapjuk. Minél több a felhasználó, annál inkább megéri egy ilyen DGPS korrekciós jelszolgáltatásnak a fenntartása. DGPS korrekciók formátumára szabványt dolgoztak ki az USA-ban, amit folyamatosan fejlesztenek. A DGPS korrekciók sugárzásának mai lehetőségei a következők:

- Mobil telefonon keresztül,
- Kereskedelmi rádió segédfrekvenciáján, RDS-en keresztül,
- Távközlési műholdon (Inmarsat),
- Speciális rádióadó segítségével.

Az eddigiekben vázolt, hagyományosnak tekinthető DGPS módszer esetében feltételeztük, hogy a referenciai-ponton számított korrekciók érvényesek minden felhasználónál. Ez nyilvánvalóan csak első közelítésben igaz, ezért a referenciai-vevőtől való távolság növekedésével a pontosság romlik. Ez a pontosság-csökkenés elkerülhető, ha egy több referenciai-pontból álló hálózatot hozunk létre, amelynek segítségével a különböző korrekciók modellezhetők, és helytől függően számíthatók. Szükség van továbbá az egyes korrekciók megkülönböztetésére is (távolságtól függő vagy attól független). Így a felhasználók felé a következő elkülönített korrekciókat kell továbbítani:

- Műhold pályaadatok a fedélzeti pályaadatok felváltására. Ezeket 15 percenként újítanák meg, és 1-2 percenként sugároznák.
- Regionális atmoszférikus modell-paraméterek. Óránkénti frissítésű 8 paraméter, szintén 1-2 percenként sugározva.
- Az egyes műholdak órajavítási paraméterei. Az SA hatása miatt a lehető legsűrűbb (másodperces) időközönként szükséges a továbbításuk.

Ebben a koncepcióban a nagy területre kiterjedő referenciai-pontok hálózata (GPS array) lényegében a NAVSTAR rendszer vezérlő alrendszerét mintegy felváltaná. Az elmúlt években több ilyen műhold alapú (vagyis geoszinkron holdak adattovábbítására épülő), nagy területre kiterjedő WADGPS rendszer kezdte meg működését. Az amerikai földrészt szolgálja ki a WAAS (Wide Area Augmentation System), a Japán szigeteken használatos a MSAS, Európában pedig az EGNOS (European Geostationary Navigation Overlay Service). A WADGPS előnye, hogy sűrű referenciai-állomások nélkül, nagy (kontinensnyi) területen egyetlen vevő használatával is elérhető lesz a méteres vagy a méteren belüli pontosság.

Milyen pontos a GPS ?

A rendszer megismerése után az emberek többségében az a kérdés fogalmazódik meg: milyen pontosságú helymeghatározás érhető el GPS-szel ? A kérdés nem válaszolható meg egyetlen egyértelmű számmal. Ennek az az oka, hogy a GPS-nek sokféle gyakorlati alkalmazása létezik. minden alkalmazásnak más a célja, technikai feltétele, és így a pontossága is. Ahhoz, hogy az adott célnak, a megkívánt pontosságnak legmegfelelőbb módszert választhassuk, el kell igazodnunk a GPS mérési módszerek között.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 63. oldal
--	---	--

Amennyiben a pszeudotávolságok meghatározásához a kód-összehasonlítás technikáját használjuk (ún. kódfázis-mérés), szükségképpen a kód frekvenciájához tartozó hullámhossz (1 kódbit idejéhez tartozó jelterjedési idő) szabja meg a pontatlanságot. Ez a standard C/A kód (standard SPS) esetében $\lambda = 293.052$ m, aminek a földfelszínre adódó vetülete ennél jóval kisebb pontatlanságot (5-50 m) okoz. Az SA torzítással 2000. májusig ezt a pontosságot rontották le 50-200 m tartományba P kód alkalmazása esetén (katonai PPS mérés) a 10-szeres kódfrekvencia alkalmazása miatt a pontosság is közel a 10-szeresére javul (1-10 m). Az időmérés és a zavaró pontatlanságok kiküszöbölésével a differenciális GPS a pontosságon még egy nagyságrendet tud javítani (0.5-2 m).

A nagyobb pontosságú megoldások, melyek a kódfázis-mérés helyett a vivőfrekvenciák fázisát próbálják meghatározni (ún. vivőfázis-mérés). Az L₁ és az L₂ frekvenciák 154 ill. 120-szorosai az alaposzcillátor frekvenciájának, ami a P-kód frekvenciájával egyezik meg. Ha sikerülne tehát a vivőfrekvencia pontos fázishelyzetét is megmérnünk, az előbbi pontatlansági értékek már a vivőfrekvenciák hullámhosszainál (19 ill. 24 cm-nél) kisebb értékre adódnának. Ehhez viszont lényegesen bonyolultabb mérőelektronikára van szükség, továbbá tudnunk kell, hogy a vivőfázis-mérésnél a végeredmény nem egyértelmű (ciklus-többértelműség problémája). Az egész hullámhosszak meghatározásához a vevők hardveresen annyiban tudnak hozzájárulni, hogy rögzíteni tudják az egyes műholdakra vonatkozó teljes hullámhosszak változását. Ha ugyanis a vevő mozdulatlanul áll egy pontban, és méri négy műhold távolságát, a műholdak mozgása következtében a távolságok (a teljes hullámok száma és a nem teljes hullám részek nagyságai - azaz a vivőfázis-mérési eredmények) változni fognak. Ezek a változások rögzíthetők és tárolhatók. A gyakorlatban a vivőfázis-mérésen alapuló módszerek rendszerint két **GPS** vevőt alkalmaznak. A cél ezeknél a módszereknél az úgy nevezett relatív helymeghatározás, ami azt jelenti, hogy két pont közötti távolság x, y, z komponenseit kívánjuk meghatározni. Az egyik vevőt az ismert referencia ponton üzemeltetik, míg a másik vevő a meghatározandó ponton üzemel. Mindkét vevő a megadott programnak megfelelően megméri a kijelölt műholdakra a vivő fázishelyzetét és rögzíti a hozzájáruló idővel együtt. A feldolgozási folyamatban (utófeldolgozásról van szó) a program a vivőfázis észlelések különbségeiből számolja a keresett koordináta különbségeket. A méréstechnikára számos megoldás született már, de kutatás tárgya ma is.

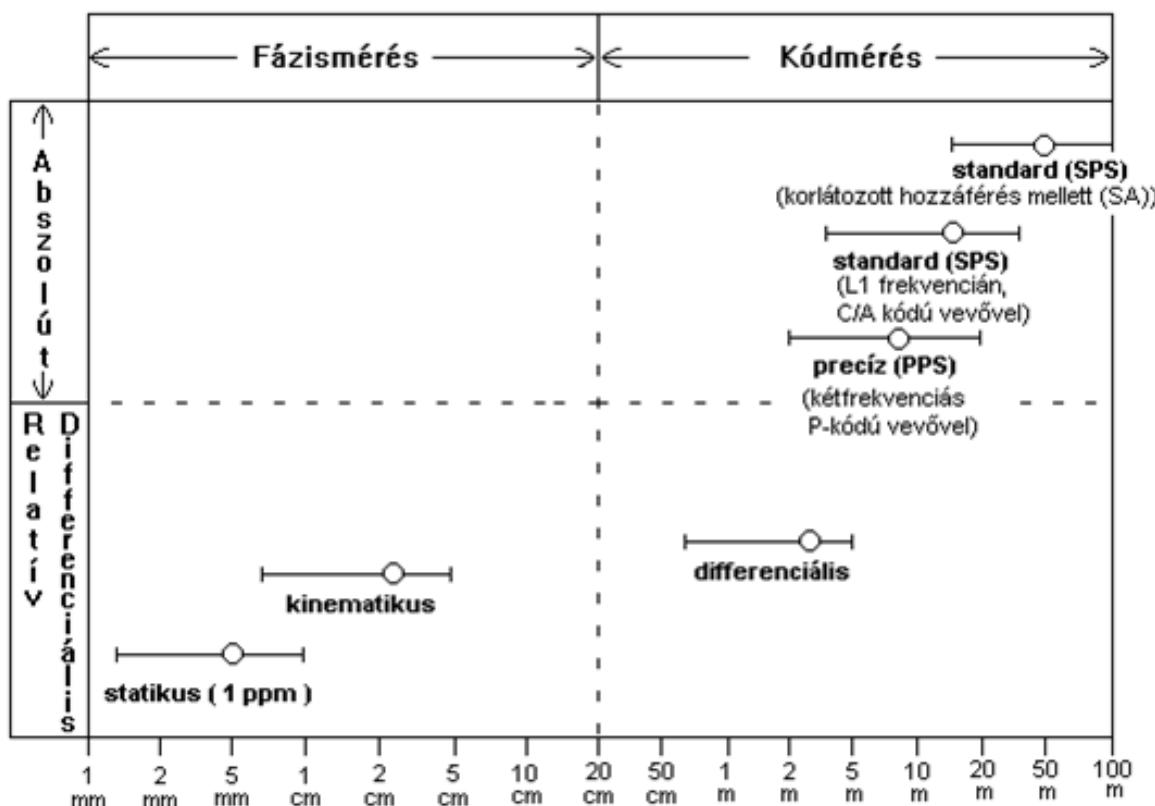
Statikus pontmeghatározás alatt azt értjük, hogy a vevő mozdulatlanul a mérendő ponton áll, míg **kinematikus** technikáról van szó, ha a vevő mozog mérés közben. Utóbbi minden pontatlannabb lesz, mivel a mérőobjektum helyzete a mérés alatt is változik - viszont ilyennek tekinthető valamennyi járműre szerelt helymeghatározó rendszer.

Fentiek lehetséges kombinációi a következő mérési eljárások:

- statikus abszolút pontmeghatározás
Viszonylag rövid mérési idő után, méréskelt (5-10 m) pontosságot érhetünk el a korábbiakban tárgyalt kódfázis-méréssel (kommersz elektronika alkalmazásával).
- kinematikus abszolút pontmeghatározás
A kódfázis-mérések felhasználásával, mozgó járművek helyzetét kb. 10-100 m-es pontossággal határozhatjuk meg.

- statikus relatív pontmeghatározás

Vivőfázis-mérés felhasználásával történik. Pl. a geodéziában ez a leggyakrabban alkalmazott módszer, melynek segítségével a GPS szélső pontossága (0.1-1 ppm) elérhető, ez már csak néhány milliméter hibát jelent kilométer nagyságú objektum távolságok esetén. Két vagy több álló vevő közötti vektor meghatározásán alapul és feltételezi egy referencia pont ismeretét. Az így kapott koordináták természetesen öröklik a referencia pont koordinátáinak esetleges pontatlanságát.



6.2.5.20 ábra – A GPS rendszerek pontossága

6.3 Mobil robotok navigációja akadályok között

A korábbi fejezetekben megismertedtünk a mobil robotok típusaival, illetve a velük kapcsolatos egyik legfontosabb kérdéssel, a helymeghatározás problematikájával. Azonban a mobil robot navigáció során önmagában a robot helyzetének ismerete még kevés, ez (és más információk) alapján megfelelő döntéseket is kell hoznunk, és megfelelő irányítási algoritmusokat kell alkalmaznunk a navigációs cél eléréséhez.

A mobil robotok tájékozódásáról szóló fejezetünk elején felvetett három fő kérdés közül a következőkben a „Hogyan jutok el a célba?” kérdés megválaszolására teszünk kísérletet. Előtte azonban hadd foglaljuk össze ismét – az előzőeket kissé kibővítve – a mobil robot navigáció alapvető részfeladatait:

- Lokalizáció („Hol vagyok?”)
- A cél megadása („Hová megyek?”)
- Térképalkotás („Hol vannak az akadályok?”)
- Mozgástervezés („Hogyan jutok el a célba?”)
- Mozgás végrehajtás (szabályozás)

A navigáció részfeladatainak illusztrálására tekintsük a minden nap autós közlekedés példáját. Ahhoz, hogy autóval sikeresen eljussunk „A” pontból „B” pontba, először is ismernünk kell „A” és „B” pontokat. Bár ez magától értetődően hangszik, ezzel márás érintettük a lokalizáció („A” pont ismerete, ahonnan indulunk) és a cél megadásának („B” pont, ahová el akarunk jutni) feladatát. Továbbá ismernünk kell az „A” és „B” pontokat tartalmazó térképet, amelyen látszanak a járható utak, illetve a mozgást korlátozó akadályok. Ezek után meg kell terveznünk saját útvonalunkat bizonyos elvek mentén (legrövidebb útvonal, leggyorsabb útvonal, bizonyos jellegű utak elkerülése stb.). Végül beülünk az autóba, és a tervet követve elvezetünk a célállomásra. A vezetés maga (forgalmi szituációk érzékelése, pedálok és kormánykerék kezelése stb.) jelenti a mozgás végrehajtását, vagy úgy is mondhatjuk, hogy a mozgásszabályozás folyamatát.

A mobil robotokkal kapcsolatban egyre elterjedtebben használjuk az „autonóm mobil robot” megnevezést, amely azt a törekvést fejezi ki, hogy robotjainkat egyre nagyobb fokú önállósággal ruházzuk fel, egyre magasabb szintű döntések meghozatalára tegyük alkalmassá. Az előző példánkban minden egyes részfeladatot az utazást tervező és végrehajtó ember végzett el. Attól függően, hogy a navigációs részfeladatok makkora részét ruházzuk át a járműre (vagy ha úgy tetszik, a mobil robotra), különböző fokú autonómiáról beszélhetünk. Természetesen nem csak az autók vagy az autószerű robotok esetén bonthatjuk a navigációt a fenti részfeladatokra, hanem bármilyen esetben, ahol akadályok között kell eljutni egy kezdeti helyzetből (pozícióból és orientációból) egy célpontba, az akadályokkal való ütközés nélkül. Foglaljuk össze tehát, hogyan valósul meg a navigáció különböző fokú autonómiával ellátott robotok esetében:

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 66. oldal
--	---	--

- Automatikus mozgásszabályozás.** Ebben az esetben csupán a mozgás végrehajtását bízzuk a robotra, minden mászt a kezelő végez. Ez a (táv)irányítás klasszikus esete. Az operátor viszonylag alacsony szintű parancsokat ad (pl. irány, sebesség megadása), a robot pedig ezeknek megfelelően alacsony szintű szabályozásokat végez. Ezt a kategóriát nem is nevezhetjük még *autonómnak*, inkább csak bizonyos folyamatok *automatizálásáról* beszélhetünk. Az autós példánál maradván ide sorolhatjuk a legtöbb vezetőt támogató rendszert (tempomat, menetstabilizátor, ABS, sávtartás stb.)
- Autonóm mozgástervezés.** Itt már a robot kognitív döntéseket hoz, tervezí a saját mozgását az akadályok és a célpont ismeretében. Fontos, hogy mind a navigációs célt, mind a környezet térképét (az akadályok alakját és elhelyezkedését), mind a robot helyzetét ismertnek tekintjük a navigáció teljes folyamata alatt. Azt viszont, hogy ezek alapján hogyan mozogjon a robot a cél elérése érdekében, már nem az operátor, hanem a robot irányítórendszerre határozza meg. Mint később látni fogjuk, a mozgástervezés részfeladatát globális és lokális perspektívából is figyelhetjük. Autós példánk esetén globális tervezésről beszélünk, amikor a GPS-alapú navigációs eszközünk megtervezzi az útvonalunkat a megadott célállomásig (bár itt autonómiáról nem beszélhetünk, hiszen az alárendelt mozgásvégrehajtást nagyrészt a sofőr végzi). Autonóm lokális mozgástervezésre és végrehajtásra példaként említhetjük az autókban egyre inkább elterjedő automata parkoló funkciókat.
- Autonóm térképalkotás.** Egy autonóm robotnak tudnia kell alkalmazkodni a környezet változásaihoz. Az autonóm navigációhoz nem elegendő egy globális térkép pusztta ismerete, hanem képesnek kell lenni a térkép frissítésére, módosítására is a környezetben tapasztalt változások alapján. Gondoljunk bele, egy autonóm, önvezető autónál elképzelhetetlen lenne ennek hiánya: egy váratlan útlezárás, vagy forgalmi akadály meghiúsíthatná a navigációs cél elérését. A jelenleg folyó intelligens autó fejlesztések célja az autonómia ezen fokának elérése: az autó képes legyen felismerni környezetét, megtervezni és végrehajtani mozgását, reagálva a nem várt változásokra. A vezető (vagy inkább már csak utas) feladata ebben az esetben csupán annyiban állna, hogy jelölje ki a navigációs célpontot, a többöt elvégzi az autonóm jármű.
- Teljes autonómia.** A felsorolásból már csak a cél autonóm kiválasztása maradt ki. Egy teljes önállósággal rendelkező robot esetében a feladatot magasabb szinten határozzuk meg, amelynek a mozgástervezés és végrehajtás, de még a navigációs célpont(ok) kiválasztása is csak részfeladata.

Ebben a fejezetben a navigációs részfeladatok közül a mozgástervezés problematikáját járjuk körül.¹ A mozgástervezés célja egy olyan mozgássorozat meghatározása, amelyben a kiindulási állapot (kezdeti pozíció és orientáció) és végső állapot (célpozíció és orientáció) között a mozgás úgy valósul meg, hogy egyik köztes pillanatban sem ütközik akadálynak a robot. A mozgástervezés eredménye tehát egy olyan – a mozgásszabályozásnak alapjelül adható – folytonos állapot-trajektória (időfüggvény), amely kizárolagosan ütközésmentes állapotokat tartalmaz. Az állapot-trajektória két fő komponensből áll össze, amelyek mentén maga a mozgástervezés is két részfeladatra bontható:

¹ A mozgásszabályozás, a térképalkotás témakörei, illetve a legmagasabb szintű autonómia, amelynél a cél kiválasztása is a robot feladata, túlmutatnak jelen tárgy keretein.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 67. oldal
--	---	--

- **Pályatervezés.** Ez a lépés a mozgás során felveendő állapotoknak, illetve azok sorrendjének meghatározását jelenti. Ha pl. a robot állapotának a pozíóját (és orientációját) tekintjük, akkor ebben a fázisban a robot által bejárandó geometriai pályát, mint térgörbét tervezzük meg.
- **Időparaméterezés.** A geometriai tervezésen túl a mozgásszabályozás alapjeléhez meg kell adnunk az időparaméter eloszlását is a bejárandó útvonal mentén (mikor hol legyünk). Az időparaméterezés peremfeltételeit főként maga a pályageometria, valamint a robot dinamikus tulajdonságai (tehetetlenségek, gyorsuláskorlátok, tapadás) adják.

6.3.1 A konfigurációs tér

A robotmozgás tervezésénél legtöbbször a kinematikai megközelítést használjuk, amikor is a robot állapotán (kvázi)statikus állapotot értünk. Ez azt jelenti, hogy pályája mentén a robot – elvben – bármikor megállítható, és az adott állapot tetszőleges ideig fenntartható. Ebben a megközelítésben az állapot általában geometriai pozíciót (és orientációt) jelent valamilyen koordinátarendszerben, azonban ezek deriváltjait (sebességek, gyorsulások) nem tekintjük az állapotvektor részének, mint a dinamikus rendszerek esetén.

Hogyan írhatjuk le egy robot (kvázi)statikus állapotát általános esetben? A konkrét geometriai tulajdonságok, amelyek megadják az állapotváltozók számát és jellegét, nagymértékben robotfüggőek. Lássunk néhány példát:

- A helyhez kötött, „klasszikus” robotkaroknál a robot statikus állapotát (térfelületi helyzetét) kétféleképpen adhattuk meg, egyrészt a csuklóváltozók segítségével, másrészt a végberendezés Descartes-koordinátarendszerbeli pozíójával és orientációjával. A kétféle felírás között a direkt és inverz geometriai feladatok megoldása jelentette az átájárást. Azt is láttuk, hogy míg a direkt geometriai feladat megoldása mindenkor egyértelmű, addig az inverz geometriai feladat több megoldást is adhat, vagyis egy végberendezés-helyzethez többféle csuklóállás is tartozhat. Ráadásul akadályok jelenlétében nem elég csupán a végberendezés helyzetére figyelnünk, hanem a teljes robotkar megfelelő geometriai elhelyezkedését és ütközésmenetességét biztosítanunk kell. Emiatt a robot állapotának egyértelmű leírására a csuklóváltozók vektora, más néven a *csuklókonfiguráció* alkalmas leginkább.
- Repülő, illetve vízben mozgó mobil robotok esetén (tegyük fel, hogy a robot egyetlen merev testtel modellezhető) a robot statikus állapotát egyértelműen megadja a Descartes-koordinátarendszerben kifejezett pozíciója és orientációja.² Ez ugyanúgy, mint a robotkarok végberendezése esetén, három pozíció- és három orientáció-

² Ne vezessen félre minket az állapotok „statikus” voltának említése. Tudjuk, hogy a valóságban a gravitáció jelenléte miatt nehéz lenne megvalósítani pl. repülő robotok esetén bizonyos pozíciók és orientációk statikusságát. Elméletben (vagy pl. gravitációmentes téren) azonban minden lehetséges. Ezen a ponton vonatkoztassunk el a konkrét fizikai megvalósítástól, és a „repülő robot” megnevezést tekintsük a „háromdimenziós tér minden irányában szabadon mozgatható mobil robot” szinonimájának.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 68. oldal
--	---	--

koordinátát, vagyis 6 szabadságfokot jelent. A pozíció és orientáció hatdimenziós vektorát itt is szokás *konfigurációnak* nevezni.

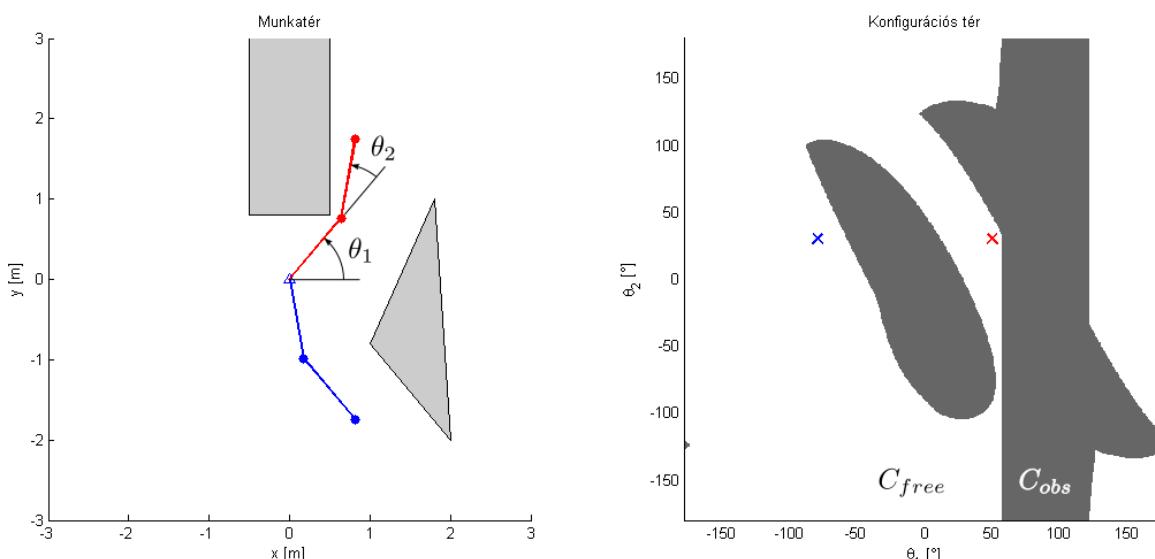
- Síkban mozgó, kerekeken guruló robotok statikus állapotát három változóval adhatjuk meg: a két (x és y irányú) síkbeli pozíció-koordinátával, illetve az egydimenziós (z tengely körüli elfordulást megadó) orientációval. Ezt a háromdimenziós vektort itt is a robot *konfigurációjának* hívjuk.

A fenti példákban a konfigurációs változók fizikai jelentése más és más az éppen vizsgált robottípustól függően: kifejezhetnek csuklóváltozó-értéket, Descartes-térbeli pozíciót, szöghelyzetet. A változók száma is különbözik: egy m -szabadságfokú robotkarnál m darab, repülő vagy vízben mozgó robotnál 6 darab, míg síkban mozgó robot esetén 3 darab koordinátával írható le a robot konfigurációja. Azonban van egy közös tulajdonságuk: egyértelműen megadják az adott robot aktuális állapotát.

A *konfiguráció* tehát egy olyan n -dimenziós vektor, amely egyértelműen jellemzi a robot – statikus – állapotát. Úgy is fogalmazhatunk, hogy a robot bármelyik pillanatban egyértelműen megfeleltethető az n -dimenziós *konfigurációs tér* egy pontjának. Mivel a robot a konfigurációs térben egyetlen ponttal (vektorral) leírható, a pályatervezés feladata robottípustól függetlenül egy pontszerű, kiterjedés nélküli test pályájának megtervezéseként interpretálható a konfigurációs térben.

E kijelentés jelentőségének megértéséhez röviden térijünk vissza a robotkarok pályatervezéséhez. Adott egy m -szabadságfokú robot, amely m darab csuklóval és $m+1$ szegmenssel rendelkezik. Olyan pályát kellene terveznünk, amely mentén úgy mozog a robot, hogy egyrészt a végberendezés az előírt helyzetbe kerüljön, másrészt a robot egyetlen szegmense se ütközzön a környező akadályokba. Tegyük fel, hogy a robot szegmenseinek geometriai leírását, valamint a környező akadályok pontos modelljét is ismerjük. minden egyes konfigurációban – csuklóállásban – a robot más és más alakot vesz fel, amelyet a direkt geometriai feladat megoldása segítségével pontosan ki is tudunk számolni. Mivel a környezet modelljét is ismerjük, egy megfelelő algoritmus segítségével azt is eldönthetjük, hogy a robot egy adott konfigurációban ütközne-e valamelyik akadályal vagy saját magával. Azonban hogyan modellezük a robot pályáját? A robotkar mozgás közben egy meglehetősen nehezen leírható térrész „söpör” végig a saját munkaterében. A tervezéshez ilyen, három dimenziós kiterjedéssel rendelkező „térfogati pályákat” kellene alkotnunk, ráadásul ütközésmentesen, amely geometriailag nem könnyű feladat. Ezzel ellentétben, mint említettük, a konfigurációs térben a robot egyetlen ponttal teljes mértékben leírható. Tehát itt a pályatervezés is nagymértékben egyszerűsödik – természetesen adott esetben a dimenziószám növekedése árán. A pályatervezés a konfigurációs térben nem más, mint egy egydimenziós ütközésmentes térgörbe megalkotása.

Már csak egyetlen kérdés maradt, mielőtt nekikezdhetünk a tervezésnek. Igaz, hogy a robot egyetlen ponttá egyszerűsödik a konfigurációs térben, de hogyan modellezük az akadályokat ebben a felírásban? A válasz meglepően egyszerű: a konfigurációs térbeli akadályok (vagy más szóval *konfigurált akadályok*) azon konfigurációk halmazát jelentik, amelyekben a robot ütközésben lenne. A dolog illusztrálásához tekintsük az alábbi ábrát:



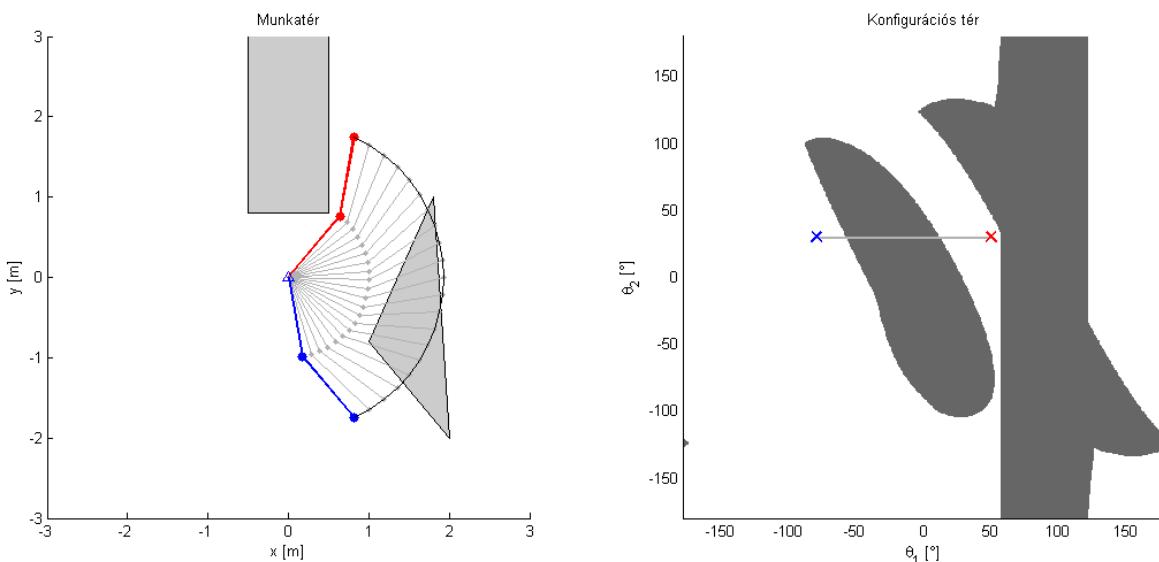
6.3.1 ábra – Kétszabadságfokú robotkar konfigurációs tere

Egy egyszerű kétszabadságfokú, síkban mozgó robotkart látunk, amely két rotációs csuklóval rendelkezik. Az ábra bal oldali része a robot munkaterét ábrázolja két akadályjal, illetve a kezdeti (piros) és célkonfigurációval (kék). A jobb oldalon ábrázoltuk a robot konfigurációs terét, a konfigurált akadályokkal együtt. Itt szintén piros, ill. kék pontokkal jelöltük a kezdeti és célkonfigurációkat. Általános esetben a következő jelöléseket használjuk:

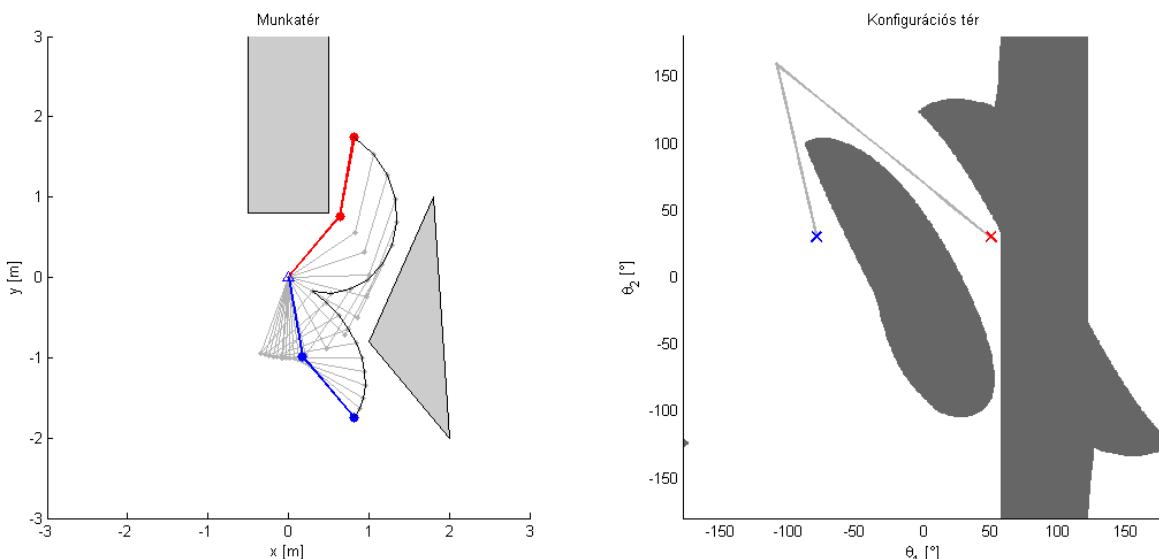
- W – Munkatér (workspace)
- C – Konfigurációs tér (configuration space, C-space)
- C_{obs} – Konfigurált akadályok halmaza (configuration obstacle)
- C_{free} – Szabad tér (free space), $C_{free} = C \setminus C_{obs}$
- $\mathbf{q} \in C$ – Konfiguráció (a konfigurációs tér egy pontja)

Ebben az egyszerű – és könnyen ábrázolható – esetben mind a munkatér, mind a konfigurációs tér kétdimenziós, a kettő viszont lényegesen különbözik. A munkatér koordinátarendszerének tengelyei pozíciót fejeznek ki, és nagyságuk elvileg bármekkora lehet $-\infty$ és $+\infty$ között. Azonban a konfigurációs változók a rotációs csuklók miatt szögfordulást reprezentálnak, amely egy periodikus mennyiség, így a koordinátatengelyek is „átfordulnak” a $\pm 180^\circ$ határán.

A kezdeti és a célkonfiguráció közötti útvonaltervezéskor akadálymentes esetben a legegyszerűbb módszer a lineáris interpoláció lenne. Azonban akadályok jelenlétében ez nem minden célravezető, ahogy azt a 6.3.2 ábra is mutatja. Ehelyett szükségünk van megfelelő pályatervezési algoritmusokra, amelyek segítségével a konfigurációs térben olyan térgörbék alkothatunk, amelyek sehol sem metszik a konfigurált akadályokat. Egy egyszerű, kézzel tervezett, szakaszosan lineáris ütközésmentes pályát láthatunk a 6.3.3 ábrán.

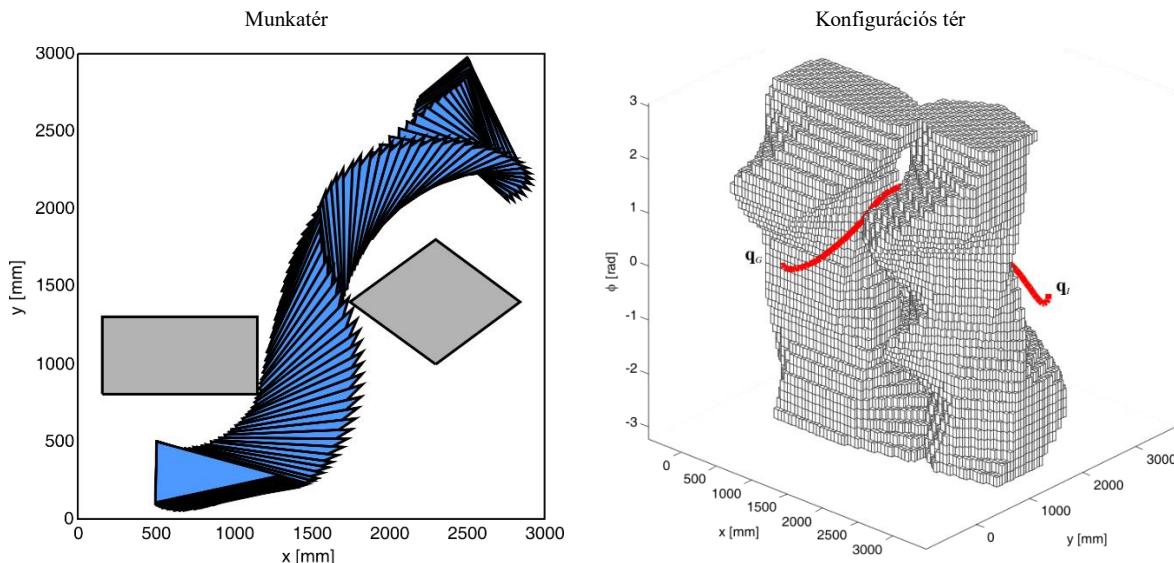


6.3.2 ábra – Lineáris interpoláció a konfigurációs térben



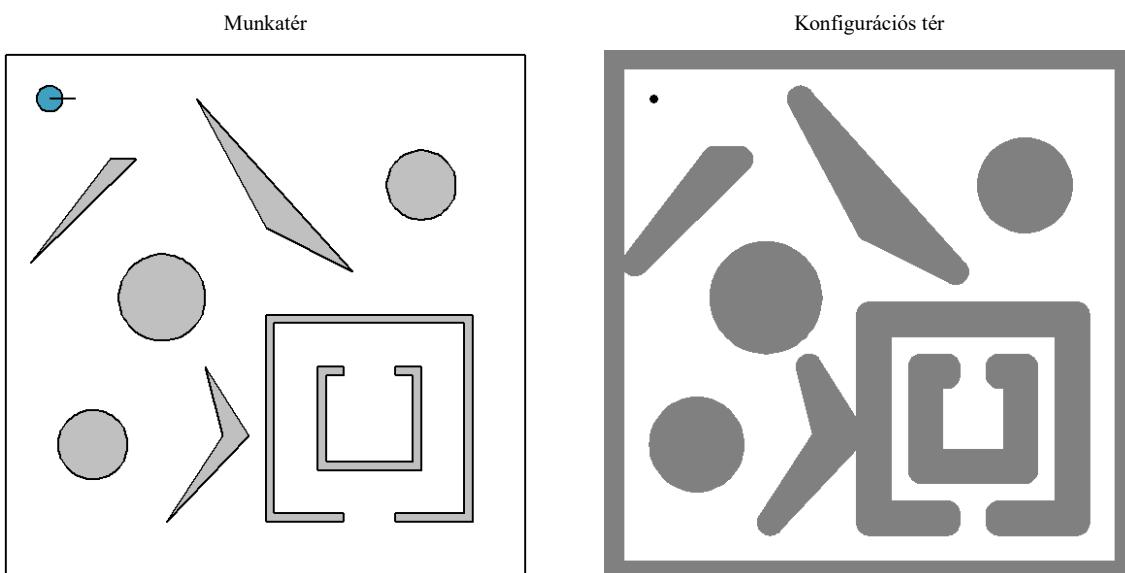
6.3.3 ábra – Ütközésmentes pályatervezés a konfigurált akadályok figyelembe vételével

Hogyan alakul minden ez a mobil robotok esetén? Mint már említettük, a síkban mozgó mobil robotok konfigurációja háromelemű vektor: $\mathbf{q} = (x, y, \varphi)$, ennek megfelelően a konfigurációs terük is háromdimenziós lesz, amelyet még viszonylag jól láthatóan lehet ábrázolni. A 6.3.4 ábra egy háromszög alakú mobil robot (mintavételezett) konfigurációs terét, illetve egy ütközésmentes pályáját mutatja két egyszerű, sokszög alakú akadály esetén. Mivel itt a harmadik konfigurációs változó az orientáció, ami megint csak periodikus mennyiség, ezért a függőleges tengely „átfordulása” itt is megfigyelhető.



6.3.4 ábra – Síkban mozgó mobil robot konfigurációs tere

Ha az egyszerűség kedvéért feltesszük, hogy a robotunk kör (Pontosabban henger) alakú, akkor az akadályok és a velük való ütközés szempontjából az orientáció elveszti jelentőségét. Ebben az esetben a konfigurált akadályok a 6.3.4 ábrán látott csavart forma helyett hengerszerűek lesznek, és a konfigurációs teret nyugodtan ábrázolhatjuk „felülnézetből”, a φ -tengely elhagyásával. Ezt láthatjuk a 6.3.5 ábrán is, ahol a munkatér és a konfigurációs tér között már csak annyi a különbség, hogy az akadályokat „felfújtuk” a robot sugarának megfelelően, a robot pedig kiterjedés nélküli, pontszerű lett. Ha tovább egyszerűsítenénk a dolgot, és a robotot már alapból pontszerűnek feltételeznénk, akkor a munkatér és a konfigurációs tér teljesen megegyezne.



6.3.5 ábra – Síkban mozgó, kör alakú mobil robot konfigurációs tere

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 72. oldal
--	---	--

A pályatervezési algoritmusoknak nagyon fontos tulajdonsága, hogy hány dimenziós konfigurációs térben alkalmazhatók. Egyszerűsége miatt számos tervező algoritmus él a pontszerű vagy kör alakú robot feltételezéssel, és csak kétdimenziós konfigurációs térben szolgáltat megoldást. Az általánosabb, dimenziószámtól függetlenül alkalmazható módszerek általában bonyolultabbak, viszont nagy előnyük, hogy elvben tetszőleges robottípusnál alkalmazhatók, legyen az tetszőleges alakú mobil robot, m -szabadságfokú robotkar, vagy akár több robotból álló rendszer.

6.3.2 Mozgástervezési módszerek osztályozása

A mozgástervezési algoritmusok többféle szempont szerint is csoporthosíthatók. Megkülönböztethetjük a módszereket például az alkalmazhatósági területeik szerint, vagyis aszerint, hogy milyen típusú konfigurációs térben és milyen robot esetén képesek a feladat megoldására. Kategóriákat állapíthatunk meg

- a konfigurációs tér dimenziószáma (pl. 2, 3, vagy több dimenziós),
- a konfigurált akadályok alakja (pl. sokszög, síklapokkal határolt térfelületi, tetszőleges alakú akadályok),
- a robot alakja (pontszerű, kör/gömb alakú, vagy tetszőleges),
- a robot kinematikája (holonomikus vagy nem holonomikus)

és még számos más szempont alapján. Mindezek részletes tárgyalására itt nincs lehetőségünk, azonban szeretnénk kiemelni egy nagyon fontos csoportosítási szempontot, amellyel a különböző módszerek működésének lényegét ragadhatjuk meg. Eszerint a mozgástervezési módszereket két nagy csoportra oszthatjuk:

- **Globális mozgástervezési módszerekre**, illetve
- **lokális (reaktív) akadályelkerülési módszerekre**.

A két csoport képviselői a lokalizáció kívül a navigációs feladat szinte minden más komponensében különböznek. Bár a mozgástervezés részleteit tárgyaljuk, fontos tudni, hogy a két kategória képviselői különböző hatással vannak, vagy eltérő igényeket támasztanak a cél megadása, a térkép tulajdonságai, illetve a mozgásszabályozás részfeladataira nézve.

A globális tervező algoritmusok képviselik a mozgástervezés „klasszikus” ágát. Itt ismertnek tekintjük a környezet térképét, és akár tetszőleges bonyolultságú akadályelrendezés esetén is garantálható a megoldás megtalálása, amennyiben az létezik. A tervezés a teljes konfigurációs térben történik, ennek megfelelően a kezdeti és a célállapot is egy-egy konfigurációval van megadva. (Ezt úgy kell érteni, hogy pl. egy síkban mozgó robot esetén nem csak a célpontot, hanem az ott felveendő orientációt is előírhatjuk.) Itt, ahogy korábban láttuk is, a mozgástervezés és a végrehajtás időben elkülönül egymástól. Először egy geometriai pályagörbét tervezünk, majd elosztjuk az időparamétert a pálya mentén, végül pedig az alárendelt mozgásszabályozás feladata a robot végigvezetése az előírt trajektorián. Ennek megfelelően ezek a módszerek leginkább off-line tervezésre alkalmasak. Globális módszerek alkalmazása esetén a nem várt környezeti változások kezelése nehézkes, legtöbbször a teljes, még a célig hátralévő mozgássorozat újratervezését jelenti, ha a robot nem várt akadályal kerül szembe.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 73. oldal
--	---	--

Ezzel szemben a lokális akadályelkerülési módszerek nagyban egyszerűsítik a mozgástervezés „klasszikus” megközelítését. Reaktív módszereknek is nevezzük ezeket, mert a hangsúly a lokális érzékelésen és az arra való gyors reakción van. Ezek a módszerek általában nem igényelnek globális térképet, hanem a robotra szerelt (pl. távolságmérő) szenzorok pillanatnyi, lokális információi alapján generálják a mozgatóparancsot. Ennek megfelelően nem is különül el a mozgástervezés és végrehajtás fázisa, hanem a pillanatnyi állapot és a szenzormérések alapján közvetlenül, on-line módon generálódik a mozgás. Mindennek az az eredménye, hogy a robot alkalmassá válik a nem várt környezeti változások érzékelésére és az azonnali megfelelő beavatkozásra. A lokális jelleg miatt azonban ezek a módszerek nem alkalmasak a teljes globális navigáció megvalósítására, különösen a sok akadályt és szűk szabad tereket tartalmazó környezetben. A lokális információk alapján meghozott pillanatnyi döntések miatt könnyen lokális minimum jellegű csapda-szituációba kerülhet a robot. A térkép használatának hiánya és az egyszerűsített tervezési folyamat miatt ezek a módszerek általában a robot alakjával nem töröknek (körrel közelítik), és az elérődő cél is csupán egy megadott pozíció, orientáció előírása nélkül.

Mint láttuk, a kétféle megközelítés rengeteg szempontból különbözik, ráadásul mindeneknek vannak előnyei és hátrányai is. A mozgástervezés és az akadályelkerülés³ ugyanannak a problémának két olyan komplementer megközelítését jelenti, amelyek közül nem választani, hanem a kettőt együttesen alkalmazni érdemes. Egy mobil robot navigációs rendszerének két, egymás fölé rendelt logikai szintjén az esetek többségében tökéletesen megférnek egymás mellett. Ezt például úgy képzelhetjük el, hogy a globális térkép alapján megtervezett útvonal köztes pontjait adjuk célponktként a reaktív akadályelkerülő algoritmusnak, így a robot képes lesz a váratlanul felbukkanó akadályok elkerülésére, miközben beragadás nélkül halad a globális cél felé. A kétféle megközelítés tulajdonságait a 6.3.1 táblázat foglalja össze.

Szempont	Globális mozgástervezés	Lokális akadályelkerülés
Cél megadása	Teljes konfiguráció	Csak pozíció (orientáció tetszőleges)
Környezet	Ismert	Ismeretlen vagy változó
Térkép	Globális, pontos térkép	Nincs térkép, csak lokális mérések
Robot alakja	Pontos modell	Általában körnek tekintik
Tervezés módja	Tervezés és végrehajtás külön fázisban	Tervezés és végrehajtás nem különül el
Megoldás jellege	Globális megoldás garantálható	Csapda-szituációkra érzékeny (beragadás)
Számításigény	Magasabb (off-line végrehajtás)	Alacsonyabb (on-line végrehajtás)

6.3.1 táblázat – Globális és lokális mozgástervezési módszerek összehasonlítása

³ Az irodalomban sokszor egyszerűen csak mozgástervezés (motion planning) és akadályelkerülés (obstacle avoidance) néven illetik ezt a két kategóriát, a „globális” és a „lokális” jelzők elhagyásával.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 74. oldal
--	---	--

6.3.3 Pályatervezési módszerek

Amint korábban már említettük, a mozgástervezés két részből áll: a geometriai pályatervezésből és az időparaméterezésből. Ezek közül itt a pályatervezési feladat megoldására mutatunk néhány példát. Az itt bemutatott módszerekben közös, hogy a tervezés lényege az n -dimenziós konfigurációs tér topológiájának, a szabad tér összefüggőségének megragadása egy jóval kevesebb adattal leírható formában. Erre a célról ezek a módszerek gráfokat használnak. A gráfok itt nem csupán egy absztrakt adatstruktúrát jelentenek, hanem konkrét geometriai tartalmuk is van. A csomópontok és élek konkrét térbeli pályapontoknak, ill. pályaszakaszoknak feleltethetők meg. Így az n -dimenziós szabad tér egy lényegében egydimenziós útvonalak hálózatát reprezentáló gráffal megragadható, és a geometriai pályatervezés egy gráfban való keresésre vezethető vissza. Az alább bemutatott három algoritmus tulajdonképpen „csupán” a gráf felépítésének folyamatában különbözik.⁴

A gráf alapú globális pályatervező algoritmusok között megkülönböztetünk *egzakt* és *mintavételes* módszereket. Az első csoportba tartoznak azok az algoritmusok, amelyek információvesztés nélkül, pontosan képesek modellezni a szabad teret a gráf segítségével. Ehhez szükségük van a konfigurált akadályok pontos modelljére. Közös előnyös tulajdonságuk a *teljesség*, ami azt jelenti, hogy véges idő alatt képesek találni megoldást, ha a feladat megoldható, illetve azt is képesek véges idő alatt megmondani, ha nem létezik megoldás. A mintavételes algoritmusok ezzel szemben valamilyen elv szerint mintákat vesznek a konfigurációs térből, és ezeket próbálják ütközésmentes útvonaldarabokkal összekötni. Ezekre jellemző, hogy a konfigurált akadályok explicit modellje nélkül is alkalmazhatók (ami nagyobb dimenziószám esetén igen komoly előny), csupán egy ütközésdetektáló modulra van szükségük, amely bármelyik konfigurációról megmondja, hogy ott ütközik-e a robot vagy sem. Hárányuk, hogy nem teljesek, azaz bár a mintaszám növekedésével egyre valószínűbb, hogy megtalálják a megoldást (ha az létezik), erre nincs véges időben garancia. Ráadásul nem tudják megkülönböztetni a létező, csak még nem talált megoldást a nem létező megoldás esetétől. A következőkben két egzakt és egy mintavételes pályatervezési algoritmust mutatunk be.

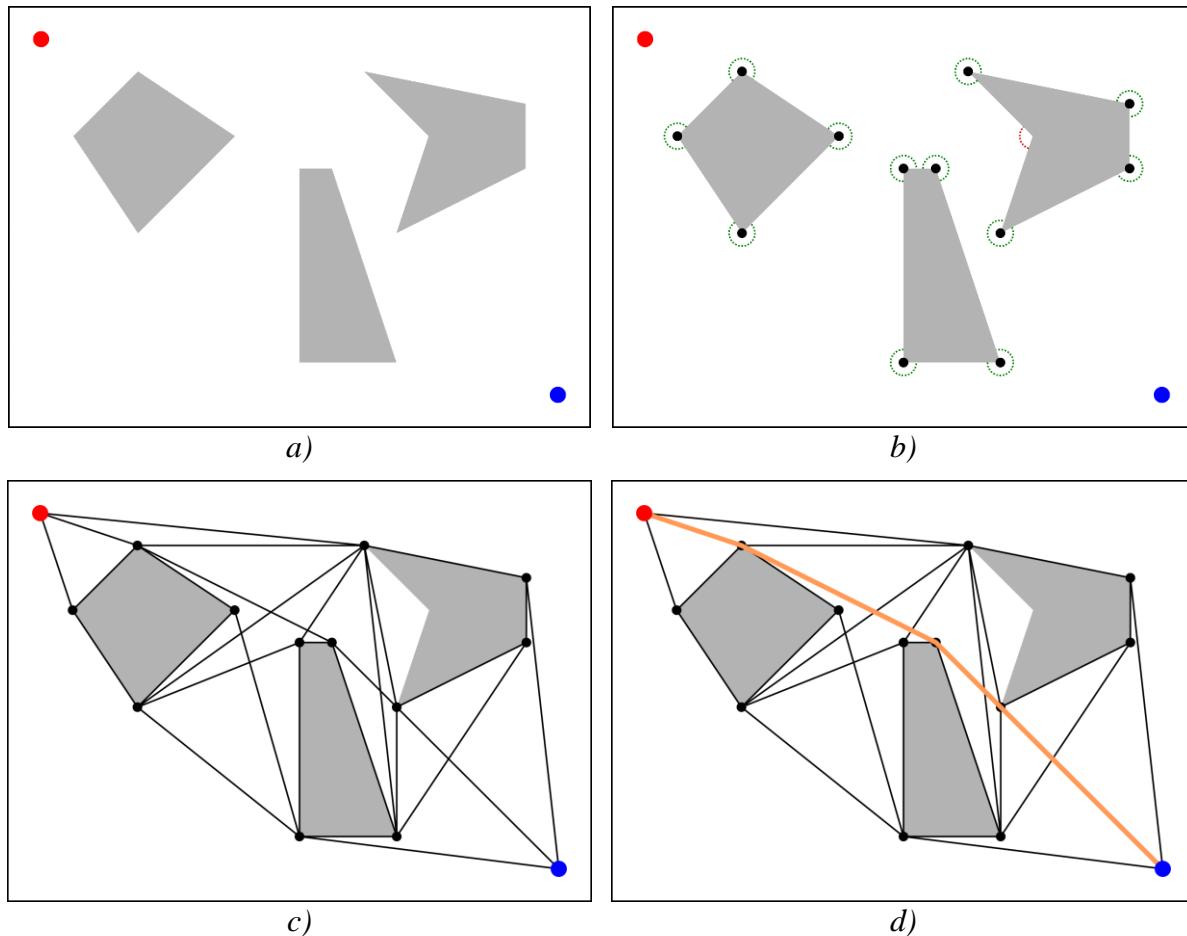
6.3.2.1 Láthatósági gráf

Első példánk a gráfot közvetlenül az akadályok geometriai modellje alapján építi fel. Azzal a feltételezéssel élünk, hogy a konfigurációs tér kétdimenziós, és az akadályok sokszögekkel modellezhetők. A láthatósági gráf csomópontjai a sokszögek csúcspontjai, valamint a kezdeti és célkonfiguráció lesznek, élei pedig ún. láthatósági élek. Ez utóbbi azt jelenti, hogy azokat a csomópontokat kötjük össze élekkel – egyenes vonalakkal –, amelyek kölcsönösen „láthatók”, vagyis az őket összekötő egyenes szakasz nem ütközik akadályba. A gráf éleit hosszukkal súlyozzuk, és egy legrövidebb utat kereső algoritmussal (pl. Dijkstra algoritmus) határozzuk meg a végleges pályát.

⁴ Léteznek nem gráf alapú tervezési módszerek is, ezekre a jelen anyagban terjedelmi korlátok miatt nem tudunk kitérni.

Ennek a módszernek előnye, hogy hossz tekintetében optimális útvonalat szolgáltat. Éppen ebből fakad viszont az a hátránya, hogy mivel a megtervezett pálya az akadályok csúcsain vezet keresztül, a robot szükségképpen „súrolja” az akadályokat. Ez biztonságossági szempontból nem a legjobb, és például az akadályok előzetes, virtuális kibővítésével orvosolható.

A láthatósági gráf módszer fent leírt alapváltozatát ritkán alkalmazzák, a gyakorlatban helyette a számításigény szempontjából kedvezőbb, módosított változatot, a *redukált láthatósági gráf* módszert érdemes használni. A redukció lényege, hogy bizonyos „fölösleges” éleket ne szerepeltesünk a gráfban, memóriát és az útvonalkeresésnél számítási időt spórolva ezzel.

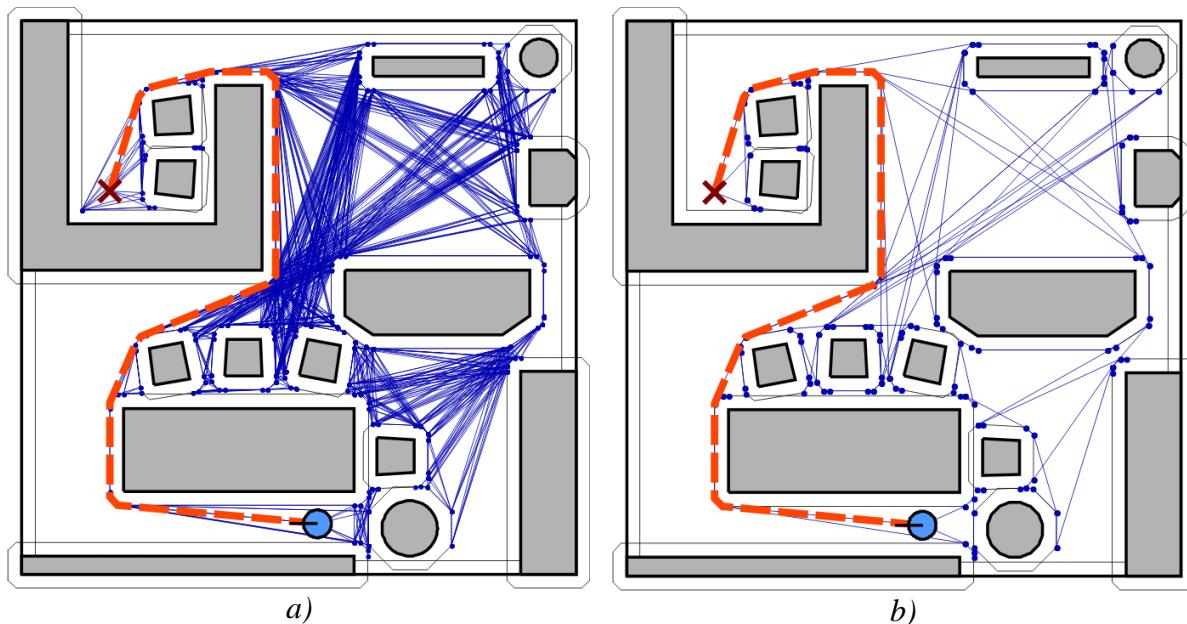


6.3.6 ábra – Redukált láthatósági gráf alapú pályatervezés. a) Akadályok, kezdeti és célkonfiguráció, b) Reflex csúcsponkok (zöld) és nem reflex csúcsponkok (piros), c) Redukált láthatósági gráf, d) Legrövidebb útvonal

A redukált láthatósági gráf alapú pályatervezést illusztrálja a 6.3.6 ábra. Lépései a következők:

1. A csomópontoknál ne minden akadály-csúcspontot, hanem csak az ún. reflex csúcspontokat vegyük figyelembe. Reflex csúcspontnak nevezzük azokat a csúcspontokat, amelyekhez tartozó – szabad térbeli – szög nagyobb, mint 180° .
2. A redukált láthatósági gráf élei a következők:
 - a. Egy sokszöghöz tartozó, szomszédos reflex csúcspontok közötti akadályoldalak.
 - b. Bitangens élek: két reflex csúcspontot összekötő egyenes szakasz, amelynek a meghosszabbításával kapott egyenes minden csúcspontnál érintő helyzetű, vagyis egyik végpontnál sem „döf bele” az akadályba.

Bebizonyítható, hogy a fenti redukcióval csak olyan éleket hagyunk el, amelyek egyébként sem lettek volna részei a legrövidebb útnak. Vagyis a redukált láthatósági gráffal tervezett útvonal minden esetben megegyezik a „hagyományos” láthatósági gráf alapú útvonallal, viszont jelentősen kevesebb számítási kapacitást igényel a megoldás. Ez a 6.3.6 ábrán bemutatott egyszerű példában talán még nem olyan szembetűnő, viszont bonyolultabb akadályelrendezésnél már számosztévő lehet a különbség, mint ahogy az a 6.3.7 ábrán is látható. Itt egy több akadályból álló elrendezést látunk, ahol a kör alakú robot sugarával kibővítettük az akadályokat. Érdemes megfigyelni, hogy a kibővítés hatására sok új csomópont jelent meg, ami a „hagyományos” láthatósági gráfnál rendkívül sok élet eredményezett. Ezzel szemben a redukált gráf jelentősen kisebb méretű, a megoldás mégis ugyanaz.



6.3.7 ábra – Láthatósági gráf alapú tervezés változatai.
a) Láthatósági gráf, b) Redukált láthatósági gráf

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 77. oldal
--	---	--

6.3.2.2 Cella-dekompozíció

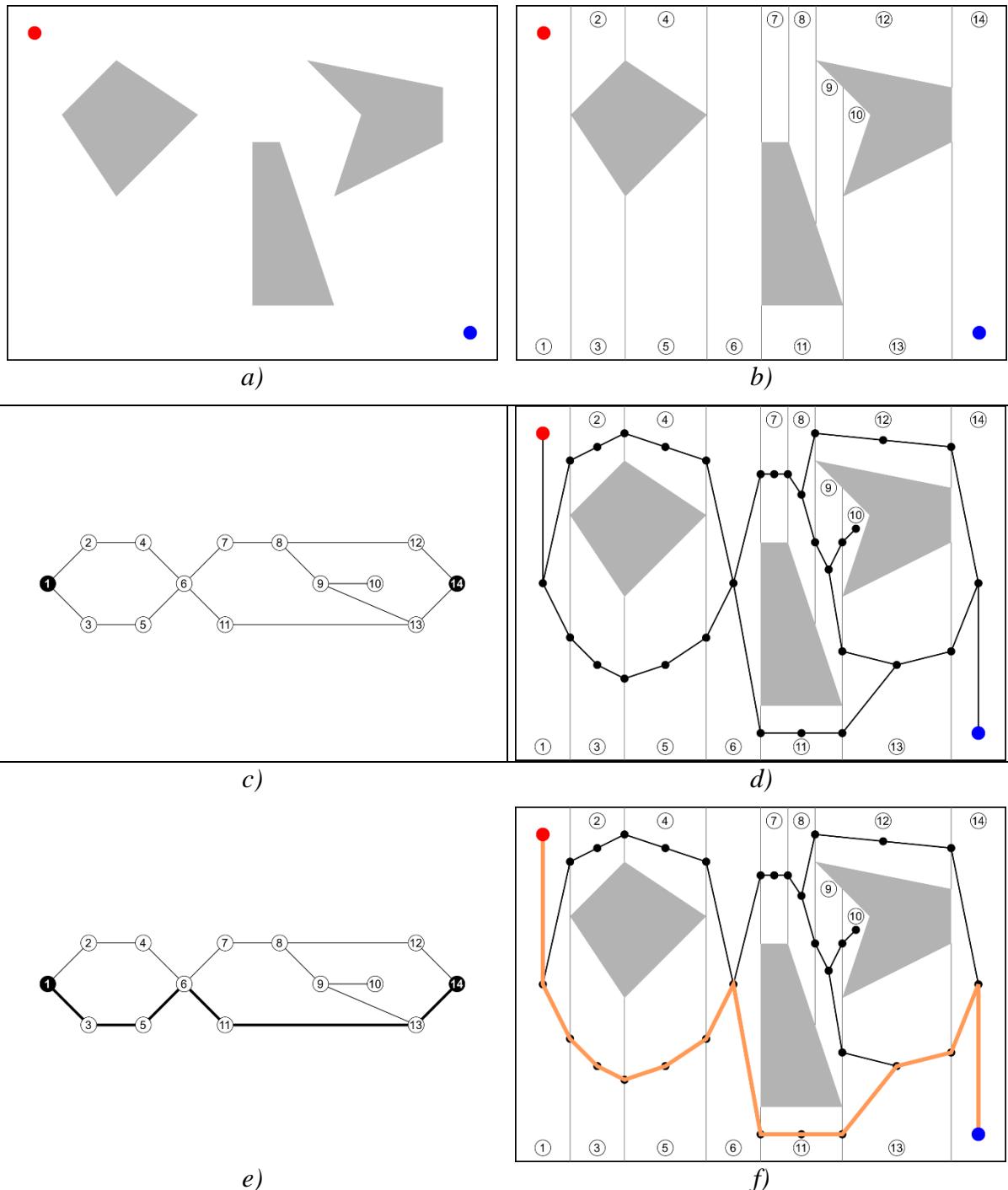
A második bemutatott egzakt pályatervezési módszer a cella-dekompozíció. Itt az útvonal tervezésekor nem az akadályokra, hanem a szabad térrészekre koncentrálunk. A cella-dekompozíció lényege, hogy a konfigurációs teret olyan elemi cellákra osszuk, amelyek mindegyike vagy teljesen foglalt, vagy teljesen szabad térrészt fed le. Ezek után a cellák szomszédossága alapján egy topológiai gráfot építünk, amelynek csomópontjai a szabad cellákat, élei pedig azok szomszédossági viszonyát reprezentálják. Az útvonaltervezéshez a következő lépéseket kell végrehajtanunk:

1. Meg kell állapítanunk, hogy a kezdeti és a célkonfigurációk mely cellákhoz tartoznak. Ezt pl. konvex cellák esetén egyszerű megállapítani.
2. A kezdeti és a célkonfigurációt tartalmazó cellák között keresünk kell egy útvonalat a topológiai gráfban.
3. A topológiai gráfot, ill. az abban megtalált útvonalat geometriailag meg kell feleltetnünk a valóságnak. Ennek egy módja például az, hogy minden cellának a geometriai középpontjából a szomszédos cella középpontjába a közös határszakasz felezőpontján keresztül húzunk egy egyenes darabokból álló útvonalszakaszt. Tehát így a topológiai gráf csomópontjainak a cellaközéppontok, a topológiai gráf éleinek pedig a cellahatárok közepén keresztülhaladó, két egyenes szakaszból álló útvonaldarabok fognak megfelelni.

A cella-dekompozíciós módszereknek több változata létezik, mi itt ismét a kétdimenziós konfigurációs tér esetére szorítkozunk. A 6.3.8 ábra segítségével nyomon követhetjük egy viszonylag egyszerű változat, a *vertikális cella-dekompozíció* lépéseit. Ennél az algoritmusnál a cellákat a következőképpen származtatjuk:

1. Feltételezzük, hogy az akadályok, és a konfigurációs tér határvonala is sokszög alakú. Balról jobbra haladva minden egyes akadály-, ill. határvonal-csúcspontból elindítunk felfelé és lefelé is egy sugarat, amennyiben az adott irány szabad.
2. minden sugarat az első ütközésig, vagy a tartomány határáig húzzuk.
3. Az így kialakult függőleges szakaszok, az akadályok oldalai, valamint a konfigurációs tér határvonalai konvex, trapéz alakú cellákra osztják a szabad teret. Ezek a cellák már megfelelő alapot nyújtanak a topológiai gráf felírásához és az útvonaltervezés fentebb leírt folyamatához.

A cella-dekompozíciós módszerek főként abban különböznek, hogy milyen algoritmust alkalmaznak a cellák létrehozására (pl. igen elterjedt a háromszög-felbontás alkalmazása is). Bizonyos algoritmusok általánosíthatók több dimenzióba, de alapvetően ez az eljárás is leginkább kétdimenziós konfigurációs térben alkalmazható. Előnye, hogy a kiadódó útvonal a szabad térből halad, az akadályok érintése nélkül. Ezzel együtt jár, hogy az útvonal nem lesz optimális (elég csak összehasonlítanunk a 6.3.6.d és a 6.3.8.f ábrákat).



6.3.8 ábra – Vertikális cella-dekompozíció. a) Akadályok, kezdeti és célkonfiguráció, b) A szabad tér cellákra bontása, c) A cellák szomszédosságát reprezentáló topológiai gráf, d) A topológiai gráfnak megfelelő geometriai útvonalhálózat, e) Útvonalkeresés a topológiai gráfban, f) Geometriai útvonal

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 79. oldal
--	---	--

6.3.2.2 Rapidly Exploring Dense Trees

Az előzőekben bemutatott két módszer előnye volt a teljesség, vagyis az a tulajdonság, hogy ha létezik megoldás, akkor azt az algoritmus véges idő alatt meg is találja. Ugyanakkor mindeneként igényelte a konfigurált akadályok pontos modelljét, ami a síkban mozgó pontszerű vagy kör alakú robotok esetén még könnyen kézben tartható, azonban bonyolultabb robotok esetén szinte lehetetlenné teszi az említett módszerek alkalmazását.

Általános konfigurációs terekben az akadályok modellezésének nehézsége miatt szinte kizárolag a mintavételeles pályatervező algoritmusok jöhetsnek szóba. Ezek közös tulajdonsága, hogy az akadályok modellje helyett csupán azt igénylik, hogy álljon rendelkezésre egy ütközésdetektáló algoritmus, amely bármely konfigurációról megmondja, hogy abban ütközne-e a robot vagy nem. A mintavételeles módszerek lényegében a következő lépésekkel állnak:

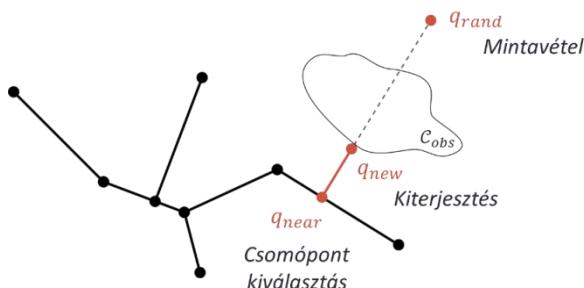
1. Vegyük egy mintát a konfigurációs térből. Az egymás után következő minták kijelölésére alkalmazhatunk determinisztikus vagy véletlen sorozatot is. Olyan sorozatot érdemes használni, amely az általa generált mintákkal egyenletesen lefedi a konfigurációs teret. Ha a kiválasztott konfiguráció nem szabad, ismételjük meg a mintavételt.
2. Próbáljuk meg összekötni a kiválasztott konfigurációt a korábbi lépésekkel rendelkezésre álló gráffal vagy gráf darabokkal, egy vagy több lokális pálya segítségével (ez leggyakrabban lineáris interpolációt, vagyis egyenes szakaszt jelent). A lokális pálya (pl. egyenes szakasz) felvételekor nem kell figyelembe venni az akadályokat.
3. Az összeköttetést biztosító lokális pályát vizsgáljuk meg az ütközésdetektáló algoritmussal: ha egyetlen pontja sem okoz ütközést, akkor felvesszük a gráfba, ellenkező esetben eldobjuk.

A mintavételeles módszereknek nagyon sokféle változata létezik. Ezekből az egyik talán legelterjedtebben használtat mutatjuk itt be, a *Rapidly Exploring Dense Trees* (RDT, gyorsan feltérképező sűrű fák) módszerét. Ennek a módszernek különlegessége, hogy kizárolag fa struktúrájú gráfot (ill. gráfokat) épít, amely gyorsan, hatékonyan és egyenletesen fedi le a szabad teret. A fák előnye, hogy bármelyik csomópontból triviális megtalálni bennük az utat a gyökérhez, így nincs szükség – a sok esetben költséges – útvonalkereső algoritmusokra, amikor a gráfból ki kell nyernünk a konkrét útvonalat.

A Rapidly Exploring Dense Trees módszerét leggyakrabban egyenletes eloszlású véletlen mintasorozattal alkalmazzák. Ebben az esetben a *Rapidly Exploring Random Trees* (RRT) megnevezés használatos. A későbbiekben az RDT (RRT) rövidítést egyaránt használjuk az algoritmus, illetve az algoritmus által generált fák megnevezésére. Az egyszerűség kedvéért a továbbiakban minden az RRT algoritmust emlíjük, de minden állítás igaz az általánosabb RDT algoritmusra is.

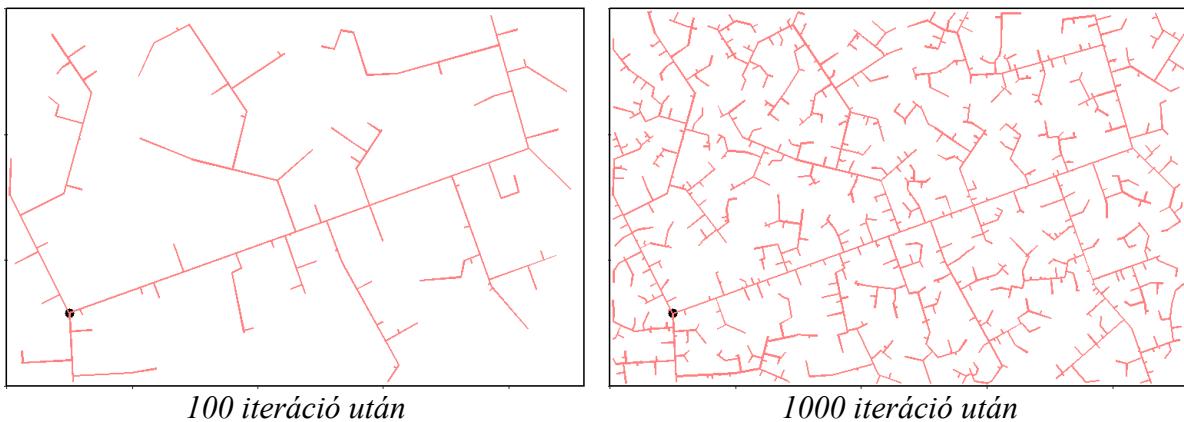
Az RRT algoritmus lépései a következők (6.3.9 ábra):

- Mintavétel.** Kijelölünk egy mintát a konfigurációs térből, amely a gráf kiterjesztés célpontja lesz (q_{rand}).
- Csomópont kiválasztás.** Megkeressük a gráfban q_{rand} -hoz legközelebb eső csomópontot. Létezik az algoritmusnak olyan változata is, amelynél nem csak a csomópontokat, hanem a gráf éleinek belső pontjait is figyelembe vesszük. Ha a legközelebb eső pont egy élen helyezkedik el, akkor az élet a pontnál kettéosztjuk, és a pontot felvesszük a gráf csomópontjai közé. Az így kiválasztott legközelebbi gráfbeli konfiguráció (q_{near}) lesz a gráf kiterjesztés kiindulópontja.
- Kiterjesztés.** q_{near} és q_{rand} konfigurációkat egy lokális pályával (pl. egyenessel) összekötjük. Amennyiben ez ütközésmentes, a lokális pályát élként, q_{rand} pontot pedig csomópontként hozzáadjuk a gráphoz. Ha a lokális pálya akadályon halad keresztül, akkor is kiterjesztjük a gráfot, de q_{rand} helyett csak az utolsó ütközésmentes pontig a lokális pálya mentén (q_{new}).



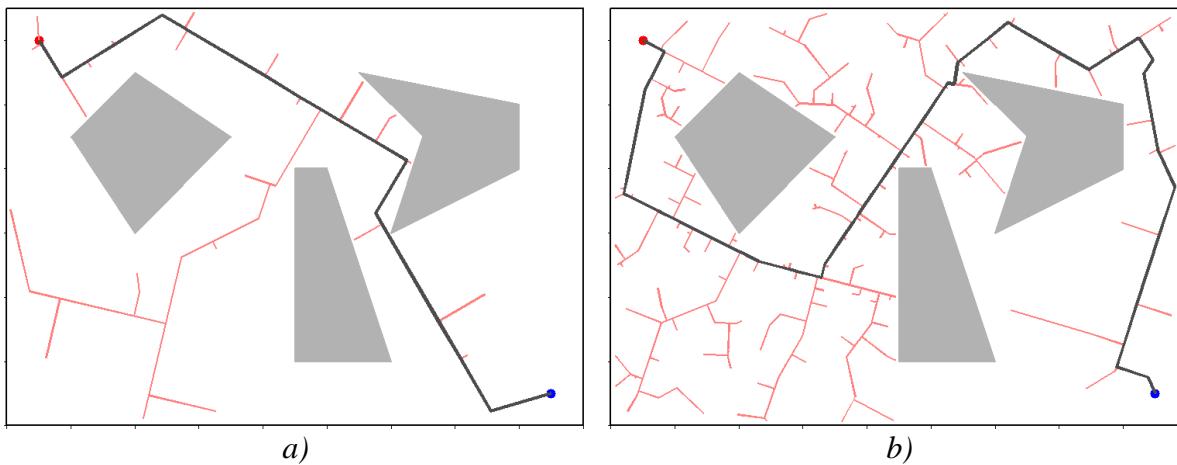
6.3.9 ábra – Az RRT algoritmus lépései

A fenti algoritmus eredményeként létrejövő fát ábrázolja a 6.3.10 ábra, amely akadályok nélküli esetben illusztrálja a tér gyors, egyenletes és a mintaszám növekedésével egyre sűrűbb lefedését.



6.3.10 – Az RRT szabad tér kitöltő tulajdonságának illusztrációja

Az RRT-t például úgy lehet pályatervezésre használni, hogy a kezdeti konfigurációból kiindulva növesztjük a fát. Mivel a fa a mintaszám növelésével egyre sűrűbben fedi le a szabad teret, ezért megfelelő számú iteráció után tetszőlegesen közel tud kerülni a célkonfigurációhoz. A feladatot megoldottnak tekinthetjük, ha a fa és a cél távolsága egy előre adott határon belülre kerül. Érzékelhető, hogy ez a nagyon egyszerű elv természetéből fakadóan csak közelítő megoldást adhat (bár a mintaszám növelésével egyre pontosabban). Emiatt úgy is módosíthatjuk a fa növeztését, hogy a teljesen egyenletes eloszlású mintasorozatot kis mértékben elhajlítjuk: néhány százalékos valószínűsséggel a célkonfigurációt választjuk a véletlen minták helyett (sampling bias). Így, ha egy iterációban a cél kerül sorra mintaként, és a fa kiterjesztése ütközésmentesen sikerül, akkor pontos megoldást kapunk. Ezt a megközelítést láthatjuk a 6.3.11 ábrán, 5% sampling bias alkalmazása mellett. Az ábrán az is látható, hogy az RRT valószínűségi jellege miatt ugyanazt a tervezési feladatot különböző futtatások esetén akár lényegesen eltérő lépésszámmal oldja meg. A lépésszám csökkentésére szokták azt a megoldást alkalmazni, hogy nem csak egy, hanem két fát növesztenek: egyet a kezdeti, és egyet a célkonfigurációból kiindulva. Ebben az esetben azt kell felismerni, amikor a két fa összeér, és a metszésponttól kezdve már könnyedén felírható a megoldást jelentő útvonal.



6.3.11 – Az RRT alkalmazása a pályatervezésben.
Megoldás a) 63 iteráció, b) 385 iteráció után

Az RDT (RRT) módszer hátránya – ami minden más mintavételeles módszerre is igaz –, hogy nem tudja garantálni a megoldás megtalálását véges idő alatt. Hozzá kell tenni ugyanakkor, hogy mivel a mintaszám növelésével a fa egyenletesen sűrűn fedi le a szabad teret, ezért a megoldás valószínűsége egyhez tart. Mindemellett a tapasztalat azt mutatja, hogy az RRT módszer általában igen gyorsan megoldáshoz vezet, emiatt egyszerűbb akadályelrendezés esetén futásidő szempontjából akár a következő részben bemutatott reaktív akadályelkerülő algoritmusok vetélytársa is lehet. További nagy előnye, hogy a bemutatott gráf építési algoritmus dimenziószámtól és akadálytípustól függetlenül bármilyen konfigurációs térben alkalmazható, ezért igen szerteágazó robottípusoknál és felhasználási területeken találkozhatunk vele.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 82. oldal
--	---	--

6.3.4 Akadályelkerülési módszerek

A globális pályatervezés után lássunk néhány példát a lokális (reaktív) akadályelkerülésre is. Itt az akadályok és a szabad tér globális ismeretén alapuló tervezés helyett a korlátozott érzékelési tartományú szenzorok mérései alapján származtatott pillanatnyi beavatkozásban van a hangsúly. Az akadályelkerülési módszerek alapvető működése a következőkben foglalható össze:

- Adott egy érzékelési és beavatkozási periódusidő, ennyi időnként fut le az algoritmus.
- A periódusidőt úgy választjuk meg, hogy egy periódus alatt a robot mozgása mindenkorábban kisebb legyen, mint a szenzorok érzékelési távolsága.
- minden egyes periódusban a célunk egy olyan beavatkozó jel generálása, amely egrészt nem okoz ütközést, másrészt közelebb viszi a robotot a célhöz.

A továbbiakban három ilyen akadályelkerülési módszert mutatunk be.

6.3.4.1 Virtuális erőtér

A virtuális erőtér (Virtual Force Field – VFF) módszer [3] alapötlete, hogy a robot mozgásának meghatározására virtuális erőket vezet be úgy, hogy az akadályok tasztíto, a célpont pedig vonzó hatást fejenek ki a robotra. A robotra ható virtuális erők eredője határozza meg a mindenkorábban haladási irányt és sebességet.

A módszer eredeti változatában az akadályokat egy ún. hisztogram rács reprezentálja. Ez tulajdonképpen egy lokális térkép, amely négyzet alakú cellákból és az azokhoz rendelt foglaltság valószínűségi értékekből áll. A hisztogram rácsban minden (i, j) indexű cellához egy $c_{i,j}$ foglaltsági érték tartozik. Minél nagyobb ez az érték, annál valószínűbb az adott rácsponthoz tartozó cella foglaltsága. A hisztogram rács foglaltsági értékeit menet közben folyamatosan frissülnek az aktuális szenzoradatok (pl. ultrahangos szenzorok távolságadatái) alapján.

Az algoritmus a hisztogram rácsnak mindenkorábban csak egy kisebb, a robot körül, $w_s \times w_s$ méretű részét, az ún. *aktív ablakot* veszi figyelembe. Az aktív ablakot \mathbf{C}^* , cellaértékeit $c_{i,j}^*$ jelöli. minden aktív ablakbeli foglalt cella a következő virtuális tasztító erőt fejt ki a robotra:

$$\mathbf{F}_{i,j} = \frac{F_{cr} c_{i,j}^*}{d_{i,j}^2} \left[\frac{x_i - x_0}{d_{i,j}} \hat{x} + \frac{y_j - y_0}{d_{i,j}} \hat{y} \right], \quad (6.3.1)$$

ahol

- | | |
|---------------|--|
| F_{cr} : | Taszító erő konstans |
| $d_{i,j}$: | Az (i, j) aktív cella távolsága a robottól |
| $c_{i,j}^*$: | Az (i, j) aktív cella foglaltsági értéke |
| x_0, y_0 : | A robot aktuális koordinátái |
| x_i, y_j : | Az (i, j) aktív cella koordinátái |

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 83. oldal
--	---	--

\hat{x}, \hat{y} : x, y irányú egységvektorok

Figyeljük meg, hogy az erőkomponens nagysága egyenesen arányos a cella foglaltsági értékével, és fordítottan arányos a robottól való távolság négyzetével. Ezzel a robot közvetlen közelében lévő foglalt cellák taszító hatása felerősödik.

Az eredő taszító erő a fenti erőkomponensek vektori összege:

$$\mathbf{F}_r = \sum_{(i,j) \in C^*} \mathbf{F}_{i,j} \quad (6.3.2)$$

Mindeközben egy konstans nagyságú vonzó virtuális erő „húzza” a robotot a célpont irányába:

$$\mathbf{F}_t = F_{ct} \left[\frac{x_t - x_0}{d_t} \hat{x} + \frac{y_t - y_0}{d_t} \hat{y} \right], \quad (6.3.3)$$

ahol

- F_{ct} : Vonzó erő konstans
- d_t : A célpont távolsága a robottól
- x_t, y_t : A célpont koordinátái

Amint az a

6.3.12 ábrán is látható, a taszító és vonzó erők vektori összege adja az eredő \mathbf{R} virtuális erőt:

$$\mathbf{R} = \mathbf{F}_t + \mathbf{F}_r \quad (6.3.4)$$

Az \mathbf{R} vektor hossza adja meg a haladási sebességet, irányzöge (δ) pedig szolgáltatja az alapjelet a robot szögsebesség-szabályozásához:

$$\Omega = k[\delta - \Theta], \quad (6.3.5)$$

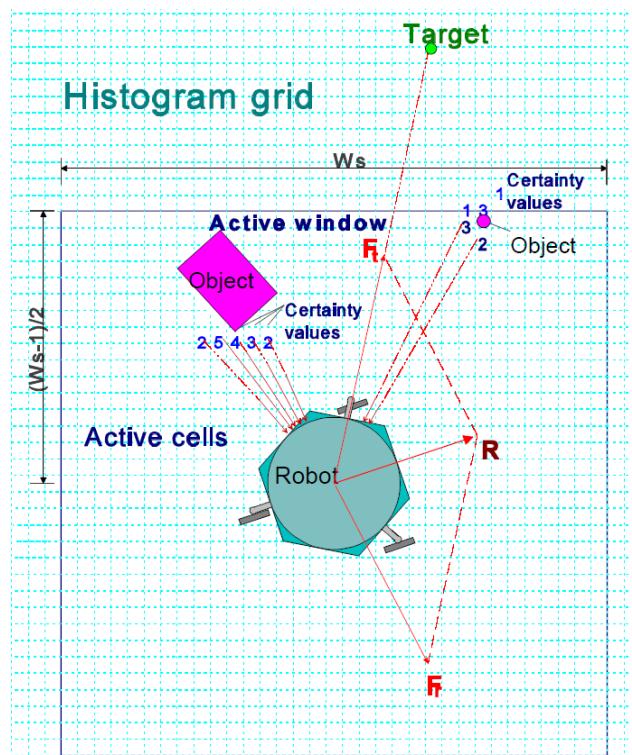
ahol Ω a szögsebesség-beavatkozójel, Θ a robot aktuális haladási iránya, k pedig egy s^{-1} dimenziójú arányossági tényező.

Bár a tapasztalat azt mutatja, hogy a virtuális erőtérről módszerével gyors és biztonságos mozgás érhető el, jó néhány hátrányos tulajdonságára is fény derült. Ezek a problémák a potenciálmező módszerek alaptulajdonságaiiból származnak, és függetlenek a konkrét implementációtól:

Lokális minimum csapdák: A legkézenfekvőbb hibalehetőség, amikor a robot „csapdába fut”, azaz a cél felé vezető út lokálisan le van zárva (pl. U alakú akadályelrendezésnél). Az ilyen csapdahelyzetek különböző heurisztikákkal vagy globális pályatervezés bevonásával oldhatók fel.

Közeli akadályok közötti áthaladás meghiúsulása: Előfordulhat, hogy bár az út nincs lezártva a célpont felé, az egymáshoz közeli akadályok (pl. egy ajtó két oldala) miatt létrejövő taszító erő éppen ellentétes irányba mutat, mint a célpont vonzó ereje, és „legyőzi” azt, lehetetlenné téve az áthaladást.

Oszcilláció keskeny folyosók esetén: Keskeny folyosón keresztülhaladva, a robot mozgása csak akkor stabil, ha a folyosó középvonalában halad. Amint kissé kitér az egyik fal irányába, a fal közeledése miatt egy erős oldalirányú tasztíterő lép fel, amely mintegy „átlöki” a robotot a középvonal másik oldalára, ahol ugyanez történik, ellenkező előjellel. Bizonyos körülmények között ez instabil, oszcilláló mozgáshoz vezethet.



6.3.12 ábra – A virtuális erőtér (VFF) módszer működési elve [3]

6.3.4.2 Vektormező hisztogram

A virtuális erőtér (VFF) módszer [4] nyilvánvaló hátrányai összönözték egy továbbfejlesztett algoritmus, a vektormező-hiszrogram (VFH – Vector Field Histogram) módszerének kifejlesztését. Az a felismerés vezetett ide, hogy a hibákat nagymértékben az irányítási parancs meghatározásánál történő extrém mértékű adatredukció okozza. Ugyanis a VFF módszer több száz cella-adatot sűrít össze egy lépésben minden két kimenő adattá (ezek az F_r vektor nagysága és iránya). Következésképpen, az akadálypontok lokális eloszlásának információja elvész.

A vektormező-hiszrogram módszer ezzel ellentétben kétlépcsős adatredukciót alkalmaz. A környezet leírására ugyanazt a hiszrogram rácsot használja, mint a VFF, és ugyanúgy definiálja az aktív ablakot is. Az adatredukció két lépése:

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 85. oldal
--	---	--

- Első lépésben a kétdimenziós hisztogram rács \mathbf{C}^* aktív ablakának foglaltsági információját egy egydimenziós \mathbf{H} poláris hisztogrammá (*polar histogram*) transzformálja, amely a robot körüli *poláris akadályszűréséget* hivatott reprezentálni.
- Második lépésben a poláris hisztogram alapján történik a kívánt haladási irány és sebesség meghatározása.

Első lépés: a poláris hisztogram

Az adatredukció első lépése tehát a \mathbf{C}^* aktív ablak transzformációja a \mathbf{H} poláris hisztogrammá. Az aktív cellák tartalmát ún. *akadályvektorokként* értelmezzük, amelyeknek a robot referencia pontjához képesti szögét és nagyságát a következőképpen definiáljuk:

$$\beta_{i,j} = \arctan \frac{y_j - y_0}{x_i - x_0}, \quad (6.3.6)$$

$$m_{i,j} = (c_{i,j}^*)^2 (a - b d_{i,j}), \quad (6.3.7)$$

ahol

- | | |
|-----------------|--|
| $\beta_{i,j}$: | Az (i, j) aktív cella robothoz képesti iránya (az akadályvektor szöge) |
| $m_{i,j}$: | Az (i, j) aktív cellához tartozó akadályvektor nagysága |
| $d_{i,j}$: | Az (i, j) aktív cella távolsága a robottól |
| $c_{i,j}^*$: | Az (i, j) aktív cella foglaltsági értéke |
| x_0, y_0 : | A robot aktuális koordinátái |
| x_i, y_j : | Az (i, j) aktív cella koordinátái |
| a, b : | Pozitív konstansok |

Az a, b konstansokat úgy választjuk meg, hogy $a - b d_{max} = 0$ teljesüljön, ahol $d_{max} = \frac{\sqrt{2}(w_s - 1)}{2}$ az aktív ablak legtávolabbi cellájának távolságát jelöli. Ezáltal $m_{i,j} = 0$ érvényes a legtávolabbi cellára, és $m_{i,j}$ értéke ($c_{i,j}^*$ -től eltekintve) lineárisan növekszik a közelebbi cellák felé haladva.

A \mathbf{H} poláris hisztogram szögfelbontását α jelöli, amelynek értékét úgy kell megválasztani, hogy $n = 360^\circ / \alpha$ egész szám legyen (pl. $\alpha = 5^\circ$ és $n = 72$). A poláris hisztogram minden egyes szektora egy diszkrét $\rho = k\alpha$ szögnek felel meg, ahol $k = 0, 1, 2, \dots, n-1$. Az (i, j) cellák hozzárendelése az egyes szektorokhoz a következőképpen történik:

$$k = \text{INT}\left(\frac{\beta_{i,j}}{\alpha}\right), \quad (6.3.8)$$

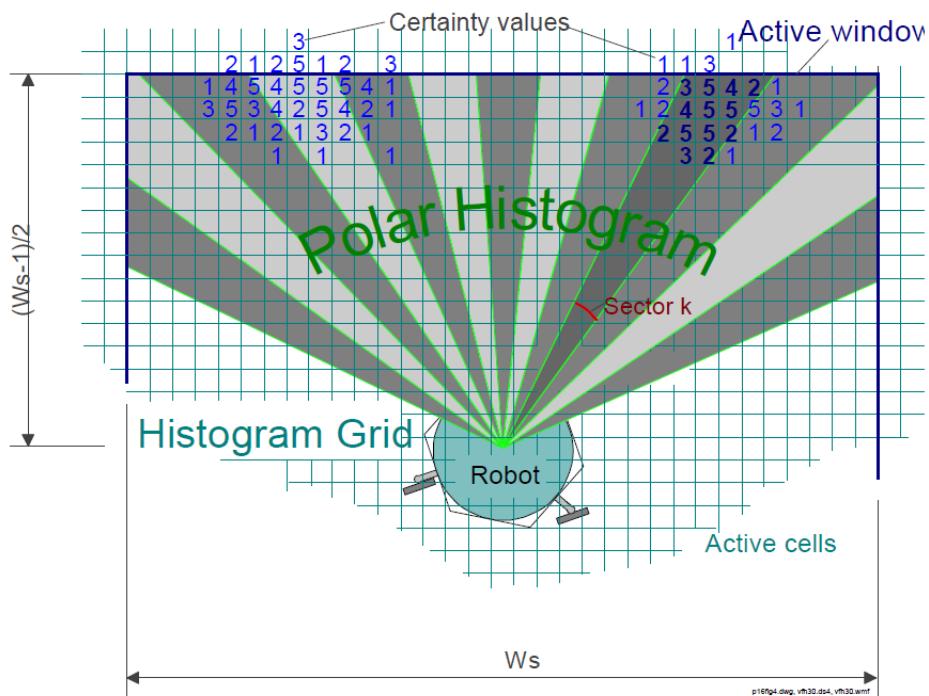
ahol $\text{INT}(\cdot)$ az egészre kerekítés operátora.

Az egyes k szektorokhoz tartozó h_k poláris akadályszűréseket (*POD – polar obstacle density*) az adott szektorhoz rendelt $m_{i,j}$ értékek összegeként definiáljuk:

$$h_k = \sum_{i,j} m_{i,j} \quad (6.3.9)$$

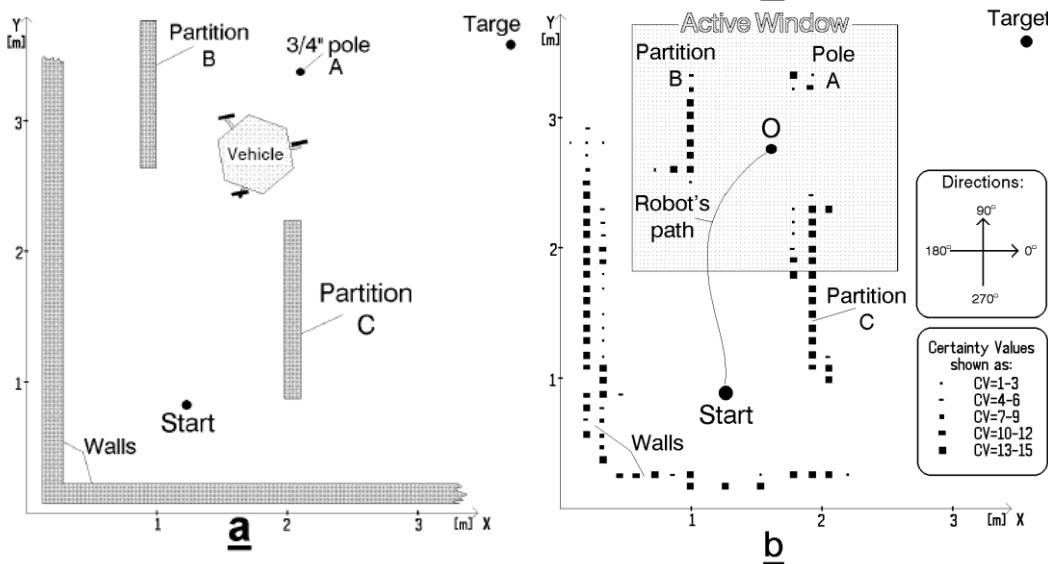
A \mathbf{C}^* -ból \mathbf{H} -ba történő transzformációt szemlélteti a 6.3.13 ábra. A jobb áttekinthetőség érdekében az ábrán a poláris hisztogram felbontása $\alpha = 10^\circ$. A hisztogram rács diszkrét voltából adódó „csipkézettség” elkerülése érdekében érdemes még egy simítóátlagolást is elvégezni a h_k értékeken:

$$h'_k = \frac{h_{k-l} + 2h_{k-l+1} + \dots + lh_k + \dots + 2h_{k+l-1} + h_{k+l}}{2l+1}, \text{ pl. } l=5 \quad (6.3.10)$$

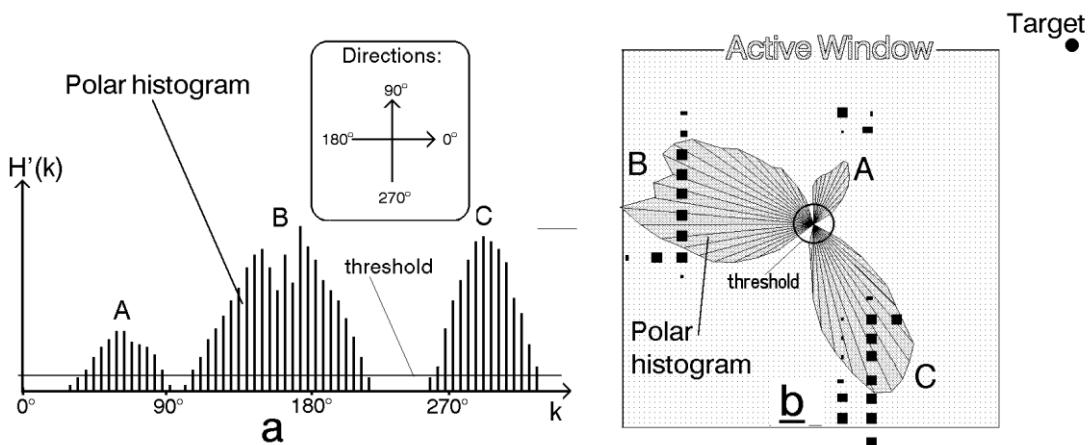


6.3.13 ábra – Aktív cellák hozzárendelése a poláris hisztogramhoz [4]

A transzformációs eljárás illusztrálására tekintsük a 6.3.14 és 6.3.15 ábrákat. A 6.3.14.a ábrán egy lehetséges akadály-elrendezést, illetve egy mozgó robotot láthatunk. Az 6.3.14.b ábra a mozgás közben felépített hisztogram rácsot mutatja. Az O pontban felvett (simított) poláris hisztogramot láthatjuk az 6.3.15.a ábrán. A poláris hisztogramban megfigyelhető A, B és C csúcsok az 6.3.14 ábrán A, B, C-vel jelölt akadályoknak felelnek meg. Az 6.3.15.b ábrán ugyanez a poláris hisztogram szerepel, a robot köré vetítve.



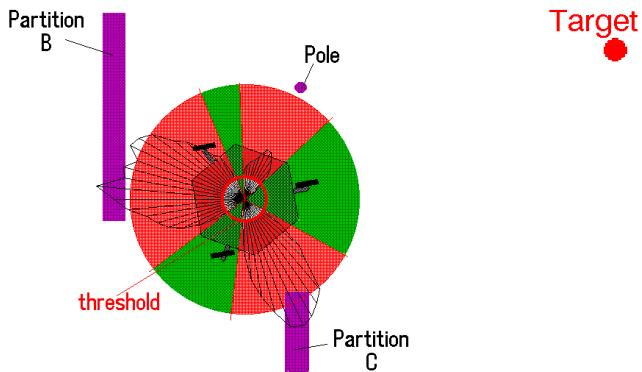
6.3.14 ábra – (a) Egy lehetséges akadály-elrendezés és (b) a hozzá tartozó hisztogram rács reprezentáció [4]



6.3.15 ábra – (a) Az O pontban felvett poláris hisztogram, (b) a poláris hisztogram a robot köré vetítve [4]

Második lépés: haladási irány és sebesség

Az adatrédukció második lépései között a poláris hisztogram alapján származtatjuk a haladás kívánt θ irányát és V sebességét. Amint az a 6.3.15 ábrán is látható, a simított poláris hisztogram tipikusan „csúcsokból” (nagy POD értékű szektorcsoportok) és „völgyekből” (kis POD értékű szektorcsoportok) áll. A „csúcsok” és „völgyek” határait egyszerű küszöbözéssel húzhatjuk meg. A küszöb alá eső POD értékű szektorcsoportokat „esélyes völgyeknek” (*candidate valleys*) nevezzük, és ezek körében folytatjuk a haladási irány keresését (6.3.16 ábra).



6.3.16 ábra – A küszöbözés után kiadódó lehetséges haladási irányok
(az „esélyes völgyek” zölddel jelölve) [4]

Általában több „esélyes völgy” keletkezik a küszöbözés hatására, ezek közül azt választja ki a VFH algoritmus, amely a legközelebb esik a célpont irányába mutató k_{targ} szektorhoz. A megfelelő „völgy” megtalálása után azon belül még ki kell választanunk azt a szektort, amelynek irányába a robotnak haladnia kell. Ha a k_{targ} egy „völgyhöz” tartozik, nincs szükség további keresésre. Ellenkező esetben a következő történik. Az algoritmus megkülönböztet „széles” és „keskeny völgyeket”. Azokat a „völgyeket” tekinti „szélesnek”, amelyek legalább s_{max} darab szektorból állnak. Mindkét féle „völgynek” van egy *közeli* (k_n) és egy *távoli* (k_f) határszektora. A közeli határszektor minden esetben a k_{targ} szektorhoz legközelebb eső szektor. A távoli határszektor „keskeny völgy” esetén a völgy másik határa, „széles vögynél” azonban a $k_f = k_n + s_{max}$ képlet szerint adódik. Az előírt haladási irány:

$$\theta = \frac{k_n + k_f}{2} \alpha \quad (6.3.11)$$

A „széles völgy” esete a 6.3.17 ábrán, a „keskeny völgy” esete a 6.3.18 ábrán figyelhető meg. A haladási irány ilyen módon történő kiválasztása lehetővé teszi a – VFF módszernél problematikus – közeli akadályok közötti biztonságos áthaladást, mivel a robot minden a „völgy” közepe irányába fog haladni. Kevés akadály esetén s_{max} megválasztásával lehet beállítani, hogy a robot mennyire közelítse meg az akadályokat (6.3.17 ábra). A módszer további előnye, hogy a poláris hisztogram simítása miatt az irányvezérlő parancs nem változik ugrásszerűen, minimálisra csökkentve ezzel az oszcillációk kialakulásának esélyét.

A haladási sebesség megválasztása a következőképpen történik:

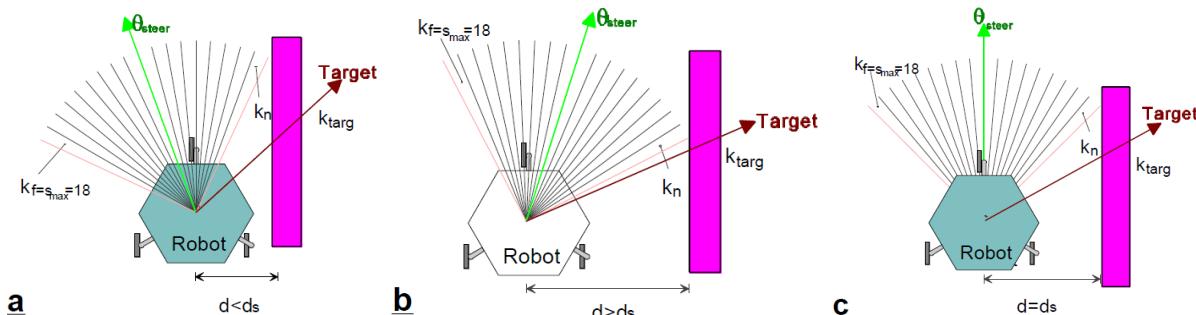
$$V' = V_{\max} \left(1 - \frac{\min(h'_c, h_m)}{h_m} \right), \quad (6.3.12)$$

ahol V_{\max} a robot maximális sebessége, h'_c az aktuális haladási irányba eső szektor poláris akadálysűrűség-értéke, h_m pedig egy empirikus úton megválasztott konstans. A fenti összefüggés minden esetben biztosítja, hogy V' ne legyen negatív. A kanyarodás

megkönnyítése érdekében további sebességsökkentést írunk elő az aktuális szögsebesség figyelembe vételevel:

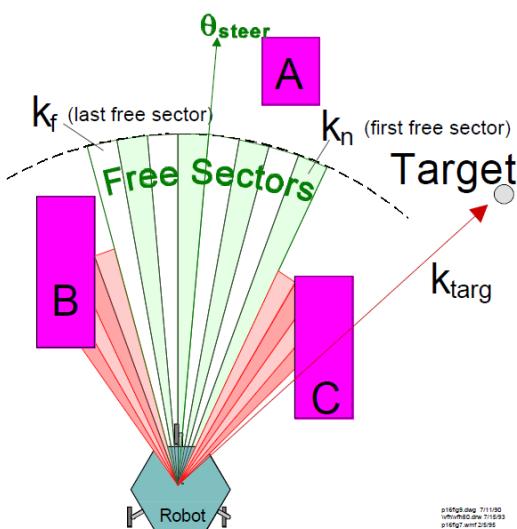
$$V = V' \left(1 - \frac{\Omega}{\Omega_{max}} \right) + V_{min} , \quad (6.3.13)$$

ahol Ω_{max} a robot maximális szögsebessége, Ω az aktuális szögsebesség. Azt, hogy az akadályok és a kanyarodás miatti sebességsökkentés ne okozza a robot megállását, egy $V_{min} > 0$ alsó sebességlimitált biztosítja.



6.3.17. ábra – A „széles völgy” esete, elhaladás egy akadály mellett

- (a) Távolodó mozgás, ha az akadály túl közel van
- (b) Közeledő mozgás, ha az akadály távol van
- (c) Megfelelő távolság esetén a robot az akadálytalál párhuszamosan mozog [4]



6.3.18. ábra – A „keskeny völgy” esete, elhaladás közötti akadályok között [4]

A fentiek alapján elmondható, hogy a VFH módszer a VFF-nél kifinomultabb módon interpretálja a környezeti információkat, aminek köszönhetően sikeresen kezeli a VFH módszer számára kritikus szituációkat is. Tekintve, hogy természeténél fogva ez is egy lokális mozgástervező algoritmus (a mozgatóparancs származtatásakor csak a robot lokális

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 90. oldal
--	---	--

környezete játszik szerepet), a lokális minimumokba való „beragadás” veszélye itt is fennáll. Ebben az esetben itt is különböző heurisztikákra, vagy egy fölérendelt globális pályatervezőre van szükség.

6.3.4.3 Dinamikus ablak módszer

Az előzőleg tárgyalt két akadályelkerülési módszer számos előnyös tulajdonsággal rendelkezik, azonban van egy komoly hiányosságuk. Mindkét módszer elsődleges célkitűzése, hogy valamilyen módon meghatározzon egy biztonságos haladási irányt. Az irány ismeretében azután egy különálló lépéssben generálódik az a mozgatóparancs, amely a robotot ténylegesen az előírt irányba hivatott mozgatni. Ez a megközelítés nem számol a robot dinamikus tulajdonságaival, így a robot tehetetlensége miatt a meghatározott irányú mozgás csak közelítőleg tud megvalósulni. Másképpen fogalmazva, elméletileg csak akkor lenne lehetséges az előírt irányú mozgás pontos megvalósítása, ha a beavatkozó szervek (meghajtó motorok) akár végtelen nyomatéket tudnának kifejteni a robotra. A pontatlan mozgásmegvalósítás bizonyos körülmények között a mozgás biztonságát veszélyeztetheti.

Az említett problémára ad megoldást a *dinamikus ablak módszer* (*DWA – Dynamic Window Approach*), amely a 90-es évek második felében jelent meg, és azóta több módosítása, kiegészítése is napvilágot látott [5]. A módszer sikereségét frappáns alapötletének köszönheti, miszerint nem választja külön a kívánt irány és az azt megvalósító mozgatóparancs meghatározását, hanem közvetlenül a sebességek és szögsebességek terében fejezi ki az irányítási célt. Mindeközben figyelembe veszi a robot dinamikus korlátait is (korlátozott gyorsulások formájában). Az eredeti módszert nem holonomikus, szinkron hajtású robotokra dolgozták ki, amely módosítás nélkül alkalmazható differenciális hajtású platformokra is, tekintve, hogy a két konfiguráció mobilitási korlátozása megegyezik. E korlátozás abban nyilvánul meg, hogy az aktuális sebesség mindenkorban az orientáció irányába mutat, ami jól látszik a két platform mozgásellenetéből:

$$\begin{aligned}\dot{x}(t) &= \cos(\varphi(t)) \cdot v(t) \\ \dot{y}(t) &= \sin(\varphi(t)) \cdot v(t) \\ \dot{\varphi}(t) &= \omega(t)\end{aligned}\tag{6.3.14}$$

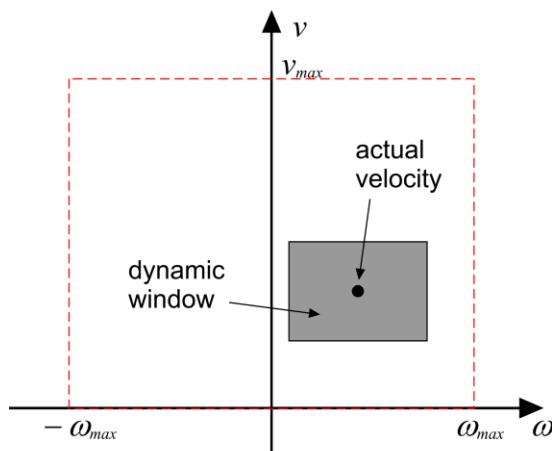
ahol (x, y, φ) a robot pozíciója és orientációja, (v, ω) a sebessége és szögsebessége⁵. Később megjelentek olyan módosítások is, amelyek holonomikus robotokra is kiterjesztették a dinamikus ablak módszert.

Az eredeti megközelítés megalkotói megmutatták, hogy egy, a fenti kinematikai mozgásellenettel rendelkező robot pályája körív-darabok sorozatával közelíthető. Egy ilyen körív-darabot egyértelműen meghatároz a robot pozíciója és orientációja, valamint sebessége és szögsebessége. Könnyen belátható, hogy a robot adott (x, y, φ) helyzetéből induló, (v, ω) sebességű mozgás egy $|v/\omega|$ sugarú, érintőirányú, körív alakú trajektóriát eredményez. A DWA algoritmus célja: minden k időpillanatban meghatározni a következő elemi körív-

⁵ A későbbiekben a kételemű (v, ω) vektort egyszerűen „sebesség”-ként fogjuk említeni.

darabot, illetve a hozzá tartozó (v_k, ω_k) sebességet úgy, hogy a robot az akadályokat elkerülve a cél felé haladjon.

A DWA módszer a robot dinamikus korlátainak figyelembe vételére bevezeti a *dinamikus ablak* fogalmát, amelyet a 6.3.19 ábra szemléltet. A dinamikus ablak azoknak a sebességeknek a halmaza, amelyek az aktuális sebességből a következő időszelet végére elérhetők. A dinamikus ablak méretét tehát a gyorsuláskorlátok és az algoritmus ciklusideje határozza meg. Ha tehát betartjuk azt a feltételt, hogy minden esetben a dinamikus ablakból választunk sebességet, biztosan megvalósítható mozgást kapunk.



6.3.19. ábra – A dinamikus ablak fogalma

Az ütközésmentes haladás biztosítása érdekében a sebességtér további szűkítésére van szükség. Az algoritmus a dinamikus ablakon belül megkülönböztet *megengedhető* és *nem megengedhető* sebességeket. Azokat a sebességeket nevezük *megengedhetőnek* (*admissible velocities*), amelyeket a következő időszeletben alkalmazva a robot nem ütközik, illetve szükség esetén biztonságosan megállítható (a gyorsuláskorlátok figyelembevételével), mielőtt elérné a legközelebbi akadályt az adott körív mentén.

Az algoritmus a dinamikus ablakon belüli megengedhető sebességek közül végül egy optimalizálási eljárás segítségével választja ki a következő időszeletben alkalmazandó sebességet.

A DWA algoritmus tehát a következő lépésekkel áll:

Keresési tér szűkítése:

- Körív alakú trajektória-elemek:** Kizárolag körívek sorozatából álló trajektoriákat veszünk figyelembe. A trajektória-elemeket egyértelműen meghatározzák a hozzájuk tartozó (v, ω) sebességek. Ezáltal egy kétdimenziós sebességtérben keressük az irányítási parancsot.

- b) **Dinamikus ablak:** Csak azokat a sebességeket vesszük figyelembe, amelyek elérhetők a következő időszeletben.
- c) **Megengedhető sebességek:** Csak azokat a sebességeket vesszük figyelembe, amelyek lehetővé teszik az ütközésmentes haladást (és szükség esetén megállást).

Optimalizálás: A fennmaradó sebességek közül azt választjuk ki, amelyik maximalizálja a következő célfüggvényt:

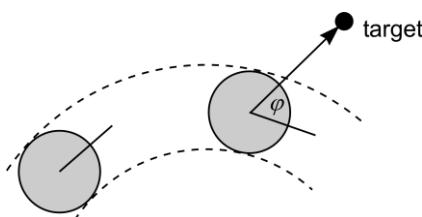
$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega)) \quad (6.3.15)$$

A függvény a következő három dolgot jutalmazza (az aktuális robotpozíció és orientáció figyelembevételével):

- d) **Haladás a célpont irányába:** A *heading* (v, ω) függvény a célpont felé haladás mértékét adja meg. Értéke akkor a legnagyobb, ha a robot pontosan a cél felé mozog. Meghatározása a következők szerint történik:

$$\text{heading}(v, \omega) = 180^\circ - \varphi(v, \omega) \quad (6.3.16)$$

ahol $\varphi(v, \omega)$ a robot jövőbeli orientációhibája a (v, ω) sebesség alkalmazása esetén a következő időszelet végén (6.3.20 ábra).



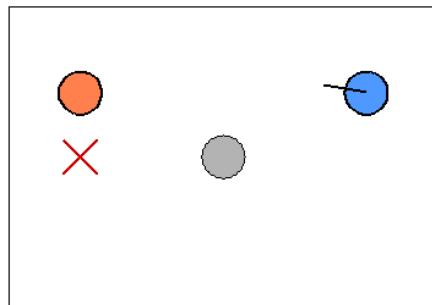
6.3.20. ábra – A *heading* függvény meghatározása

- e) **Távolság:** A *dist* (v, ω) függvény a (v, ω) sebességhez tartozó köríven lévő legközelebbi akadály körív menti távolságát adja vissza. Azokat a pályaelemeket részesíti előnyben, amelyeknél ez a távolság minél nagyobb.
- f) **Sebesség:** A *vel* (v, ω) függvény egyszerűen a v haladási sebességet adja vissza, tehát a minél gyorsabb haladást jutalmazza.

α, β, γ az egyes komponensek súlyozó tényezői, $\sigma(\cdot)$ pedig egy simító függvény.

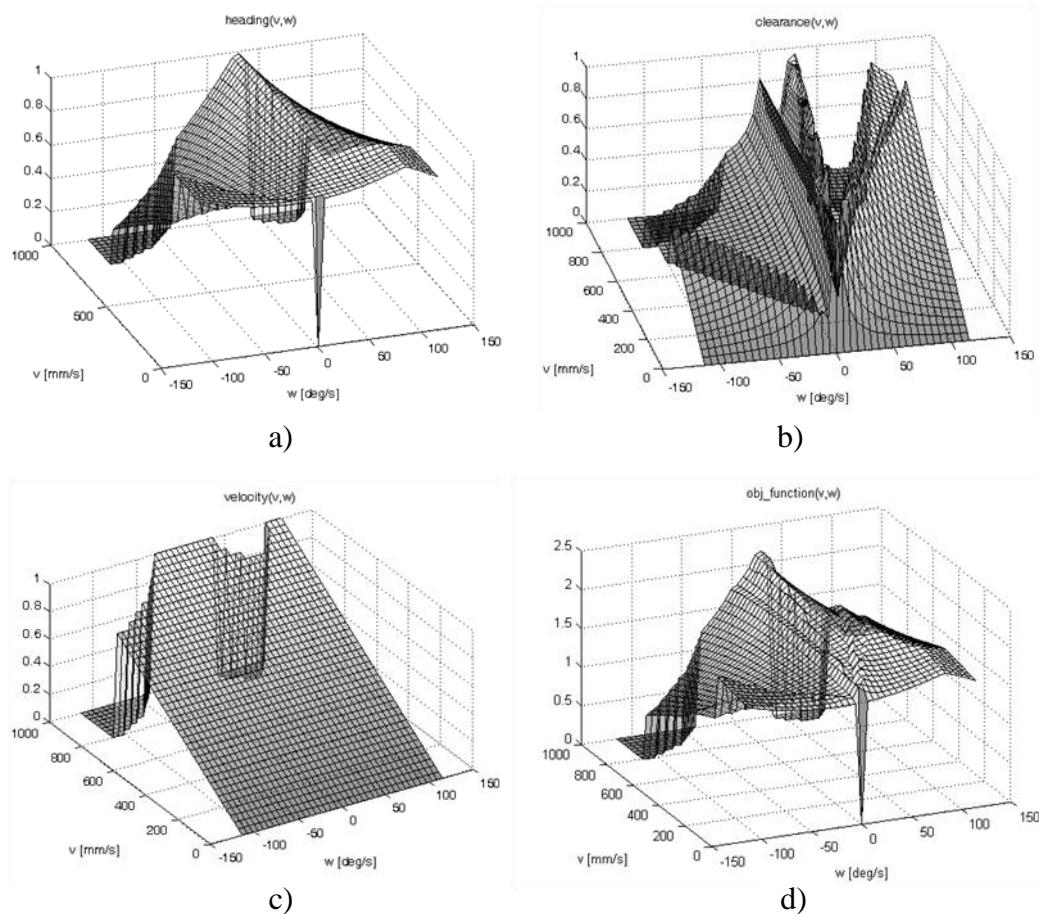
A kiválasztott (v, ω) sebességet alkalmazzuk a következő időszeletben, majd az egész kezdődik előlről.

A szemléletesség kedvéért tekintsünk egy-egy példát az egyes célfüggvény-komponensekre. Az 6.3.21. ábra egy egyszerű, szimulált szituációt mutat be. A robotot a jobb oldali kék kör jelzi, a másik két kör két hengeres akadály. A célpontot a bal oldali piros „X” jel mutatja.



6.3.21. ábra – Egy egyszerű akadályelrendezés

A 6.3.22. ábra az egyes célfüggvény-komponenseket és magát az összesített célfüggvényt ábrázolja az 6.3.21. ábrán bemutatott szituációban. A szemléletesség kedvéért a függvényeket az egész sebességtérre kiértékeltük, nem csak az aktuális dinamikus ablakra. Az egyes részábrákon jól kivehető a nagy haladási sebességek sávjában lévő két mély „szakadék”. Ezek nem megengedett sebességeknek felelnek meg, amelyek a kizárást miatt nulla értékkel szerepelnek a célfüggvényben.



6.3.22. ábra – Példa az optimalizálási célfüggvényre:
(a) heading, (b) dist, (c) vel, (d) az összesített célfüggvény

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 94. oldal
--	---	--

A DWA módszerrel gyors és biztonságos haladás érhető el, viszonylag sima pálya mentén, ha nem túl bonyolult az akadályok elrendezése.

Összegezve elmondható, hogy a DWA módszer egy egyszerű és elegáns megoldást kínál az akadályelkerülés problémájára. Ugyanakkor néhány hátrányos tulajdonságáról sem szabad megfeledkezni:

Lokális minimum csapdák: Alapvetően lokális mozgástervezési módszerről van szó. A lokális jelleg abból származik, hogy az optimalizálás során kizárolag a következő időszak végéig tekintünk előre az időben. Emellett a célpontnak csak az *irány* játszik szerepet a sebesség kiválasztásánál, az algoritmus nem vesz tudomást a szabad tér összefüggőségéről. Ezek miatt a DWA algoritmus is érzékeny a lokális minimum jellegű csapda-szituációkra.

Súlyok megválasztása: A célfüggvényben szereplő súlyok megválasztására nincs semmilyen algoritmus, azokat empirikus úton kell beállítani. A helyzetet rontja, hogy különböző szituációk sikeres megoldásához különböző súlyozásra lehet szükség.

IRODALOM

- [1] Bruno Siciliano; Oussama Khatib (Eds.), Springer Handbook of Robotics. Springer Verlag, 2008, ISBN 978-3-540-23957-4
- [2] S.M. LaValle, Planning Algorithms. Cambridge University Press, 2006. Online változat: <http://planning.cs.uiuc.edu/>
- [3] J. Borenstein and Y. Koren, “Real-time Obstacle Avoidance for Fast Mobile Robots,” IEEE Trans. Syst., Man, Cybern., vol. 19, pp. 1179–1187, 1989.
- [4] J. Borenstein and Y. Koren, “The vector field histogram – fast obstacle avoidance for mobile robots,” IEEE Trans. Robot. Autom., vol. 7, pp. 278–288, 1991.
- [5] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” IEEE Robot. Autom. Mag., vol. 4, pp. 23–33, 1997.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszer technikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 95. oldal
--	--	--

FÜGGELEK

F.1 feladat – Odometria hibaszámítás és UMBMark kalibráció

Egy differenciális meghajtású mobil robot relatív pozícióját és orientációját odometria segítségével határozzuk meg. A hajtott kerekek és a hozzájuk tartozó motorok közötti áttételi arány 5. A kerekek elmozdulását a motorokra szerelt, $10000 \frac{\text{impulzus}}{\text{fordulat}}$ felbontású, négyzetes kiértékeléssel működő inkrementális adók segítségével mérjük. A kerekek névleges átmérője 50 cm , a névleges keréktávolság 1 m . A legutóbbi szerviz alkalmával a robot kerekeit lecseréltek. Az új kerekek szélesebbek, és puhább anyagból készültek, mint az eredetiek. A szélesebb kerekek miatt a tényleges keréktávolság (a kerekek talajjal való érintkezési pontjainak távolsága) 10%-kal nagyobb lett a névlegesnél. A puhább anyagú kerekek felülete a robot súlya miatt kissé benyomódik, ezért a tényleges kerékátmérő 1%-kal kisebb lett a névlegesnél. A dead-reckoning algoritmus paramétereit azonban elfelejtették átállítani.

- a) A robot irányítórendszerének egy 60° -os egy helyben való fordulást adunk ki parancsként. Mennyit fog valójában fordulni a robot?
- b) A fordulás után egy 30 méter hosszú, egyenes út megtételét adjuk ki parancsként. Mennyi lesz a ténylegesen megtett út hossza?
- c) Mennyi lesz a mozgássorozat végén a felhalmozódott abszolút pozícióhiba? Tegyük fel, hogy a kerekek nem csúsztak meg, és a mozgás előtt a robot abszolút pozícióját pontosan ismertük.
- d) A fenti hibák korrigálására futtassuk le az UMBMark algoritmust egy 4 méter oldalhosszúságú négyzeten, hogy a fenti hibákat kiküszöböljük. Mutassuk meg, hogy az algoritmus ténylegesen korrigálja a keréktávolság hibáját.

A megadott adatok:

$$\text{Inkrementális adó felbontása: } C_{enc} = 10000 \frac{\text{impulzus}}{\text{fordulat}} = 40000 \frac{\text{inkrement}}{\text{fordulat}}$$

$$\text{Motor és kerék közötti áttétel: } n = 5$$

$$\text{Névleges kerékátmérő: } D_{nom} = 0,5\text{ m}$$

$$\text{Kerék kerületének relatív hibája: } h_p = -0,01$$

$$\text{Tényleges kerékátmérő: } D_{act} = D_{nom}(1 + h_p)$$

$$\text{Névleges keréktávolság: } W_{nom} = 1\text{ m}$$

$$\text{Keréktávolság relatív hibája: } h_W = 0,1$$

$$\text{Tényleges keréktávolság: } W_{act} = W_{nom}(1 + h_W)$$

Dead-reckoning összefüggések:

$$\Delta s_l = \frac{D\pi}{nC_{enc}} N_l, \quad \Delta s_r = \frac{D\pi}{nC_{enc}} N_r \quad (1)$$

ahol N_l és N_r a bal és jobb kerekekhez tartozó inkrementszám, Δs_l és Δs_r pedig a kerekek által megtett út.

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszerteknikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 96. oldal
--	--	--

A robot elmozdulása egyenes haladás esetén:

$$\Delta s = \frac{\Delta s_l + \Delta s_r}{2} \quad (2)$$

A robot szögelfordulása egy helyben fordulás esetén:

$$\Delta\varphi = \frac{\Delta s_r - \Delta s_l}{W} \quad (3)$$

a) A fenti összefüggésekkel kifejezhető a robot tényleges elfordulása:

$$\Delta\varphi = \frac{D_{nom}(1+h_p)\pi}{nC_{enc}} \cdot \frac{N_r - N_l}{W_{nom}(1+h_w)} = \Delta\varphi_{nom} \frac{(1+h_p)}{(1+h_w)} = 60^\circ \cdot \frac{0,99}{1,1} = 54^\circ \quad (4)$$

b) Hasonló módon kifejezhető a tényleges elmozdulás is:

$$\Delta s = \frac{D_{nom}(1+h_p)\pi}{nC_{enc}} \cdot \frac{N_l + N_r}{2} = \Delta s_{nom}(1+h_p) = 30m \cdot 0,99 = 29,7m \quad (5)$$

c) A robot irányítórendszer által feltételezett végleges pozíció, ha a kezdőpontot (0,0)-nak tekintjük:

$$\begin{aligned} x_{nom} &= 30m \cdot \cos 60^\circ = 15m \\ y_{nom} &= 30m \cdot \sin 60^\circ = 25,98m \end{aligned} \quad (6)$$

A tényleges végpozíció:

$$\begin{aligned} x &= 29,7m \cdot \cos 54^\circ = 17,46m \\ y &= 29,7m \cdot \sin 54^\circ = 24,03m \end{aligned} \quad (7)$$

A pozícióhiba:

$$d = \sqrt{(x - x_{nom})^2 + (y - y_{nom})^2} = \sqrt{(17,46 - 15)^2 + (24,03 - 25,98)^2} = 3,14m \quad (8)$$

Megjegyzés:

Látható, hogy a feladat megoldásához nem szükséges az inkrementális adó felbontás, a motor és kerék közötti áttétel, valamint a névleges kerékátmérő és keréktávolság ismerete. Az összefüggések egyszerűen átrendezhetők úgy, hogy csak a relatív hibákkal, illetve azoknak a szögelfordulásra és elmozdulásra gyakorolt hatásával kelljen számolni.

d) UMBMark algoritmus alkalmazása⁶

Mivel az UMBMark algoritmus feltételezi, hogy a kerékátmérőt pontosan ismerjük, ennek hibáját tekintsük zérusnak⁷. Tehát vizsgáljuk a robot mozgását $h_p = 0$ és $h_w = 0,1$ értékek mellett. Ekkor az előző részfeladatok eredményeit felhasználva:

$$\Delta\varphi = \frac{D_0(1+h_p)\pi}{nC_{enc}} \cdot \frac{N_r - N_l}{W_{nom}(1+h_w)} = \Delta\varphi_0 \frac{(1+h_p)}{(1+h_w)} = 90^\circ \cdot \frac{1}{1,1} = 80,82^\circ \quad (9)$$

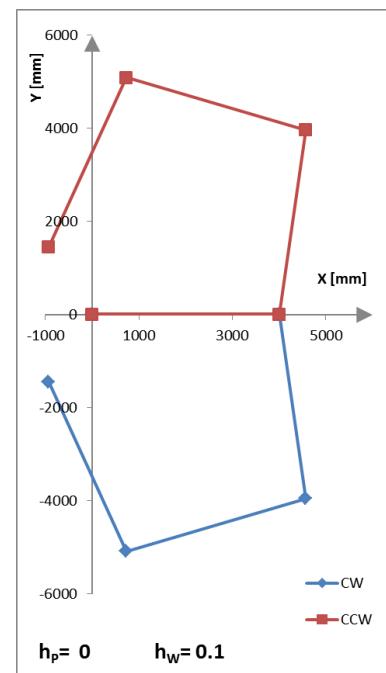
$$\Delta s = \frac{D_0(1+h_p)\pi}{nC_{enc}} \cdot \frac{N_l + N_r}{2} = \Delta s_0(1+h_p) = 4000 \text{ mm} \cdot 1 = 4000 \text{ mm} \quad (10)$$

Ezek után szimuláljuk a robot mozgását. Az egyszerűbb számítások kedvéért feltettük, hogy a robot kerekeinek átmérője megegyezik, tehát $E_d = 1$, így a robot egyenes vonalú mozgása valóban egyenes. Ezek után számítsuk ki a robot mozgását a négyzet mentén:

$$\begin{aligned} \varphi[k] &= \varphi[k-1] + \Delta\varphi \\ x[k] &= x[k-1] + \Delta s \cdot \cos \varphi[k] \\ y[k] &= y[k-1] + \Delta s \cdot \sin \varphi[k] \end{aligned} \quad (11)$$

CW irány							
Lépésszám	x_{nom}	x_{sim}	y_{nom}	y_{sim}	φ_{nom}	φ_{sim}	
0	0	0	0	0	0	0	
1	4000	4000	0	0	0	0	
2	4000	4569	-4000	-3959	-90	-81,82	
3	0	731	-4000	-5086	-180	-163,6	
4	0	-930	0	-1448	-270	-245,5	

CCW irány							
Lépésszám	x_{nom}	x_{sim}	y_{nom}	y_{sim}	φ_{nom}	φ_{sim}	
0	0	0	0	0	0	0	
1	4000	4000	0	0	0	0	
2	4000	4569	4000	3959	90	81,82	
3	0	731	4000	5086	180	163,6	
4	0	-930	0	1448	270	245,5	



⁶ Az UMBMark algoritmus leírását és összefüggéseit, valamint a használt jelöléseket a mobil robotok tájékozódásáról szóló 6.2.1 fejezet tartalmazza

⁷ Az algoritmus a kerékátmérők arányának hibáját képes kiküszöbölni, annak feltételezése mellett, hogy a kerékátmérők számtani átlaga megegyezik a névleges értékkal

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás renderszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 98. oldal
--	---	--

A *nom* indexű elemek a várt értékek, a *sim* indexűek pedig a hibával terhelt robot szimulációjából adódnak. Vegyük észre, hogy az odometria algoritmusával számítjuk ki az értékeket, hiszen az odometria is elemi elfordulásokból és elmozdulásokból képzi a robot pozíóját, csak jóval gyakrabban.

Látható, hogy mivel a robot tényleges szögelfordulása minden esetben kisebb, mint a szükséges, ezért a négyzet kifelé nyílik.

Az UMBMark algoritmus számításait elvégezve az eredményeinket az alábbi táblázatban foglaljuk össze:

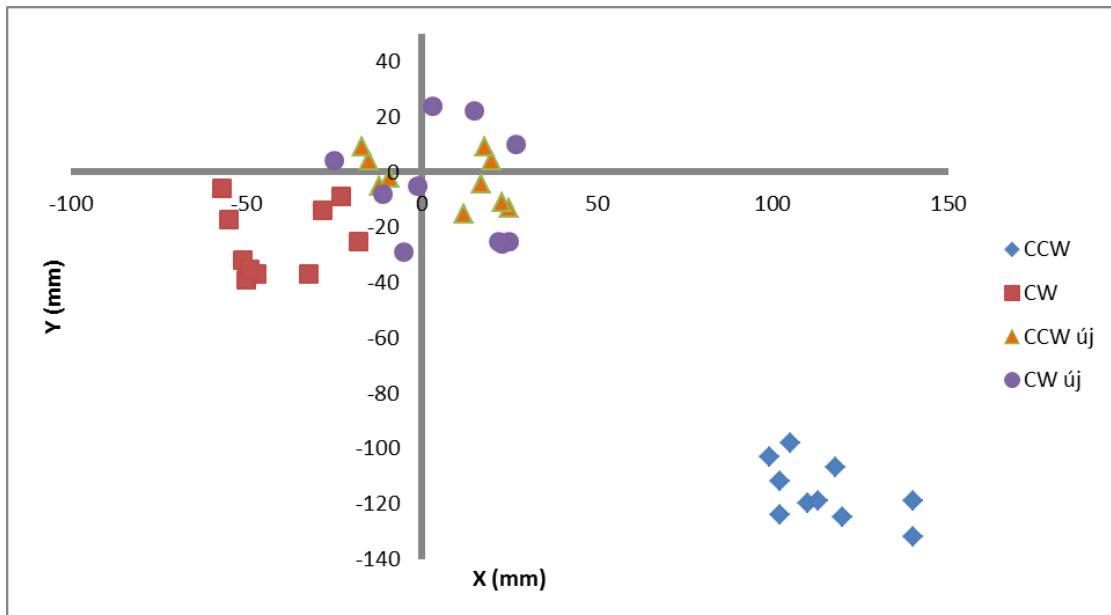
x_{cw} [mm]	-930	α [rad]	0.148629
x_{ccw} [mm]	-930	β [rad]	0.000000
y_{cw} [mm]	-1448		
y_{ccw} [mm]	1448		
α_x [rad]	0.116296574	R [mm]	∞ (nem számítható)
β_x [rad]	0.000000	E_w	1.104509
α_y [rad]	0.180961	Számított h_w	0.104509
β_y [rad]	0		
		E_b	1 (nem számítható)

A kapott eredményekből látszik, hogy a kalibráció a hibát $h_w = 0.1045$ értékre közelítette, tehát javította az eredményeinket. A „nem számítható” cellák esetén zéróosztó miatt nem elvégezhető a képletek kiértékelése. Ennek tudatában módosíthatjuk a paramétereket, és újból elvégezhetjük a kalibrációt egy sokkal pontosabb értékkel (a következő kalibrációkor a módosított paramétereket tekintjük „névlegesnek”).

A kapott eredmény helyett $h_w = 0.1$ értéket vártunk volna. Az eltérés oka a módszer levezetésében keresendő: a feltételezhetően kis α és β szögek miatt a szerzők sok helyen élnek a $\sin(x) \approx x$ közelítéssel, így nem is lehet pontos az eredményünk.

Az ebben a példában található 10%-os eltérés a keréktávolságban egy nagyon nagy érték, ennél a valóságban sokkal pontosabban tudjuk méni a robot paramtereit. Amennyiben mégsem, az algoritmus többszöri lefuttatásával minden lépében javíthatunk.

Elvégeztük a kalibrációt egy valós roboton is, amely kerékátmérője $D_0 = 0,037\text{ m}$, keréktávolsága $W_{nom} = 0,1285\text{ m}$. A kalibráció során a robot egy 3 méter oldalhosszúságú négyzetet járt be 10 alkalommal. Az alábbi ábrán a kalibráció előtti és utáni eredmények láthatóak:



Látható a nem szisztematikus hibák okozta szórás az eredményekben, valamint az eltérő kerékátmérők miatt a CW és CCW eredmények nem szimmetrikusak az x tengelyre.

Az eredmények:

x_{cw} [mm]	-41.0
x_{ccw} [mm]	114.0
y_{cw} [mm]	-25.0
y_{ccw} [mm]	-114.0

α_x [rad]	-0.006083333
β_x [rad]	0.012917

α [rad]	-0.006750
β [rad]	0.012250

R [mm]	244899
E_w	0.995721
Számított h_w	-0.004279

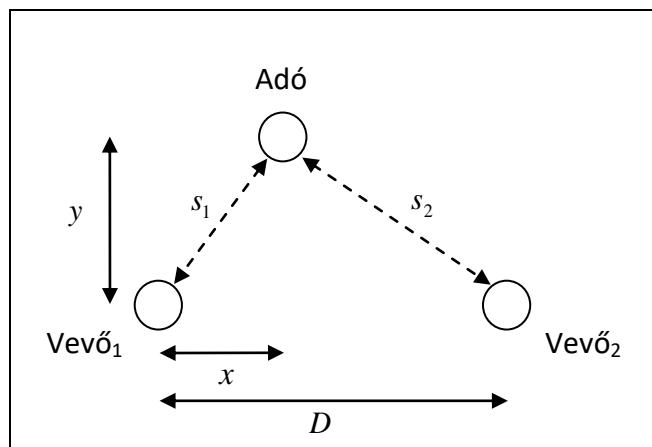
α_y [rad]	-0.007417
β_y [rad]	0.011583333

E_D	1.000154
-------	----------

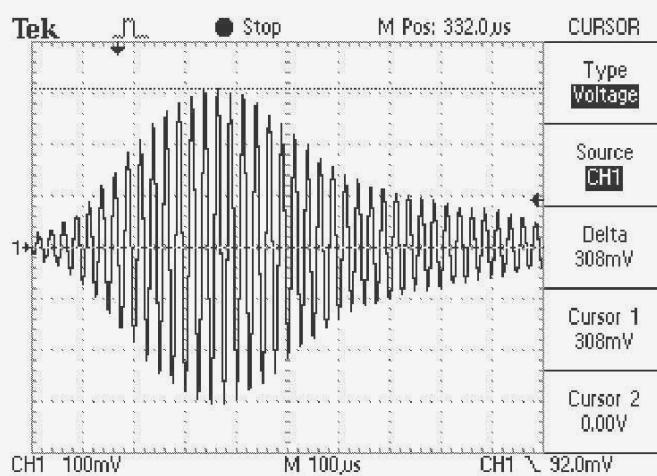
Ahogy várható, itt már a kerekek átmérőjének különbségében is adódott korrigálálandó hiba, habár egy nagyságrenddel kisebb, mint a keréktávolság hibája.

F.2 feladat – Ultrahangos szonár rendszer

Egy mobil robot pozícióját ultrahangos szonárrendszer segítségével határozzuk meg, amely két rögzített vevőből, és egy, a robotra szerelt adóból áll. Az adó és a vevőegységek elrendezését az alábbi ábra mutatja:



- Adjuk meg az adóegység pozíciójának x és y koordinátáit a vevőktől mért s_1 és s_2 távolságmérési eredmények, valamint a két vevő D távolságának ismeretében!
- A távolságmérések mérési bizonytalanságát Δs_1 és Δs_2 additív hibával modellezzen. Adja meg, hogy ezek a mérési hibák hogyan befolyásolják az x irányú pozíció-koordináta pontosságát! A pozícióhiba csak a távolságmérési hibáktól függ, vagy más tényezők is befolyásolják? Ha vannak ilyen tényezők, melyek ezek?
- Legyen a két vevő távolsága $2m$, a méréshez használt ultrahang frekvenciája $40kHz$, a környezeti hőmérséklet $20^\circ C$. Az ultrahang-csomag beérkezési idejét a vevő a maximális amplitúdóhoz tartozó időpillanat alapján határozza meg (ld. ábra). Tegyük fel, hogy a beérkező jelre szuperponálódó zajok miatt a maximum érzékelésében lehet ± 1 periódusidőnyi hiba.



Tekintsük a következő abszolút pozíciókat:

- i. $x = 1m$, $y = 5m$
- ii. $x = 1m$, $y = 1m$
- iii. $x = 1m$, $y = 0.15m$
- iv. $x = 0.15m$, $y = 1m$

Számítsuk ki minden pozícióban az „elvétett” maximumhelyek által okozott maximális x és y irányú pozícióhibákat!

- d) Tegyük fel, hogy nincs hőmérőnk a rendszerben, ezért az egyszerűség kedvéért egy táblázatból kinézett, 0°C -hoz tartozó hangsebességgel számolunk, amelynek értéke 331.45 m/s . Számítsuk ki az előző részfeladatban megadott első három pontra, illetve az $(1, 0.3)$ pontra az így keletkező pozíciómérési hibát!

- a) A Pitagorasz-tétel segítségével felírjuk az x , y , s_1 és s_2 közötti összefüggéseket:

$$\begin{aligned} x^2 + y^2 &= s_1^2 \\ (D - x)^2 + y^2 &= s_2^2 \end{aligned} \tag{12}$$

Ebből átrendezéssel:

$$\begin{aligned} x &= \frac{D^2 + s_1^2 - s_2^2}{2D} \\ y &= \sqrt{s_1^2 - x^2} \end{aligned} \tag{13}$$

Vegyük észre, hogy az y-koordináta számításánál két megoldás is lehetséges lenne. Mivel a gyakorlatban a két vevőegység a robot mozgásterének sarkaiban kerül elhelyezésre, ezért csak a pozitív y-értéknek van fizikailag értelme (a negatív érték nem lehetséges, mivel ahhoz a robotnak a pályán kívül kellene tartózkodnia).

- b) Jelölje $s_{1,m}$ és $s_{2,m}$ a mért távolságokat, Δs_1 és Δs_2 pedig a mérések hibáit:

$$\begin{aligned} s_{1,m} &= s_1 + \Delta s_1 \\ s_{2,m} &= s_2 + \Delta s_2 \end{aligned} \tag{14}$$

Behelyettesítve a x-koordináta képletébe:

$$x_m = \frac{D^2 + s_{1,m}^2 - s_{2,m}^2}{2D} = \underbrace{\frac{D^2 + s_1^2 - s_2^2}{2D}}_x + \underbrace{\frac{2s_1\Delta s_1 + \Delta s_1^2 - 2s_2\Delta s_2 - \Delta s_2^2}{2D}}_{\Delta x} = x + \Delta x \tag{15}$$

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 102. oldal
--	---	---

A képletből látható, hogy az x -irányú pozícióhiba nem csak a távolságmérések hibáitól függ. Befolyásoló tényezők még:

- Az egyes vevőktől való abszolút távolság (s_1, s_2), vagyis ugyanakkora távolságmérési hiba különböző pozíciókban különböző pozícióhibákat eredményez.
- A két vevő távolsága (D)

c) A hang terjedési sebessége 20°C-on:

$$c = 331.45 \sqrt{\frac{T[K]}{273.15}} = 331.45 \sqrt{\frac{293.15}{273.15} \frac{m}{s}} = 343.37 \frac{m}{s} \quad (16)$$

Az ultrahang periódusideje:

$$t_p = \frac{1}{40000} s = 25 \mu s \quad (17)$$

Egy periódus „elvétéséből” származó távolságmérési hiba:

$$\Delta s = c \cdot t_p = 8.584 mm \quad (18)$$

Attól függően, hogy a két távolságmérésnél milyen előjelű a hiba, négy pozíciószélsőérték adódhat:

$$\begin{aligned} x_{m,1} &= \frac{D^2 + (s_1 - \Delta s)^2 - (s_2 - \Delta s)^2}{2D}, & y_{m,1} &= \sqrt{(s_1 - \Delta s)^2 - x_{m,1}^2} \\ x_{m,2} &= \frac{D^2 + (s_1 - \Delta s)^2 - (s_2 + \Delta s)^2}{2D}, & y_{m,2} &= \sqrt{(s_1 - \Delta s)^2 - x_{m,2}^2} \\ x_{m,3} &= \frac{D^2 + (s_1 + \Delta s)^2 - (s_2 - \Delta s)^2}{2D}, & y_{m,3} &= \sqrt{(s_1 + \Delta s)^2 - x_{m,3}^2} \\ x_{m,4} &= \frac{D^2 + (s_1 + \Delta s)^2 - (s_2 + \Delta s)^2}{2D}, & y_{m,4} &= \sqrt{(s_1 + \Delta s)^2 - x_{m,4}^2} \end{aligned} \quad (19)$$

Az x - és y -irányú pozícióhibák:

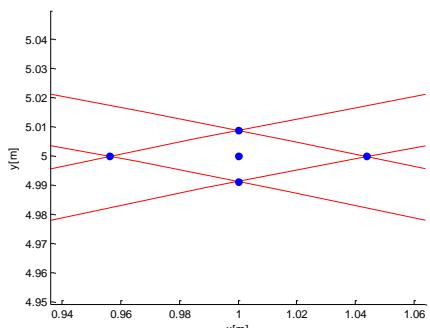
$$\begin{aligned} \Delta x_i &= x_{m,i} - x, \\ \Delta y_i &= y_{m,i} - y, \quad i \in \{1,2,3,4\} \end{aligned} \quad (20)$$

BME Villamosmérnöki és Informatikai Kar Automatizálási és Alkalmazott Informatikai Tanszék	Robotirányítás rendszertechnikája előadás 6. fejezet	RRT_EA_6.DOCX 2017.12.08. Dr. Tevesz Gábor VI / 103. oldal
--	---	---

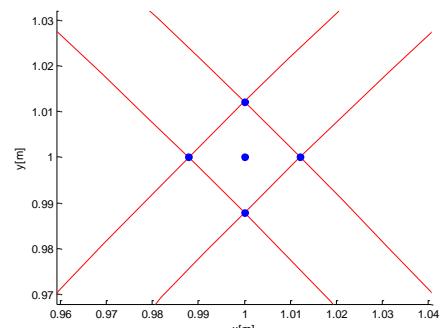
A következő táblázatban a megadott pozíciókban lévő névleges távolságok, valamint a hibával terhelt pozíció-koordináták mellett a pozíciókoordináta-hibák értékeit láthatjuk:

x	y	1	5	1	1	1	0.15	0.15	1
s_1	s_2	5.099	5.099	1.414	1.414	1.011	1.011	1.011	2.103
$x_{m,1}$	$y_{m,1}$	1	4.991	1	0.988	1	0.072	0.155	0.991
$x_{m,2}$	$y_{m,2}$	0.956	5	0.988	1	0.991	0.15	0.137	0.993
$x_{m,3}$	$y_{m,3}$	1.044	5	1.012	1	1.009	0.15	0.163	1.007
$x_{m,4}$	$y_{m,4}$	1	5.009	1	1.012	1	0.2	0.145	1.009
Δx_1	Δy_1	0	-0.009	0	-0.012	0	-0.078	0.005	-0.009
Δx_2	Δy_2	-0.044	0	-0.012	0	-0.009	0	-0.013	-0.007
Δx_3	Δy_3	0.044	0	0.012	0	0.009	0	0.013	0.007
Δx_4	Δy_4	0	0.009	0	0.012	0	0.05	-0.005	0.009

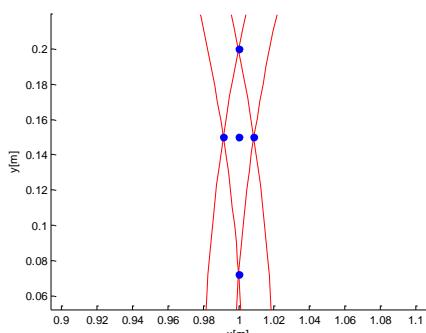
Ábrázoljuk a négy elrendezés mérési eredményeit (a középső pont minden esetben a névleges pozíciót jelöli):



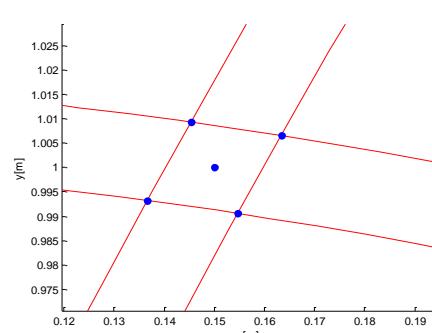
(1, 5)



(1, 1)



(1, 0.15)



(0.15, 1)

Megjegyzés:

Vegyük észre, hogy az első három pozíciótól, ha a két távolságmérési hiba előjele megegyezik (mindkettő pozitív, vagy mindkettő negatív), akkor csak az y-koordináta változik. Ha a mérési hibák előjele különböző, akkor csak az x-koordináta változik. Ennek az az oka, hogy ezeknél az eseteknél a névleges x-koordináta értéke 1m, vagyis ezek a pontok éppen a két vevőt összekötő szakasz felező merőlegesén vannak.

- d) A hang terjedési sebessége 0°C-on kisebb, mint 20°C-on. Ennek megfelelően a mért terjedési időkből származtatott távolságok is kisebbek lesznek a ténylegesnél. Mivel a hiba sebesség dimenziójú, ezért a pozícióhiba várhatóan távolságfüggő (illetve abszolút pozíciófüggő) lesz.

Az egyes névleges pozíciókhöz először meghatározzuk a tényleges időmérési adatokat:

$$t_1 = \frac{s_1}{c_{20^\circ C}}, \quad t_2 = \frac{s_2}{c_{20^\circ C}}, \quad (21)$$

majd ebből kiszámítjuk a mért távolságokat:

$$s_{1,m} = t_1 \cdot c_{0^\circ C}, \quad s_{2,m} = t_2 \cdot c_{0^\circ C}, \quad (22)$$

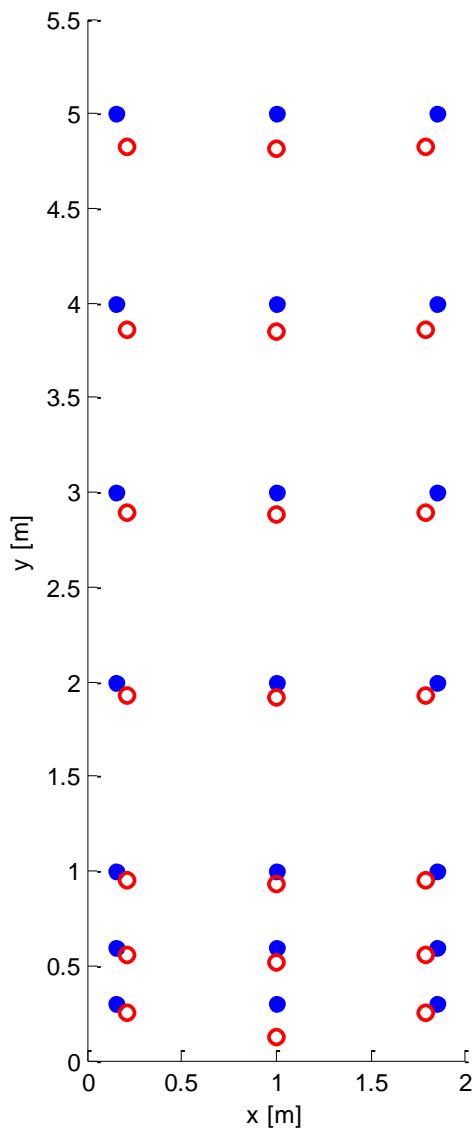
Ebből meghatározhatók a mért pozíciók, illetve a pozícióhibák. Az eredményeket az alábbi táblázat tartalmazza:

x	y	1	5	1	1	1	0.3	1	0.15
s_1	s_2	5.099	5.099	1.414	1.414	1.044	1.044	1.011	1.011
t_1 (ms)	t_2 (ms)	14.85	14.85	4.12	4.12	3.04	3.04	2.95	2.95
$s_{1,m}$	$s_{2,m}$	4.922	4.922	1.365	1.365	1.008	1.008	0.976	0.976
x_m	y_m	1	4.819	1	0.929	1	0.125	1	-
Δx	Δy	0	-0.181	0	-0.071	0	-0.175	0	-

Mivel csak a középvonalon elhelyezkedő pontokat vizsgáltuk, és a sebességhiba azonos előjelű távolságmérési hibát okoz minden távolságban, ezért csak y-irányú pozícióhibát tapasztalunk. Érdekes megfigyelni, hogy az x-tengelyhez közeledve a pozícióhiba előbb csökken, majd növekedni kezd.

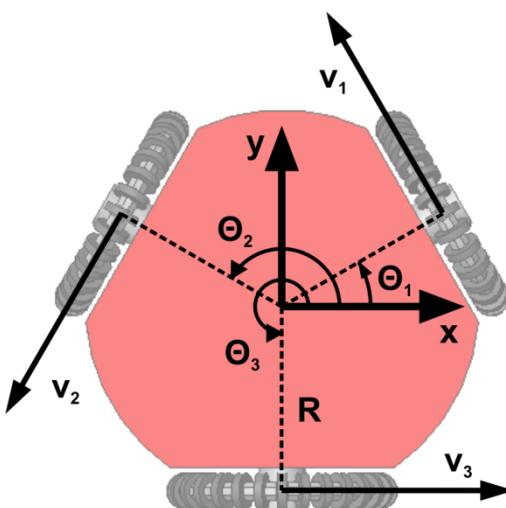
Az utolsó pontban már nem is lehetséges a pozíció számítása, ugyanis az $s_{1,m}$ és $s_{2,m}$ távolságok összege kisebb, mint D. Ezt úgy is megfogalmazhatjuk, hogy a két vevő köré rajzolt $s_{1,m}$, illetve $s_{2,m}$ sugarú köröknek nincs metszéspontja.

A következő ábra 21 különböző pontban mutatja a hőmérsékletmérés hibájából adódó pozícióhibákat. A teli körök jelentik a valódi, az üres körök a mért pozíciókat:



F.3 feladat – Omnidirekcionális hajtás

Egy mobil robot háromkerekű omnidirekcionális hajtással rendelkezik. A kerekek érintőirányban ideálisan tapadnak, tengelyirányban szabadon gördülnek (ld. ábra). A kerekek szimmetrikus elrendezésűek, tengelyeik egymással 120° -os szöget zárnak be. Mindhárom kerék érintési pontja $R = 0.2\text{m}$ távolságban van a robot referencia pontjától. A robot koordinátarendszerét az ábrán látható módon vettük fel. A kerekek kerületi sebességét v_1 , v_2 és v_3 jelöli, a nyilak a pozitív forgásirányt mutatják. A robot referencia pontjának sebességét $v = (v_x, v_y)$, szögsebességét ω jelöli a robot koordinátarendszerében.



Határozza meg az egyes kerekek sebességét a következő esetekben:

- a) $v = 0, \omega = 2 \text{ rad/s}$
- b) $v = v_y = 1 \text{ m/s}, \omega = 0$
- c) $v = v_x = 1 \text{ m/s}, \omega = 0$
- d) $|v| = 1 \text{ m/s}, \angle v = 45^\circ, \omega = 0$
- e) $|v| = 1.5 \text{ m/s}$, és a robot egy 1m sugarú körpályán halad, jobbra kanyarodva

- a) A robot egy helyben forog, sebessége nincs, csak szögsebessége. A robot kerületi sebessége megegyezik a kerekek kerületi sebességével:

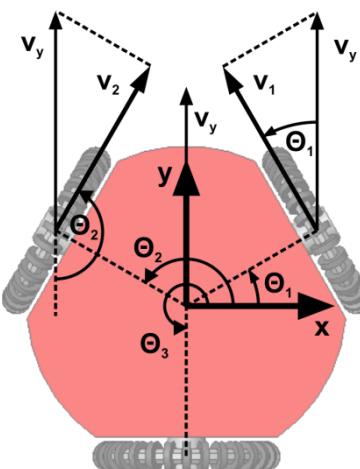
$$v_1 = v_2 = v_3 = \omega \cdot R = 0.4 \text{ m/s}, \quad (23)$$

- b) Az 1. és a 2. kerék járul hozzá a mozgáshoz, a 3. kerék merőleges irányban szabadon gördül. Az ábra alapján könnyen belátható:

$$v_1 = v_y \cdot \cos \theta_1 = 1 \cdot \cos(30^\circ) = \frac{\sqrt{3}}{2} \frac{m}{s}$$

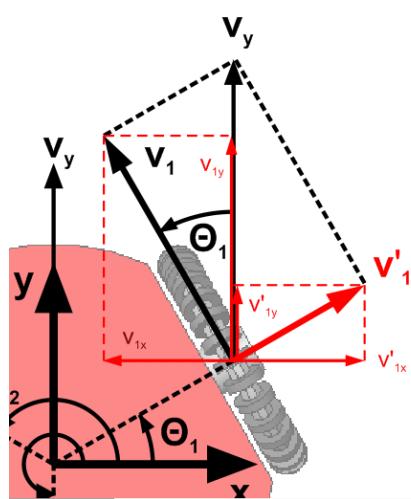
$$v_2 = v_y \cdot \cos \theta_2 = 1 \cdot \cos(150^\circ) = -\frac{\sqrt{3}}{2} \frac{m}{s}$$

$$v_3 = 0 \quad (=1 \cdot \cos(270^\circ) = v_y \cdot \cos(\theta_3)), \quad (24)$$



Első ránézésre furcsának tűnhet, hogy a kialakuló sebesség nagyobb, mint bármelyik kerék kerületi sebessége. Annak megértéséhez, hogy ez lehetséges, nézzük meg közelebbről egy kerék hatását a robot sebességére! Tegyük fel, hogy a robotot az y-tengely mentén megfelezzük, és a középvonalánál megvezetjük egy súrlódásmentes sínen. Ekkor egy kerék is elég a meghajtáshoz, a megvezetés gondoskodik arról, hogy ne boruljon fel. Ez a képzeletbeli megvezetés azt eredményezi, hogy a robot x-irányú sebessége nullára korlátozzuk

(ezt a valóságban a szimmetrikusan elhelyezkedő másik kerék ellenkező irányú forgatásával érjük el). A v_1 keréksebesség két komponense v_{1x} és v_{1y} . Ahhoz azonban, hogy az eredő x-irányú sebesség zérus legyen, szükségünk van egy másik, ellenkező x-irányú sebességek komponensre. Ezt fogják szolgáltatni a kerék felületén lévő görgők. A kontaktuspontnál lévő görgő sebessége (v'_1) éppen akkora, hogy x-irányú komponensére teljesüljön $v_{1x} + v'_{1x} = 0$. Ugyanakkor v'_1 -nek van y-irányú komponense is (v'_{1y}). Az eredő y-irányú sebesség a kerék és a görgők y-irányú sebességeinek összege: $v_y = v_{1y} + v'_{1y}$, ami nagyobb is lehet, mint v_1 .

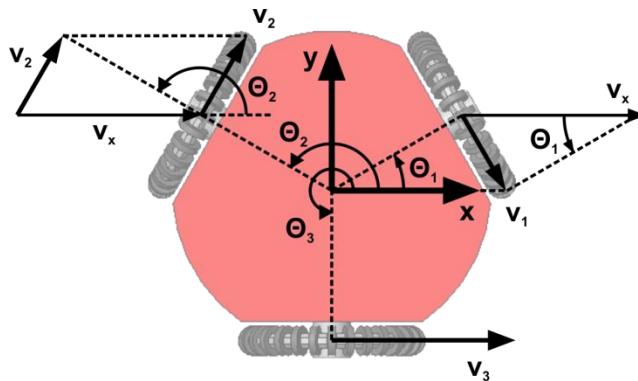


c) Az ábra alapján könnyen belátható:

$$v_1 = -v_x \cdot \sin \theta_1 = -1 \cdot \sin(30^\circ) = -\frac{1}{2} \frac{m}{s}$$

$$v_2 = -v_x \cdot \sin \theta_2 = -1 \cdot \sin(150^\circ) = -\frac{1}{2} \frac{m}{s}$$

$$v_3 = v_x = 1 \frac{m}{s} \quad (= -1 \cdot \sin(270^\circ) = -v_x \cdot \sin(\theta_3)), \quad (25)$$



d) Miután felírtuk v_x , v_y és ω hatását külön-külön, foglaljuk össze őket egy összefüggésbe. A szuperpozíció elve miatt ezt megtehetjük.

$$v_1 = -v_x \cdot \sin \theta_1 + v_y \cdot \cos \theta_1 + \omega \cdot R$$

$$v_2 = -v_x \cdot \sin \theta_2 + v_y \cdot \cos \theta_2 + \omega \cdot R$$

$$v_3 = -v_x \cdot \sin \theta_3 + v_y \cdot \cos \theta_3 + \omega \cdot R, \quad (26)$$

Mátrixos formában:

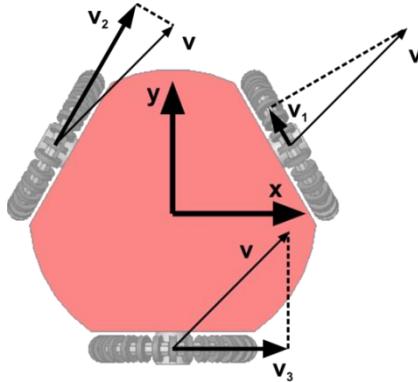
$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{bmatrix} -\sin \theta_1 & \cos \theta_1 & R \\ -\sin \theta_2 & \cos \theta_2 & R \\ -\sin \theta_3 & \cos \theta_3 & R \end{bmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} = \begin{bmatrix} -0.5 & 0.866 & 0.2 \\ -0.5 & -0.866 & 0.2 \\ 1 & 0 & 0.2 \end{bmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} \quad (27)$$

A feladat szerint a robot 45° -ban mozog, forgás nélkül: $|v| = 1 m/s$, $\angle v = 45^\circ$, $\omega = 0$
Ebből következik:

$$v_x = \sqrt{2}/2, \quad v_y = \sqrt{2}/2, \quad , \quad (28)$$

Amiből:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{bmatrix} -0.5 & 0.866 & 0.2 \\ -0.5 & -0.866 & 0.2 \\ 1 & 0 & 0.2 \end{bmatrix} \cdot \begin{pmatrix} 0.707 \\ 0.707 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.259 \\ -0.966 \\ 0.707 \end{pmatrix} \quad (29)$$



- e) A robot jobbra kanyarodik, tehát a szögsebessége negatív. A kerületi sebességből és a körpálya sugarából adódik a szögsebesség nagysága is:

$$\omega = -\frac{|v|}{r} = -\frac{1.5 \text{ m/s}}{1\text{m}} = -1.5 \frac{\text{rad}}{\text{s}}, \quad (30)$$

A robot pillanatnyi sebességének csak y irányú komponense van, tehát $v = v_y$. Az előző részfeladatban felírt képlet segítségével:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{bmatrix} -0.5 & 0.866 & 0.2 \\ -0.5 & -0.866 & 0.2 \\ 1 & 0 & 0.2 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1.5 \\ -1.5 \end{pmatrix} = \begin{pmatrix} 0.999 \\ -1.599 \\ -0.3 \end{pmatrix} \quad (31)$$

