

Óbudai Egyetem

Habilitációs téziszfüzet



Szenzor alapú, nem konvencionális pályatervező eljárások fejlesztése mobil ágensek részére

Dr. Nagy István
egyetemi docens

**Alkalmazott Informatikai és Alkalmazott
Matematikai Doktori Iskola**

Budapest, 2026.

Tartalomjegyzék

1. A kutatás előzményei	5
1.1 Bevezetés, motiváció és célkitűzések	5
1.2 Bevezető információk és alapvető definíciók a témakörrel kapcsolatban.....	8
1.2.1 Mobilrobot generációk.....	9
1.2.2 Térképkészítés, munkaterület és annak modellezése, konfigurálása	12
1.2.3 Alapvető ismeretek a pályatervezésről és navigációról	16
1.3 Szakirodalom áttekintése a témakörrel kapcsolatban	19
1.3.1 Energia optimalására törekvő pályatervező eljárások	20
1.3.2 Hibrid elektromos (HEV) járművek irányítási rendszerei	21
1.3.3 Multi-ágensű dinamikus környezetben működő pályatervező és feltérképező algoritmusokkal kapcsolatos kutatások	22
1.3.4 VR alapú technológia alkalmazása az irányítási rendszerekben	23
2. Új tudományos eredmények	23
2.1. UAV-k energiaoptimális pályatervezésére irányuló navigációs algoritmus kifejlesztése	24
2.1.1. A tézis rövid kifejtése.....	25
2.1.2. Eredmények és összefoglalás	26

2.2. Multiágens alapon történő potenciálmezős térképkészítés	27
2.2.1. A tézis rövid kifejtése.....	27
2.2.2. Eredmények és összefoglalás	29
2.3. Lágygörbe (<i>Spline</i>) alapú optimális pályatervezés mobil ágensek számára	30
2.3.1. A tézis rövid kifejtése.....	32
2.3.2. Eredmények és összefoglalás	33
2.4. Multiágensű alapon történő lokalizációs hibaszámításos modell dinamikus környezetben.....	33
2.4.1. A tézis rövid kifejtése.....	34
2.4.2. Eredmények és összefoglalás	36
2.5. Elektromos járművek (EV) szenzor és vezérlőhálózatának feltérképezése és azok egymásra épülő modellezése	37
2.5.1. A tézisek rövid kifejtése.....	39
2.5.2. Eredmények és összefoglalás	41
2.6. A virtuális valóság (VR) bevonása a járművek irányításába	43
2.6.1. A tézis rövid kifejtése.....	44
2.6.2. Eredmények és összefoglalás	46
3. Irodalmi hivatkozások listája.....	48
4. Tézisekhez kapcsolódó legfontosabb irodalmi hivatkozások listája	52

4.1	A témához kapcsolódó további irodalmi	
	hivatkozások listája.....	54

1. A kutatás előzményei

1.1 Bevezetés, motiváció és célkitűzések

A mérnöki világ egyre közelebb áll annak felismeréséhez, hogy lassan már nincs az iparban olyan gyártási folyamat, amit ne lehetne robotokkal megvalósítani. Kezdődött azzal, hogy egy adagolási folyamatot megautomatizáltunk, majd a megmunkálást, tovább haladva a szállítást automatizáljuk, és végül ott tartunk, kis túlzással, hogy 3D nyomtatókkal állítjuk elő a 3D-s nyomtatókat. Lassan már tényleg közel állunk Elon Musk egyik „tréfás” definíciójához, ahol az embereket „*kognitív lépegető mobilrobotként*” próbálta szabadalmaztatni.

Visszatérve a „való világba”, a robotika, és persze az ipari robotika, szinte fékezhetetlen fejlődése magával vonzotta a mobilrobotok egyre szélesebb körű alkalmazását.

Míg Elon Musk az emberiséget próbálta besorolni a mobilrobotok közé, én megpróbálom mérnöki szemmel besorolni a mobilrobotot az emberiség közé. Szerintem *a mobilrobot (ágens), intelligenciájától függően, egy olyan autonóm vagy félautonóm mechatronikai eszköz, amely képes a mozgásának koordinálására, meghatározott feladat(ok) végrehajtására különféle működési területeken. A multi-ágensű rendszerekben, az optimális megoldás érdekében az ágensek különböző szinteken kooperálnak (együttműködnek) egymással a feladat(ok) sikeres végrehajtása érdekében*¹. Itt

¹ Igaz, hogy ezt én határoztam meg, de nem tézis, inkább csak egy általam kifejlesztett definíciónak számom.

érdeemes megemlíteni néhány működési területet, ahol a mobilrobot(ok) már elterjedten használatosak:

- **Ipari gyártás és logisztika** (raktározás, anyagszállítás, alkatrészellátás, ...)
- **Egészségügy** (gyógyszerek, vérminták szállítás, betegellátás megkönnyítése, takarítás)
- **Logisztikai feladatok** (vezető nélküli teherszállító eszközök, élelmiszer ellátás, csomagszállítás)
- **Mezőgazdaság** (permetezés, műtrágyaszórás, vadgazdálkodás, betakarítás)
- **Közzolgáltatások**, ...

A mobilrobotokkal kapcsolatos kutatások első *motivációját* számomra a NASA adta 1997-ben, mikor megtörtént a „Sojourner” első leszállása a Marson. Szinte minden érdekelt, ami ezzel a „kutató (felfedező) mobilrobottal” kapcsolatos volt. A járószerkezet, tápellátás, kommunikáció, lokalizálás, navigálás, kutató munka. Annyira bele is merültem a kutatásokba, hogy nem sokkal később el is jutottam egy konferenciára, ahol két előadásom is volt, az egyik látásalapú navigációról, a másik egy dinamikus pályatervezési módszerről szólt. [Nagy, 1997-1], [Nagy, 1997-2]. Későbbiekben továbbra is ezen a területen maradtam, csak kicsit szűkült a kutatási területem a pályatervezésekre illetve a lokalizációs hibákra. Ezzel kapcsolatban írtam a PhD értekezésemet is [Nagy, 2009].

Amikor az értekezésemet befejeztem (2009) a mobilrobotokkal kapcsolatos kutatások már óriási dinamikával fejlődtek. Eljutottunk az egyszerű (konvencionális) pályatervezésektől a napjainkban szinte már hétköznapivá vált „MI”-vel támogatott pályatervezésekig. Ez a fejlődés adta meg számomra azt a lökést, hogy továbbra is foglalkozzam ezzel a tudományterülettel és legyek naprakész a témakörben.

Követve a fejlődési tendenciákat úgy láttam, és *célként* azt tűztem ki, hogy a multi ágensű platformokkal foglalkozzak, mindezt úgy, hogy bizonyos szempontok szerint egyre jobb pályákat tudjak tervezni. Ezt azzal kívántam elérni, hogy az egyes ágensek szenzorfelszereltségét igyekszem megnövelni, irányítási rendszerét hierarchikussá tettem. Kutatásaim ebben az irányban folytatódtak.

A címben használt „**mobil ágensek**” megfogalmazás elég felületesnek tűnhet, ezért egy kicsit pontosítom, miért is ezt használom. A mobilrobotok egyik jellemző (fő) tulajdonsága a mobilitás, amit nagymértékben (de nem teljességében) két körülmény befolyásol. Egyik a környezet, amiben a mobil ágens (mobilrobot) mozog, másik annak járószerkezete. *Egyik dolgozatomban dolgozatban a pályatervező algoritmusokat jellemzően 2 környezeti közegre terveztem, melyek közül az egyik az általánosan elterjedt földfelszíni szabad térben történő pályatervezés (Field Robots), másik a levegőben lévő 3D-s pályatervezés.*

Ami az „**ágens**” szót illeti (lehet a „mobilrobot” hasonlóan kifejező lett volna), multi ágensű környezetben, és tudományos kutatásaimban általános elterjedése és jelentése miatt mindenhol ezt használtam.

Már csak egy szöösszetétel maradt, ami még talán egy csipetnyi magyarázatra szorul a „**nem konvencionális**”. Igen, itt lehetne vitát nyitni azon, hogy mi a konvencionális és hol a határa a nem konvencionálisnak. Minden pályatervezésnek az a célja, hogy a kezdőpont és a végpont közé kell egy folytonos, végrehajtható pályát tervezni, amely állhat: a.) véges számú lineáris szakaszokból (gráf típusú pálya); b.) véges számú geometriai primitívekből (Reeds-Shepp algoritmus); c.)

folytonos lágygörbéből (Spline-ok). Ezek bejárhatósága, a hajtásokra adott dinamikai hatása, erősen függ e sima szakaszok illesztési módjától, így a folytonos pályák sokfélék lehetnek, és velük kapcsolatban optimalizálási szempontok merülnek fel. Ezeket a módszereket (ismét saját egyéni besorolás alapján) a konvencionális pályatervezések közé sorolnám. A nem konvencionális pályatervezések a heurisztikus pályatervezésekre (2. táblázat), illetve a multiágensű környezetekre biztosan vonatkoznak (pl.: bio inspirált pályatervezés (SWARM, ACO²), önszerveződő (GPM+Ga³) alapúpályatervezés, -2.tézis-), de ide sorolnám a különleges elvárások alapján módosított „konvencionális” pályatervezéseket (-1.tézis-).

Mindenképpen meg kell említeni *Weierstrass-Mandelbrot* matematikusok nevét, akik fraktálszerű pályák (érdes felületek, kiugrások) leírására felállítottak egy függvényt, amely sajnos sehol sem differenciálható, így végrehajtása gyakorlatilag nem is lehetséges, de érdes felületek, illetve kaotikus mozgások modellezésére nagyon is alkalmas [Claire, 2020].

1.2 Bevezető információk és alapvető definíciók a témakörrel kapcsolatban

Annak érdekében, hogy ez a tézisfüzet mindenki számára érthető legyen, annak elején néhány alapvető tájékoztatást szeretnék adni az Olvasónak. Az 1.1 fejezet dőlt betűs részében

² ACO-Ant Colony Optimization

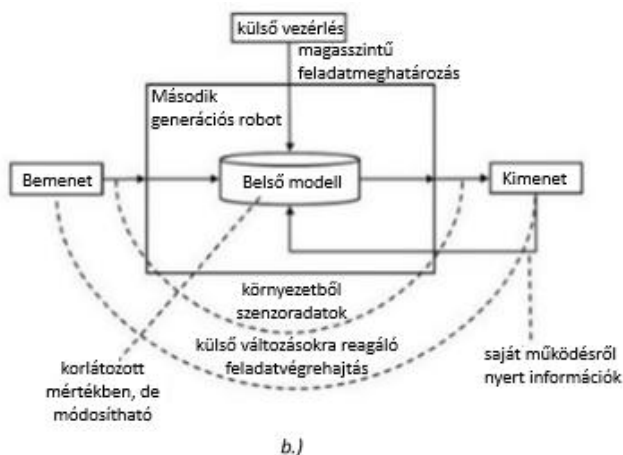
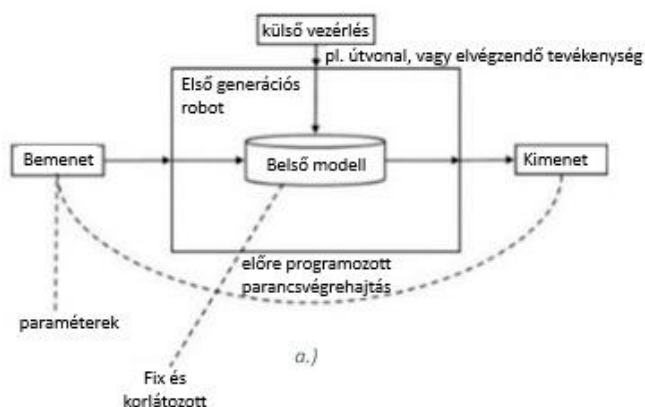
³ GPM+GA: Global Potential Map and Genetic Algorithm; Globális potenciáltérkép építés GA segítségével

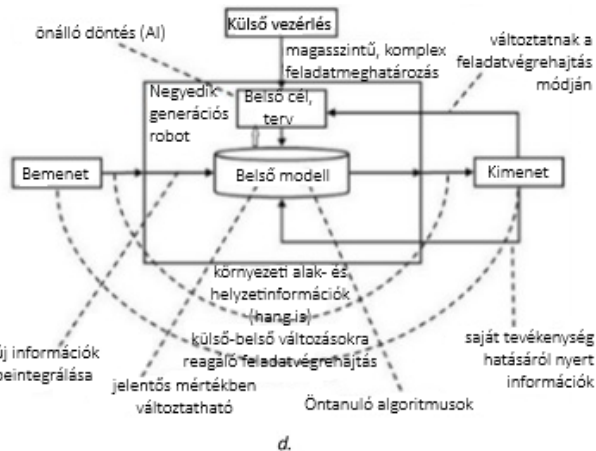
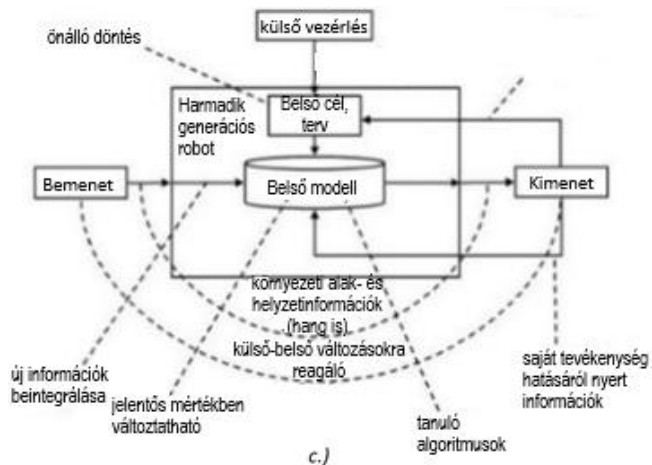
már definiáltam a mobilrobot fogalmát. Ezt a gondolatot tovább folytatva rátérnék az intelligencia kérdésére.

1.2.1 Mobilrobot generációk

A mobilrobot(ok) intelligenciáját a *generációk fejlettségi fokával* szeretném érzékeltetni. A generációk bemutatásához segítségül veszem *saját definícióimat* és ábraszorozatomat, (1.ábra) amit az előadásaimon is használok. Az ábrán egy klasszikus mechatronikai rendszer felépítését látjuk (bemenet-vezérlés-kimenet), ahol a generációk jobb érzékeltetése érdekében az „irányítási” rész kicsit részletesebben van szemléltetve. Itt valójában két részre osztottam a vezérlést, egy „belső modellre”, ami az „on board” processzor(ok)at és a működtető programot jelképezi, és ehhez kapcsolódik egy „külső vezérlés”, ami valójában a globális pályatervezésnél játszik szerepet. A későbbi generációknál (3., 4. generációk) a „belső modellt” tovább bővítettem egy „belső cél, terv” modullal (ez egy szoftveres modul), ami az „MI”-s (Mesterséges Intelligencia) illetve öntanuló rendszereknél kap majd jelentős szerepet. Igaz, hogy az ábrák részletesen feliratozottak, de egy rövid magyarázatot leírok a könnyebb értelmezés érdekében. A szaggatott vonalak jelölik az egyes blokkok feladatát, illetve az egyes feladatok hatásköreit (íves szaggatott vonalak). Az **1. generációs** (1. ábra a.) robotokba sorolom a processzor nélküli (mert valamikor ilyen mobilrobotok is voltak) nem változtatható, visszacsatolás nélküli, irányítóprogrammal rendelkező ágenseket. A **2. generációs** (1. ábra b.) robotok már visszacsatolással ellátott ágensek, melyeknél a belső modell (irányítóprogram) már kis mértékben változtatható. A **3. generációs** (1. ábra c.) robotok már kisebb fokú mesterséges intelligenciával ellátott robotok.

Az MI működése érdekében a belső modellt ki kellett bővíteni egy belső céllal, hogy a betanítás folyamata létrejöhessen. A **4. generációs** (1. ábra d.) ágensek már az öntanuló kategóriába sorolhatók. Itt az öntanulást a „belső modell” és a „belső cél, terv” modulok közötti visszacsatolás teszi lehetővé. Valójában 5 generációt állapítottam meg, ahol az **5. generáció** a „Multi-ágensű Rendszerek” (MÁR) kategóriája. Az 5. generáció a továbbiakban al-generációra bontható az ágensek közötti kommunikáció és az egyes ágensek „intelligenciaszintje” (master/slave) alapján (pl.: kommunikáló „master”-ek; nem kommunikáló „slave”-ek; nem kommunikáló „master”-ek; ... ;).





1. ábra Mobilrobot generációk szemléltetése: a.) első generáció; b.) második generáció; c.) harmadik generáció; d.) negyedik generáció

Azért tértem ki egy kicsit részletesebben ezekre a generációs meghatározásokra, mert a *pályatervezésben* fő szerepet játszik a mobilrobot *belső modellje*, illetve *külső vezérlése*, ugyanis a

pályatervezés módjától függően (globális/lokális) itt futnak a pályatervezést és navigációt megvalósító programok.

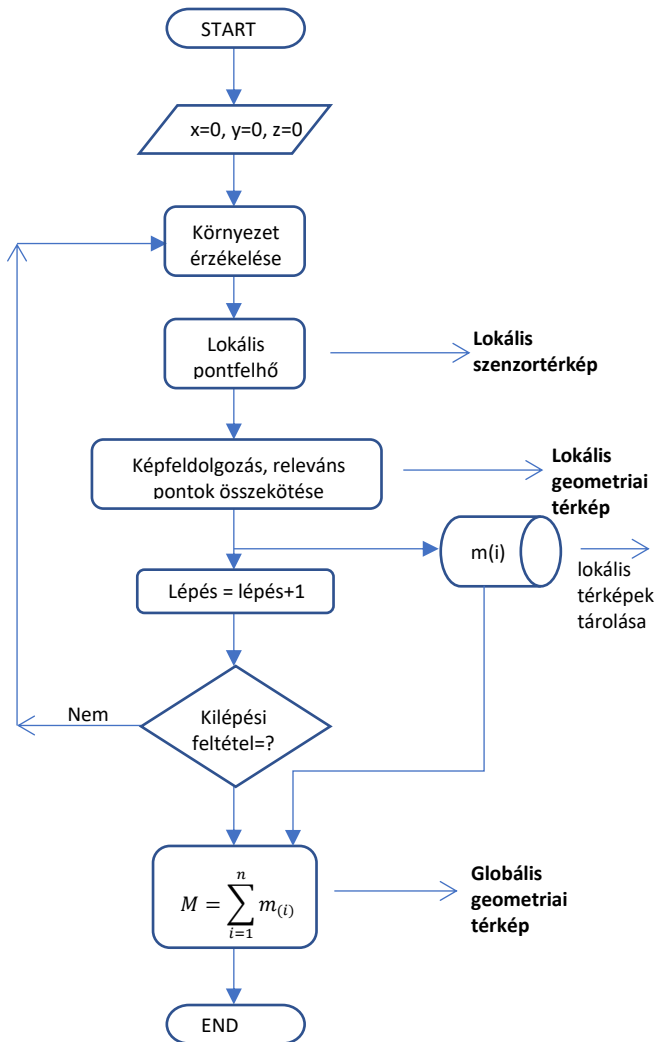
1.2.2 Térképkészítés, munkaterület és annak modellezése, konfigurálása

A munkatér (lehet 2D – ekkor munkaterület, illetve 3D - ez esetben munkatérnek hívjuk) az a része a térnek, ahol a mobilrobotnak valamilyen munkavégzést kell végrehajtania. A munkaterületek lehetnek ismert illetve ismeretlen környezetben. Ismeretlen környezetből, *feltérképezés* után (pl. valamilyen SLAM⁴ módszerrel) ismert munkaterületet kapunk. A munkaterületnek két jellemző része van: a *szabad területek (terek)*, ahol a mobilrobot mozoghat, illetve az *akadályok*, ahol nem mozoghat. Multi-ágensű rendszereknél az ágenseket mozgó (dinamikus) akadályoknak tekinthetjük (ez főleg nem kommunikáló rendszerekre érvényes). A munkaterületnél és akadályoknál maradvá megkülönböztetünk *statikus* (álló akadályok), illetve *dinamikus* (mozgó akadályok) munkatereket. Az ismeretlen környezettől az ismert munkatérig a térképkészítés folyamatával jutunk el. Megkülönböztetünk *lokális* illetve *globális* térképeket. Valójában a SLAM segítségével történő térképkészítésnél⁵ kiindulunk egy referenciapontból (legyen ez a $x=0/y=0/z=0$ koordináta), majd folyamatos egységlépésekkel lokalizáljuk magunkat ehhez a referenciaponthoz képest, utána „körülnézünk” a felszerelt szenzorok segítségével (LiDaR/Radar/US), aminek eredményeként megkapjuk a

⁴ SLAM- „Simultaneous Localization and Mapping”, Egyidejű lokalizáció és térképkészítés

⁵ A SLAM csak az egyik lehetséges módja a térképkészítésnek.

pontfelhőt a környezetünkről. Ebben a pontfelhőben, egy képfeldolgozó algoritmus segítségével összekötjük a releváns pontokat és megkapjuk a lokális környezetünk *geometriai* (néhány irodalomban „*metrikus*”-nak is emlegetett) térképét. Ezt a folyamatot ciklikusan ismételjük a megadott kilépési feltételig, majd ezeket a lokális térképeket, a megfeleltetést követően, ikonszerűen összeillesztjük, és megkapjuk a környezetünk, megadott határokkal körülvett, *globális* térképét. A folyamatot a 2. ábra szemlélteti az egyes lépésekben megtalálható térképfajtákkal.



2. ábra A térképkészítés folyamata és az egyes térképek fajtái

A 2. ábrából az is jól látható, hogy a térképkészítés egy folyamat, ahol a *szenzortérképből* kiindulva, elkészítjük a

geometriai térképet, amely már geometriai primitívek formájában tartalmazza az akadályok alakját. Ezt a térképet „*konfiguráljuk*” fel oly módon, hogy alkalmazni tudjuk a *pontszerű mobilrobot* modellt. A felkonfigurálás folyamata során meghatározzuk a mobilrobot köré írt kör (gömb) sugarát („*R*”), majd ezzel a sugárral megnöveljük az akadályok fizikai méreteit. Ennek a folyamatnak az eredménye a *konfigurált munkaterület* (*C-space*). Ezen a konfigurált munkaterületen megállapítjuk a *szabad*, majd *foglalt területeket* (tereket), majd elkészíthetjük a munkaterület *topológiai* (gráf) *térképét*. A gráftérkép elkészítésének több módja is ismeretes, de a legelterjedtebb eljárás, mikor a munkaterület felosztása után (általában bináris szegmentáció/négyes fa/egzakt (pontos) felosztások után) megjelöljük a szabad területek középpontjait (ezek lesznek a gráf csomópontjai), majd összekötjük őket (ezek lesznek a gráf élei) oly módon, hogy az élek ne haladjanak keresztül a konfigurált akadályokon. Amennyiben szükséges, extra pontokat is beiktathatunk.

A mobilrobot vezérlő paramétereit általános esetben (3D-s környezet) a következő vektoriális egyenlettel definiálhatjuk:

$$\mathbf{Q} = [\mathbf{X}, \mathbf{R}]$$

ahol, $\mathbf{X} = [x, y, z]^T$ a pozícióvektor, és $\mathbf{R} = [\mathbf{R}_x]_{3 \times 3} * [\mathbf{R}_y]_{3 \times 3} * [\mathbf{R}_z]_{3 \times 3}$ az orientációs mátrix, ami az orientáció elforgatását (irányát) írja le az „*x*”, „*y*”, illetve „*z*” tengelyek körüli elforgatással (térben).

Síkbeli ábrázolásnál a képlet jelentősen leegyszerűsödik. Maradva az előző jelöléseknél: $\mathbf{X} = [x_{MR}, y_{MR}]^T$, és $\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, vagyis, egyszerűbben a vezérlő paraméterek:

$$\mathbf{q} = [x_{MR}, y_{MR}, \theta]$$

Annak érdekében, hogy a mobilrobot helyesen tudja értelmezni a munkaterületet, annak szabad és foglalt mezőit, fel kell osztanunk ezt a munkateret. A legáltalánosabb *munkaterület felosztásokat* és azok jellemzőit az 1. táblázatban foglalom össze.

	Jellemzők
Négyzethálós	Egyforma négyzetek (pixel/voxel). Nagy memóriaigény. Statikus hiba. Egyszerű számolás. Átlapolásos.
Négyes fa	Nem egyforma négyzetek. Ahogy közelítünk az akadályhoz a felbontás is növekszik. Átlapolásos.
Bináris szegmentáció	Nem feltétlenül egyforma négyzetek. Átlapolás az akadály és szabad terület között.
Pontos (egzakt)	Teljesen különböző nagyságú szabad területek. Nincs átlapolás.

1. táblázat A munkaterületek felosztása és azok jellemzői

Háromdimenziós környezetben a felosztás is bonyolódik. A négyzethálós egy dimenzióval feljebb kocka felosztásra emelkedik, ekkor az alapegységet „voxeleknek” nevezik. A bináris szegmentáció értelemszerűen *nyolcasfává* (octatree) bővül.

1.2.3 Alapvető ismeretek a pályatervezésről és navigációról

A *pályatervezés* és *navigáció* egymáshoz nagyon közel álló folyamatok, gyakran keverik is őket, ezért először a kettő közti különbséget próbálom tisztázni. Ennek legegyszerűbb módja talán a következő mondat lehet: választok egy pályát „A” és „B” között (pályatervezés), majd *elnavigálok azon a pályán „A”-ból „B”-be* (navigáció).

Mondhatjuk, hogy a *pályatervezés* a navigációt megelőző folyamat, melynek során a *több lehetséges pálya közül kiválasztom az optimális végrehajtandó útvonalat*, majd a *navigációs* algoritmus segítségével *végig haladok azon az útvonalon* úgy, hogy folyamatosan ellenőrzöm pozícióm, és figyelem, hogy eltértem-e a (tervezett) pályától. A pályatervező algoritmusokat két nagy csoportra oszthatjuk: *Lokális pályatervezés*, *Globális pályatervezés*. A **globális pályatervezés** az egész környezetet figyelembe veszi, a globális térképen dolgozik, megkeresi a lehetséges útvonalakat, majd valamilyen *költségfüggvény* segítségével kiválasztja az optimálisat. A **lokális pályatervezés** iteratíván dolgozik. Halad a globális pályatervezés által megadott útvonalon, miközben figyeli a környezetet. Amennyiben valamilyen változás állt be, vagy lokálisan talál egy jobb megoldást, lokálisan újra tervezi az útvonalat. A pályatervező algoritmusokat tovább osztályozhatjuk a pályák alakja és a tervezés módszerei alapján. Ismerünk **gráf alapú** pályatervezéseket, melyek talán a legelterjedtebbek, mert a költségfüggvény számolás szempontjából (A*, Dijkstra, ...,) jó matematikai háttérrel biztosít a gráf. Közülük, csak pár példa, a teljesség igénye nélkül: láthatósági gráf, érintőleges láthatósági gráf, MAKLINK. Dinamikus környezetben elterjedten alkalmazott a **potenciálmezős** pályatervezés, melyek közül a két legismertebb módszer a VFF (*Virtual Force Field*, -Virtuális erőter-) és a VFH (*Vector Field Histogram*, - Vektormező hisztogram-). Az első pályatervező algoritmusok közé lehet sorolni a **vonal alapú** pályatervezéseket, melyek közül a legismertebbek a BUG (bogár) 1,2,3 algoritmusok, a Voronoi diagram (középvonal), vagy a geometriai primitívekből összerakott *Reeds-Shepp* algoritmus. A *globális*

pályatervezések közül a legelterjedtebbek a **hullámterjedés** módszerek (négyzetrácsos, vagy négyzethálós), illetve a **véletlenszerű** (PPL – Probabilistic Path Planning) pályatervezés. Napjainkban az ún. „finomított” pályák létrehozására használatosak a **lággyörbés** (Spline) pályatervező algoritmusok, melyek közül a legismertebbek a Lagrange féle interpolációs, majd a Beziére féle approximációs görbék, illetve a B-Spline –ok. Nem szeretném, hogy bármelyik pályatervező algoritmus és jellegzetessége lemaradjon, ezért egy táblázatban összefoglalom őket.

Kategória	Algoritmus	Előnyök	Hátrányok	Alkalmazási terület
Klasszikus	Dijkstra, A*, D*	Optimális útvonal, jól definiált térben működik	Nagy számítási igény nagy térben	Statikus térképek, globális tervezés
	Voronoi diagram	Biztonságos távolság akadályoktól	Nem mindig optimális útvonal	Szabadtéri navigáció
	Dubins, Reeds-Shepp	Nem-holonomikus járművekhez alkalmas	Csak bizonyos járműmodellekhez	Autószerű robotok
Potenciálmező alapú	Potenciálmező, VFH, VFF	Gyors, egyszerű implementáció	Lokális minimum problémák	Lokális akadályelkerülés
Bug-algoritmusok	Bug1, Bug2, VisBug	Egyszerű, kevés számítás	Nem optimális, lassú lehet	Ismeretlen környezet, egyszerű robotok

Kategória	Algoritmus	Előnyök	Hátrányok	Alkalmazási terület
Mintavételes	PRM, RRT, RRT*	Nagy dimenziós térben hatékony	Nem garantált optimális megoldás	Dinamikus, komplex környezetek
Metaheurisztikus	Genetikus, Ant Colony, PSO	Jó közelítő megoldások, adaptív	Lassú konvergencia, sok paraméter	Optimalizálás, komplex feladatok
Gépi tanulás alapú	RL, Deep RL	Tanul adaptívan, dinamikus környezetben	Nagy adat- és számítási igény	Autonóm járművek, intelligens robotok

2. táblázat Mobilrobot pályatervező eljárások összefoglalása

1.3 Szakirodalom áttekintése a témakörrel kapcsolatban

Mint ahogy azt már említettem a bevezetésben, a 2K utáni években óriási ütemet vettek fel a mobilrobotokkal kapcsolatos kutatások. Ennek megfelelően a témával kapcsolatos publikációk száma is exponenciálisan emelkedett. Az 1997-s Marsjárótól eljutottunk a „*robottársadalmakig*”, amely kifejezés robotok olyan csoportját valósítja meg, amelyek képesek kollaborálni, emberekkel/egymással egy közös cél elérése érdekében. Japánban, még jóval a Marsra-szállás előtt (1993-ban) már hasonló robot társadalmakban gondolkodtak, ami a MÁR egy továbbfejlesztett változata, ahol az egyes ágensek bizonyos autonómiával rendelkeznek, de egy közös cél érdekében tevékenykednek [Halme, 1993]. Ebben a dolgozatban ilyen mélységű *viselkedés alapú* pályatervezésbe

nem megyek bele, csak érzékeltetni szeretné a rohamos fejlődést ebben az ágazatban.

Már a bevezetésben említésre került, hogy hol is a határ a konvencionális vs. nem konvencionális pályatervezések között és minek is lesz köszönhető, hogy a pályatervezés nem konvencionális. A nem konvencionális megközelítéseket (a szerző szerint) a feladat adja meg. Nézzünk pár esettanulmányt, hogyan válik nem szokványossá egy pályatervezés.

1.3.1 Energia optimalására törekvő pályatervező eljárások

Legyen egy UAV (drón) eszközünk, melynek feladata elszállítani valamit „A”-ból „B”-be. Ez egy egyszerű feladatnak látszik, húzunk egy vonalat a két pont között és, mivel drónról van szó, legjobb esetben a vonalkövetéses módszerrel (egy egyenes mentén) eljutunk a célpozícióba.

A feladatot bonyolítja, ha van egy hegyvidékes, sziklás környezetünk, rossz időjárási körülménnyel, sok kiálló szirttel, és a bajba jutott hegymászóknak kell eljuttatni egy egészségügyi csomagot úgy, hogy a drónnak legyen energiája még a visszaútra, esetleg valamilyen műszer (újra élesztés) feltöltésére. Ekkor már ún. energiaoportimális pályát kell tervezni, ami figyelembe veszi az emelkedéseket, süllyedéseket, szélirányokat, ... etc. Ezt a megközelítést már lehet nem konvencionálisnak nevezni. Ezzel a témakörrel kapcsolatban több konferenciaközlemény is megjelent, melyek közül csak azokat emelném ki, melyek számomra, és a habilitációs tézisek témáját tekintve, relevánsak [Benotsmane, 2022], [Sabban, 2013]. Mindkét közleményben a szerzők részletesen leírják a kötelező modell meghatározásokat,

vezérlő paraméterek definiálását, sebességvektorokat, kiszámolják az emelkedéshez/süllyedéshez szükséges erőket, nyomatékokat, de ahogy átolvastam, egyik cikkben sem foglalkoznak a szerkezet (UAV) működéséhez szükséges alkatrészek energiafogyasztásával, vagyis a rendszer alapműködéséhez szükséges energiafogyasztásával. Ezt a hiányosságot küszöbölöm ki az **1. tézisemben**, azzal, hogy felállítottam egy rendszer *energiatérképét* és ezt figyelembe véve terveztem meg az *energiaoptimális* útvonalat. A másik fontos dolog az UAV-k elektromos energiatárolási képessége [Pham, 2022]. Ebben a tudományos közleményben a szerzők már részletezik a rendszer egyes összetevőinek energiafogyasztását, vizsgálják hogy, milyen energiatároló (akkumulátor, szuperkapacitás, esetleg üzemanyagcella) lenne ideális a rendszer számára, vizsgálja a hibrid hajtás lehetőségeit, de nem veszi figyelembe a külső összetevőket, pl. szélirányok.

1.3.2 Hibrid elektromos (HEV) járművek irányítási rendszerei

Az elektrifikáció egyre nagyobb teret követel magának a járműiparban. Ennek kézzelfogható eredménye az elektromos autók (EV, HEV, PHEV („*plug-in*” elektromos), BEV (tisztán elektromos), ...stb.) elterjedése a közlekedésben. Amíg a belső égésű járművek felépítését már középiskolás szinten ismertették velünk, az elektromos járművek irányításáról még most is (főleg az EV-k, HEV-k megjelenésekor) elég keveset tudtunk. Voltak próbálkozások, melyek foglalkoznak az elektromos járművekkel, de főleg itt is az energiaigény/energiafogyasztás körüli témák lettek kitérve [Desreaveaux, 2019]. Egy jó áttekintést ad a [Kebriaei, 2015],

[Fantin, 2021] összefoglalása, de nem megy részletekbe az elektromos „feltérképezést” illetőleg. Ezt a hiányt próbáltam pótolni az **5. tézisben** összefoglalt cikkemben, ahol az Óbudai Egyetem első elektromos autójának (*Nissan Leaf*) készítettem el (mértem ki) az elektronikus alkatrészek (szenzorok, aktuátorok, vezérlő egységek, modulok) közötti kapcsolat elektronikus hálózatát. Ezt a hálózatot kihasználva elkészítettem az elektronikus irányítás hierarchikus rendszerét. Ehhez a témakörhöz tartoznak még további cikkeim [T1], [T2].

1.3.3 Multi-ágensű dinamikus környezetben működő pályatervező és feltérképező algoritmusokkal kapcsolatos kutatások

A multi-ágensű mobilrobot rendszerek olyan együttműködő robotokból állnak, amelyek közös cél érdekében osztják meg egymással az információt és koordinálják a mozgásukat. Az ilyen rendszerek előnye a *skálázhatóság*, a *redundancia* és a feladatok párhuzamos végrehajtása, ami különösen hasznos nagy kiterjedésű vagy dinamikus környezetekben. A legújabb kutatások, a feltérképezés kapcsán előnyben részesítik a SLAM módszereket és „hanyagolják” a potenciálmezős módszert, pedig dinamikus környezetre (saját megállapítás) szerintem nagyon megfelel. Igaz, a pontossággal kapcsolatban vannak némi fenntartásaim. A [Deng, 2025] hasonló a szerző módszeréhez, csak neurális hálós SLAM módszert alkalmaz, míg a szerző GA módszert és potenciálmező segítségével készíti a térképet. Rátérve a Spline-os (lágýgörbés) módszerre, bizonyosfokú pályaoptimalizálásról is beszélhetünk. A navigáció során a robotoknak sima, és dinamikailag megvalósítható útvonalat kell követniük. A lágýgörbék

ideálisak erre, mert: folytonosságot (minimum C^2) és simaságot biztosítanak; paramétereizhetők; jól kombinálhatók optimalizációs algoritmusokkal (GA, NN). Azzal, hogy a B-spline kerüli a dinamikus mozgásokat, bizonyos fokú energiamegtakarítást is elérhetünk [Kai-Tang, 2025]. Másik dolog, hogy a B-spline görbék illesztése is jól véghezvihető egy bizonyos fokig linearizált környezetre [Eshtehardian, 2023]. Hasonló tématerületeket takar le a **2., 3., és 4. tézis**.

1.3.4 VR alapú technológia alkalmazása az irányítási rendszerekben

A virtuális valóság (VR) technológia egyre elterjedtebbé vált a technológiák és a kommunikáció fejlődésével. Nem kell az alkalmazásokkal kapcsolatban messzire menni, elég csak követni a háborús cselekményeket. A VR integrációja a mobilrobotok vezérlésébe az utóbbi években jelentős fejlődést hozott az ipari és kutatási alkalmazásokban [Galarza, 2023]. Ezen teleoperációs rendszerek célja, hogy *immerzív* (3D-s valós időben való látás), *intuitív és biztonságos* ember–robot interakciót biztosítsanak, különösen távoli vagy veszélyes környezetekben [Alsamhi, 2023]. Ez utóbbi mondat inspirált, hogy mélyebben megismerjem ezeknek a rendszereknek a működését, és foglalkozzam a témával. Hasonló témájú publikációim eredményeit a **6. tézisben** foglaltam össze.

2. Új tudományos eredmények

Kutatómunkámat az 1.1 fejezetben megfogalmazott motiváció és célkitűzések alapján mutatom be. Először összeállítom a tézis előzményeit, majd megfogalmazom a tézist és végül a tézis részletes magyarázatát és az eredmények

rövid összefoglalását prezentálom. A téziseknél megjelölöm a tézishez/altézisekhez tartozó legfontosabb publikációkat is.

2.1. UAV-k energiaoptimális pályatervezésére irányuló navigációs algoritmus kifejlesztése

Ez a kutatás egy többéves nemzetközi projekt keretén belül indult⁶, melyben a *DGU* (Donja Gorica University, Montenegro) és a *MoDrone*, D.o.o. (Podgorica, Montenegro) mellett, az *Óbudai Egyetem* vett részt. A projekt, részünkre eső célja energiatakarékos pályatervezés életmentő drónok (UAV-k) számára. Az ötlet a montenegrói kollégáktól (hegymászóktól, életmentőktől) érkezett. Nagyon sok olyan helyen rekedtek turisták (sportolók) megszorult helyzetben, amelyek nehezen megközelíthetőek. Itt a gyors segítség, beavatkozás életmentő lehet. Ezekre a helyzetekre kerestük a megoldást, ami a végén innovációs beadvány is lett Montenegróban, a fentebb említett cég részéről. Az ezzel kapcsolatos eredmények az egyes tézisben kerültek összefoglalásra.

⁶ KTE, Smart4All, Grant Agreement and Innovation Action No: 872614, *An AI-based algorithm for optimized 3D drone flight path planning, DELTA*.

1. tézis: Az alábbi, UAV-ok számára szolgáló energia-optimalizált 3D-s pályatervezési eljárás a következő lépésekből áll

- első lépése az „energia térkép” létrehozása
- második lépés a 3D munkatér partícionálása
- harmadik lépés és a gráf-alapú reprezentáció létrehozása
- negyedik lépés a módosított A* algoritmus alkalmazása, amely a szélirány és sebesség befolyásolásával nemcsak az élek, hanem a csomópontok súlyozását is figyelembe veszi

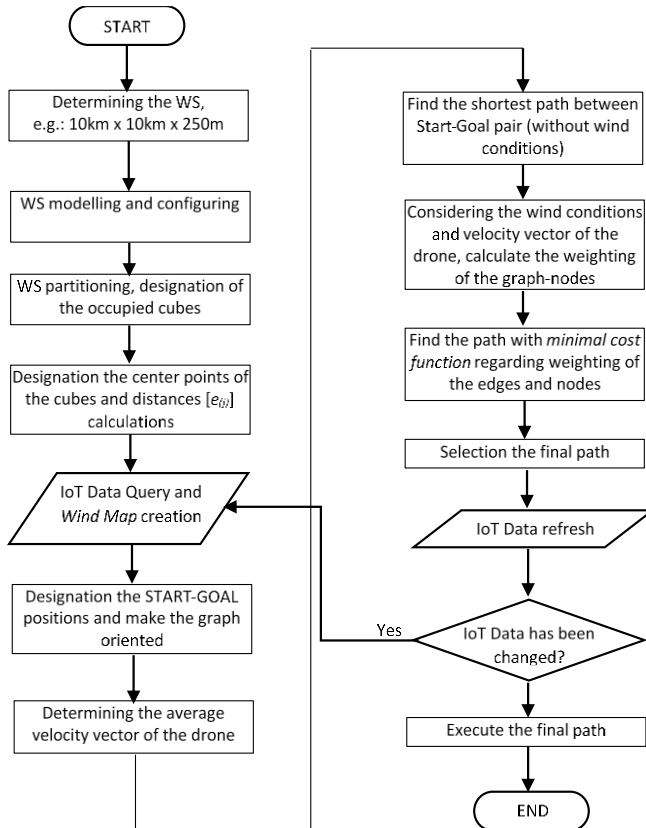
(MDPI, Applied Sciences 2024, [1])

2.1.1. A tézis rövid kifejtése

Az alábbi, UAV-k számára szolgáló energia-optimalizált 3D pályatervezési módszer az következő lépésekből áll:

- mivel a projekt egy adott UAV-re vonatkozott, amit a projektben résztvevő cég biztosított, először elkészítettem az adott drón energiatérképét. Itt kétféle állapotot különböztettem meg, melyeket különböző színekkel is jelöltem cikkemben
- második lépésként eljárást dolgoztam ki, mely elkészíti a munkaterület 3D-s felosztását.
- harmadik lépésben létrehoztam a munkaterület gráf alapú reprezentációját, természetesen csak a szabad területekre vonatkoztatva (sztatikus munkatérrel dolgoztam, nem vettem figyelembe a mozgó akadályokat)
- kiszámoltam a különböző szélirányokra (ellenszél, hátszél, oldalszél) vonatkozó energiaigényeket
- egy általam módosított A* algoritmus segítségével (figyelembe veszi a szélirányt -IoT adatok frissítése-,

sebességet, és még a csomópontokat is súlyozza) meghatároztam az energiaoptimális útvonalat.



3. ábra Az előzőekben leírt algoritmus részletes folyamatábrája

2.1.2. Eredmények és összefoglalás

A szimulációk három különböző szélhelyzetben (hátszél, oldalszél, ellenszél) igazolták, hogy az algoritmus képes az energiaigény minimalizálására: ellenszél esetén a „sailing” manőverek csökkentették a fogyasztást, míg hátszélben a

repülési idő és energiafelhasználás jelentősen mérséklődött. A tézis dokumentálása megtalálható [1].

2.2. Multiágens alapon történő potenciálmezős térképkészítés

A multi-ágensű környezetben akkor lehet optimálisan és energiahatékonyan dolgozni, ha működik az effektív munkamegosztás. Ebben a kutatásban egy olyan GA alapú, 2 fitnessfüggvénnyel működő térképkészítő algoritmust dolgoztam ki, mely az ágensek által meglátogatott területek átlapoltságát igyekszik a minimálisra csökkenteni. Vagyis, az ágensek ne menjenek olyan területekre, amit egy másik ágens már nagy valószínűséggel feltérképezett.

2.tézis: Egy többügynökös mobilrobot-rendszer esetében a szenzoradatokból származtatott repulzív erők alapján létrehozott globális potenciálmező és genetikus algoritmus (GA) alkalmazásával létrehozható egy olyan mozgásstratégia, amely minimalizálja az ütközési kockázatot és maximalizálja a lefedettséget. Az algoritmus fő lépései:

- 1. Érzékelés és lokális kódolás**
- 2. Lokális → globális potenciálmező frissítése**
- 3. Önszervező térképezés**
- 4. GA – jelötlépések optimalizálása (fitness-1, fitness-2)**
- 5. Végrehajtás és ciklus**
(ICCC 2013 [2], ICC 2003 [3], IAESTED 2007 [4])

2.2.1. A tézis rövid kifejtése

A multi-ágensű platformok működésével és modellezésükkel, még PhD tanulmányaim alatt kezdtem foglalkozni. Ez látható is a publikációs listában, a [3]) –ban

megjelent konferenciákból. Itt még „csak” arra voltam kíváncsi, hogy működik-e a multi ágensű platform és egyáltalán modellezhető, vagy használható-e valamire. Az ezzel kapcsolatos további kutatás a GA paraméterek optimalizálásával foglalkozik [2]. Mindezek mellett le kell szögezni, hogy a PhD értekezésemben ezzel (ezekkel) a témákkal nem foglalkoztam.

A többügynökös potenciálmező építése GA segítségével – algoritmus leírása:

1. Érzékelés és lokális kódolás

Minden robot (ágens) 8/16 ultrahangos szenzorral rövid/hossz hatótávon szektoronként binárisan kódolja a szomszédság foglaltságát (robot/akadály), és távolságbecslést képez a közeli, még nem látogatott pozíciókra

2. Lokális → globális potenciálmező frissítése

A robot a mért adatokból lokális repulzív potenciált számít, majd a több ügynök által megosztott globális potenciáltérképet (GPM) inkrementálisan frissíti; az ütközéssel jelölt helyeket eldobja (láthatósági feltétel). Az adatok mátrix-struktúrában (WS, VE, U) tárolódnak, súlyozással feloldva az ugyanarra a cellára eső több becslést.

3. Önszervező térképépítés

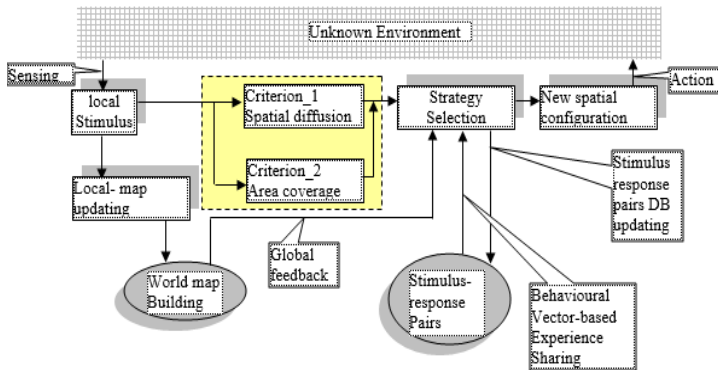
A robotok folyamatosan szinkronizálják a GPM-et és a bejárat/bejáratlan területek naplóját; a saját és szomszédok pozícióját (R_t) rögzítik, hogy a potenciálmező és a lefedettség térképe konzisztens maradjon.

4. Genetikus algoritmus – jelölt lépések optimalizálása

A GA kromoszómája a következő lépés(ek) diszkrét irányát/pozícióját kódolja; fitness: (i) új területek feltárása + ismétlés kerülése + ütközéskockázat minimalizálása, (ii) térbeli diffúzió (ügynök–ügynök távolság küszöb felett), (iii) lefedettség maximalizálása. Szelekció–keresztelés–mutáció után a legjobb stratégia kerül végrehajtásra.

5. Végrehajtás és ciklus

A kiválasztott mozgás végrehajtása → pozíció/szenzor frissítése → **visszatérés az 1. lépéshez** mindaddig, amíg a lefedettségi cél teljesül vagy a költségfüggvény (fitness) nem javul érdemben. Kutatásaimban egy centralizált rendszert használok, ahol az egyes homogén, nem kommunikáló (azonos szenzorrendszerrel felszerelt) ágensek egy közös adatbázisba dolgoznak, amely adatbázis feldolgozásával fel lehet építeni az



4. ábra A rendszer működésének ábrázolása

adott terület potenciáltérképét [3]. Ezek után, ennek a továbbfejlesztése következett, ami a viselkedés-alapú potenciálmező építése volt, ahol az eddig használt két fitnessfüggvényhez még csatlakozott egy viselkedés adatbázis, melyből a fitness függvények alapján kerültek kiválasztásra a megfelelő mozgásstratégiák (elemi viselkedések) [4].

2.2.2. Eredmények és összefoglalás

A módszer MATLAB környezetben került implementálásra különböző számú ágensek (1/3/6 ágens) és az ágenseken különböző számú szenzorok (8/16 szenzor) felhasználásával. A GA-alapú mozgásstratégia a potenciálmező statisztikai

jellemzői alapján választotta ki a következő lépést, minimalizálva az ütközési kockázatot és maximalizálva a lefedettséget. A szimulációk igazolták, hogy 40 ciklus alatt a rendszer képes volt a munkaterület teljes feltérképezésére, miközben a mozgások dinamikai optimalizálása csökkentette az redundáns lépéseket.

2.3. Lágygörbe (*Spline*) alapú optimális pályatervezés mobil ágensek számára

Az optimális pályatervezés amely mellőzi a hirtelen gyorsulásokat/lassulásokat, a lágygörbés módszerek segítségével ma már egyre elterjedtebbé váltak. Ismeretes, hogy a „spline-ok” paraméterezhetők, így a sebességeket is be tudjuk kalkulálni a görbe egyes pontjaiban. Itt a problémát a görbe „kigenerálása” okozza, méghozzá akkor, ha kritikus (szűk területeken) halad az útvonal. Ebben a kutatási projektben egy elsődleges optimalizálást készítettem egy gráf alapú útvonal megtervezésével, majd erre az útvonalra generáltam ki a lágygörbét (*Spline* formájában) a hirtelen gyorsulások / lassulások elkerülésének érdekében.

3. tézis: Az alábbi eljárással elkészíthető egy olyan *Spline* alapú pályatervezési eljárás, amely a navigációs terület optimalizált topológiai gráfjára illesztett lágygörbékkel biztosítja a pálya C^2 folytonosságát és annak megfelelő dinamikai tulajdonságát. Az eljárás egyszerűsített lépései:

- Szabadterületek meghatározása
- Gráf topológia felállítása az adott feltételek alapján (Voronoi, visibility)
- *Start-Cél* pontok meghatározása; legrövidebb útvonalak meghatározása *Start-Gráf/Cél-Gráf* pontok között
- Költségfüggvények kiszámolása a lehetséges gráf-útvonalakra
- Az interpolációs lágygörbe kigenerálása a kiválasztott gráfcsomópontokra

(Acta 2014 [5])

Felhívom a figyelmet a *Spline* vs *B-spline* görbék közötti különbségre. Az egyik legfontosabb különbség hogy a *Spline* görbék *interpolációs* tulajdonsággal, míg a *B-spline* (bázis spline) *approximációs* tulajdonsággal rendelkeznek. Felújított kutatásaimban már harmadfokú (*C-spline*) görbéket alkalmaztam a következő feltételekkel:

1. a harmadfokú polinom:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3;$$

2. Interpolációs feltétel:

$$S_i(x_i) = y_i \text{ és } S_i(x_{i+1}) = y_{i+1};$$

(áthaladjon a pontokon)

3. Folytonossági feltétel: (hogy az érintők (S'_i) és a görbületek (S''_i) megegyeznek a pontok találkozásánál.)

$$S'_{i-1}(x_i) = S'_i(x_i) \text{ és } S''_{i-1}(x_i) = S''_i(x_i);$$

2.3.1. A t  zis r  vid kifejt  se

Ez a kutat  som az el  z   multi-  gens   kutat  saimnak a folytat  sa oly m  don, hogy a felt  rk  pezett munkater  leteket part  cion  ltam egy   ltalam⁷ „egzakt (pontos) dekompoz  ci  nak” nevezett m  dszerrel, majd erre a felosztott munkater  letre kigener  ltam a legbiztons  gosabb topol  gi  t (gr  fot). Val  j  ban, egy *Voronoi-diagramon* alapul   gr  f s  lyoz  s  val az al  bbi elj  r  si l  p  sekb  l   ll   m  dszerrel olyan p  lyag  rb  ket alak  tottam ki, amelyek kiel  g  tik a C^2 folytonoss  gi felt  telt   s a dinamikai korl  tokat:

- a kiv  lasztott   tvonalra *spline* g  rb  t illesztettem
- megvizsg  ltam k  tf  le lehet  s  get:
 -    a „biztons  gos” (k  z  p  vonalon)   tvonalat
 -    majd a l  that  s  gi gr  fra illesztett   n. „gyors”   tvonalat.

a meghat  rozott navig  ci  s hiba alapj  n az egyes   tvonalak komplexit  sa a k  vetkez   egyenletekkel sz  molhat  k (1), (2):

$$Cf_{safety} = \underbrace{\min(w_{(g)}SSG_{(g)}) + \min(w_{(h)}CCG_{(h)})}_{\text{the shortest segments to the "centerline"}} + \min\left(\sum_i Ws_{(i)} + \sum_j Wn_{(j)}\right) \Bigg|_{\substack{SG(g) \rightarrow SG(\min) \\ CG(h) \rightarrow CG(\min)}} + \min(Cf_{g \times h \times k}) \quad (1)$$

$$Cf_{fast} = \min(Cf_{g \times h \times k}) + \underbrace{w_{(g)}SSG_{(\min)} + w_{(h)}CCG_{(\min)}}_{\text{Weighting of the segments between the visible nodes}} \quad (2)$$

ahol az „SSG” / „CCG” a kiindul  -/c  lpoz  ci  t a k  z  p  vonallal   sszek  t   szegmensek („g”, „h” -indexel  sekkel); „ $W_{s/n}$ ” a

⁷ angol ford  t  sb  l

szegmensek(s)/csomópontok(n) súlyozásai („ i ”, „ j ” - indexeléssel). „ C^p ” a számolt költségfüggvények.

2.3.2. Eredmények és összefoglalás

A pályatervezés a *Voronoi*-diagramon alapuló gráf súlyozásával kezdődik, majd a kiválasztott útvonalra lágygörbét illesztnek az előzőekben leírt feltételeknek megfelelően. A szimulációk során két útvonalat vizsgáltam: „*biztonságos*” (középvonalon) és „*gyors*” (láthatósági gráf alapján). Az eredmények szerint a biztonságos útvonal hossza 120,97 [m], végrehajtási ideje 65,3 [s], míg a gyors útvonal 110,35 [m] hosszú és 61,5 [s] alatt teljesíthető. Mindkét esetben a görbék kielégítették a C^2 folytonossági feltételt és a dinamikai korlátokat.

Multi-ágensű környezetben a munkaterület szegmentációjából (particionálásából) kiindulva az egyes szegmensek, úgymond „*szemafor*” alapon működnek. Azaz pirossal vannak az ágens(ek) által elfoglalt szegmens(ek) jelölve és zölddel a szabad szegmensek. Ennek megfelelően, dinamikusán történik a pályatervezés is.

2.4. Multiágensű alapon történő lokalizációs hibaszámításos modell dinamikus környezetben

Ezen kutatásom szintén multi-ágensű környezetre vonatkozik, csak itt kihasználtam a „*Unity*” adta programozói környezet párhuzamosítási lehetőségeit. A lokalizáció, marker alapú (*trilaterációs*) módszeren alapul, ahol az egyes hibaterületek nagyságát különböző színekkel jelöltem. A kutatás célja, hogy ismert lokalizációs berendezéssel felszerelt ágensek, egy dinamikusán változó környezetben (ahol a mozgó akadályok ágensek is lehetnek), minden pillanatban tudja a lokalizációs bizonytalanságát. A kérdés, mire lehet ez jó? Ez a módszer alapot ad egy lokalizációs hiba alapú pályatervezéshez, pl. mikor adott, hogy a környezetben csak

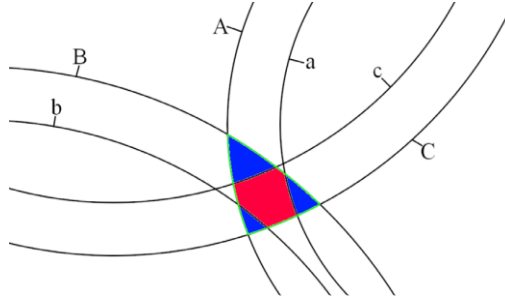
egy adott $\min(\%)$ -os lokalizációs hibával lehet közlekedni. Itt, szintén dinamikus pályatervezésről beszélünk a pillanatnyi lokalizációs bizonytalanság figyelembevételével.

4. tézis: Egy valós idejű, párhuzamosított algoritmus segítségével, amely dinamikus, multiágensű környezetben mozgó mobilrobotok (ágensek) lokalizációs hibáit veszi figyelembe, előállítható egy dinamikus változó környezet mesterséges hibatérképe (*Artificial Error Map – AEM*) színskódolt vagy 3D-s megjelenítéssel.

(ICCC 2022 [6])

2.4.1. A tézis rövid kifejtése

A lokalizációs hiba alapú dinamikus pályatervezésnél, elsősorban a környezetet és a hibát kell modellezni. Ebben a kutatásban ennek alapjait fektettem le. A környezeti modellben pont-reprezentált ágenseket és természetesen ennek megfelelően konfigurált munkateret (*C-space*). A lokalizációnál a jól ismert *tri-laterációs* módszert alkalmaztam, ahol a mérési bizonytalanságot ($\Delta = |A - a|$) az egyes körívek közti különbség képviselte.



5. ábra A hibaterület ábrázolása

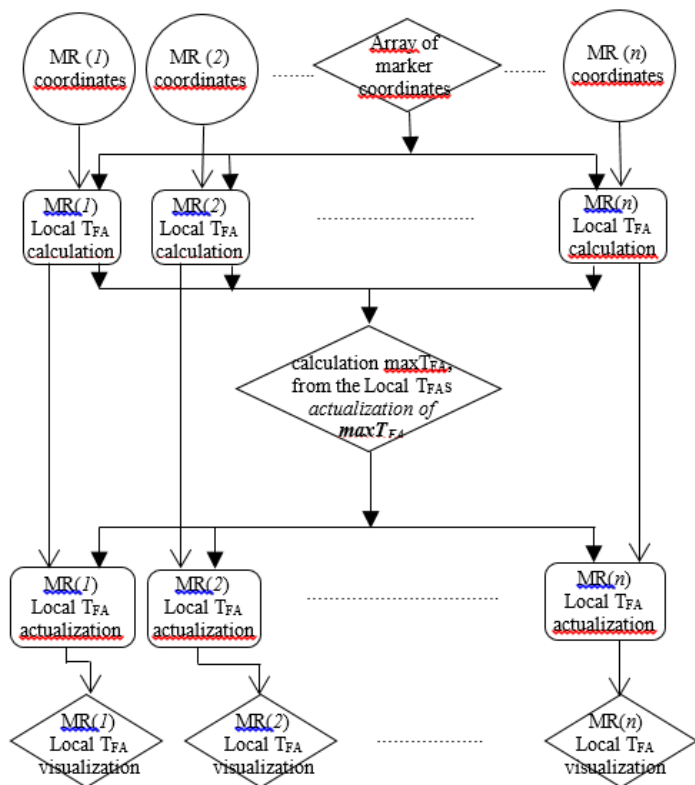
A három körgyűrű metszete (lásd piros szín) adja a pillanatnyi hibaterületet. Ez képletesen felírható a következő kifejezéssel:

$$T_{FA} = T[A, B, C] - (T[A, B, c] + T[A, b, C] + T[a, B, C])$$

$$T[\bar{X}] = \pi * (l_{(i)} \mp \frac{\Delta}{2})^2;$$

$$\bar{X} = [A, B, C, a, b, c];$$

ahol „ $l_{(i)}$ ” az egyes markertől mért távolságok. Természetesen nem mindegy az ágenshez képesti marker-konstelláció, de ezekkel már néhány korábbi kutatásomban foglalkoztam [Nagy, 2002], [Nagy, 2004]. Jelen kutatásnak a lényege a párhuzamosított algoritmus, amit a következő blokkábrával szemléltetek.

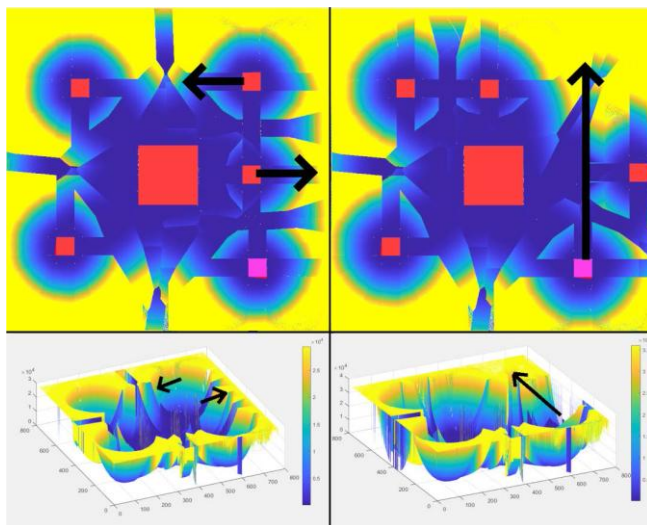


6. ábra A dinamikus hibamező elkészítésére szolgáló párhuzamosított algoritmus blokkábrája

2.4.2. Eredmények és összefoglalás

A kutatás célja egy olyan szimulációs rendszer létrehozása volt, amely képes valós időben meghatározni és megjeleníteni a lokalizációs bizonytalanságokat dinamikus, többügynökös környezetben. Az algoritmus kliens–szerver architektúrán és párhuzamos feldolgozáson alapul, Unity platformon valósítva meg, C# nyelven, GPU-gyorsítással. Az eredmények igazolták, hogy a rendszer képes valós idejű hibakép generálására,

színkódolt és Z-tengelyen ábrázolt formában, miközben a párhuzamosítás jelentősen csökkenti a számítási időt. A módszer alkalmas dinamikus útvonaltervezés támogatására, több robot együttműködésének vizsgálatára és valós környezetek szimulációjára.



7. ábra A színkódolt, illetve Z-tengelyen ábrázolt dinamikus hibamező pillanatfelvétele, ahol az egyes ágensek orientációja a nyilakkal van ábrázolva

2.5. Elektromos járművek (EV) szenzor és vezérlőhálózatának feltérképezése és azok egymásra épülő modellezése

Mikor az Óbudai Egyetemen megjelent az első teljesen villamos meghajtással működő jármű (*Nissan Leaf Z0*), egy kutatási projekt keretén belül (*EFOP-3.6.2-16-2017-00016*) feltérképeztük annak teljes elektromos hálózatát. Ez volt a kiindulás a kutatáshoz. Az egyes elemek feltérképezése után

blokkosítottam az azonos tulajdonsággal rendelkező összetevőket, majd hierarchikus rendszerbe rendeztem őket, természetesen meghagyva a fizikai kapcsolatokat az egyes blokkok között. A hierarchia felső csúcsát a komplexebb blokkok alkották többpontos összekötéssel, míg a legalján a PtP (*Point to Point*) kapcsolatok és eszközök helyezkednek el. A kutatás célja egy hipermátrix (mátrixtömbök) rendszer felállítása, ahol az egyes mátrixokon belüli összekötések és az egyes mátrixok közötti összekötések figyelésével egy komplexebb diagnosztikai rendszert tudunk felállítani.

5. tézis: Az alábbi eljárás biztosítja az elektromos járművek (EV/HEV) szenzor-, vezérlő- és kommunikációs rendszerének egységes, hierarchikus és gráf-alapú modellezését, lehetővé teszi a teljes járműirányítási architektúra formális leírását, a jelutak és kapcsolati struktúrák matematikai reprezentációját, valamint a rendszer megbízhatóságának mátrix-szintű vizsgálatát. Az eljárás lépései az alábbi pontok szerintiek:

5.1 az EV-k/HEV-k teljes szenzoros és vezérlő rendszerére alkalmazható háromszintű hierarchikus architektúr elkészítése (szenzor/aktuátor – egységek (unitok) – vezérlőmodulok), amely a kommunikációs irányok (uni- és bi-direkcionális kapcsolatok) explicitté tételével átláthatóvá teszi a jármű teljes irányítási struktúráját.

5.2 az EV-k/HEV-k szenzor-vezérlő hálózatának gráf-modelljének elkészítése, amely illeszkedési (incidencia) mátrixokkal, szomszédsági (adjacencia) mátrixokkal és rétegzett (hipermátrix típusú) struktúrákkal képes kezelni az alrendszerek átfedéseit, az irányítottságot és a jelutak egyértelmű matematikai leképezéseit.

5.3 A modul-szintű összeköttetés-mátrix (connectivity matrix) kiterjesztése egy megbízhatósági (reliability) mátrixszá, amelyben az élek súlyozott értékei a kapcsolat megbízhatóságát, hibaterjedési kockázatát és diagnosztikai lefedettségét reprezentálják.

(Acta 2021 [7], SACI 2021 [8], IOP 2022 [9], eBook [10])

2.5.1. A tézisek rövid kifejtése

5.1 lépés kifejtése - *Hierarchikus rendszerarchitektúra kialakítása*

Ebben az altézisben a kiemelendő a hierarchikus architektúra létrehozása. Ez a rendszerszintű átjárhatóság nyomon követését biztosítja. A „szenzor/aktuátor”-„egység(unit)”-„modul”, hármas felosztás egyértelműsíti a jelutakat és az irányítástechnikai felelősségi pontokat: az alsó szint tipikusan egyirányú (eszköz→egység) jeleket ad, a közép és felső szintek között pedig kétirányú (diagnosztikát és parancsot is tartalmazó) kapcsolatok futnak. Ez a felbontás lehetővé teszi a funkcionális blokkdiagramok konzisztens összeillesztését a jármű valós modul-struktúrájával (VCM, BCM, IPDM, PDM, Combination Meter), és megkönnyíti a hibaterjedés követését. A hierarchikus térkép és az alrendszer-szintek közötti irányítástechnikai és kommunikációs szerepek dokumentáltak; a Nissan Leaf Z0 példán részletesen azonosított modulok/egységek igazolják az elvet [7], [8].

5.2 lépés kifejtése - *Gráf- és mátrix alapú formális modell megalkotása, kommunikációs rétegek architektúra-konzisztencia és diagnosztikai javaslatok*

Az EV irányítási hálózatának gráf-modellje (csomópontok: modulok/egységek/szenzorok; élek: kapcsolatok) **illeszkedési**

(incidenncia) és **szomszédsági** (adjacencia) **mátrixokkal** formálisan reprezentálható. Az illeszkedési-mátrix kezeli az irányítást és a többes (párhuzamos) kapcsolatok megkülönböztetését, míg az „adjacencia” a szomszédsági viszonyok gyors áttekintését adja. Az átfedő részrendszerekre **rétegezett (hypermátrix) megoldás** javasolható. Ez a formalizmus alapot teremt megbízhatósági, útvonal-/hálózati, ill. diagnosztikai elemzésekhez.

A **kommunikációk jellemzése**: A **LIN** a hierarchia alsó szintjeinek költségghatékony, egyvezeték master-slave összeköttetése (kapcsolók, kis aktuátorok), míg a **CAN** a közép–felső szintek robusztus, kétirányú gerince (egység–modul és modul–modul). A kommunikációs réteg explicit beépítése a gráf/mátrix modellbe lehetővé teszi a **jelút-diagnosztikát** és a hibajellemzők (pl. szakadás, rövidzár, tápfesz-anomália) gyorsabb azonosítását, illetve a jelátvitel irányának egyértelmű követését. A LIN és CAN szerepe és tipikus hibái, valamint a CAN-jelzések modulonkénti T/R (transmit/receive) mátrixa konkrét példákkal szerepel; ez közvetlenül illeszthető a gráf- és incidencia-modellekhez [7], [9].

5.3 lépés kifejtése - Összeköttetés-mátrix (connectivity) → megbízhatósági mátrix (reliability) előkészítése

Az összeköttetés (mátrix) ebben az értelemben magában foglalja mindkét (szomszédsági/illeszkedési) mátrixot, attól függően, az adott helyzetben melyik használata a releváns.

A modul-szintű **összeköttetés-mátrix** (connectivity-matrix) kiterjeszthető **megbízhatósági mátrixszá**, ahol az élek elemei a kapcsolat minőségét/hatásosságát (pl. hibatűrés, hibaarány, diagnosztikai lefedettség) reprezentálják. Ez a lépés hidat képez a hálózati modell és a **rendszer-szintű megbízhatósági**

számítások között (akár FCM-ihletésű súlyozással), és utat nyit a komponens-/alrendszer-szintű redundancia és derating szabályok formalizált értékeléséhez. A módszer lehetővé teszi a járműirányítási hálózat rendszer-szintű, formális megbízhatósági analízisét, beleértve a redundanciák, kritikus pontok és hibatípusok mátrixszintű azonosítását. Az altézis dokumentált a látható cikkben és könyvfejezetben. A cikk kifejezetten javaslatot ad a *connectivity-mátrix* felépítésére és annak jövőbeli megbízhatósági (reliability) mátrixszá alakítására; a könyvfejezet matematikai apparátusa biztosítja a formális átjárást [9], [10].

2.5.2. Eredmények és összefoglalás

Kutatásaim során átfogó és egységes módszert dolgoztam ki az elektromos járművek (EV/HEV) szenzor- és vezérlőhálózatának feltárására, strukturálására és formális matematikai modellezésére. A módszer újdonsága abban áll, hogy a jármű teljes irányítási architektúráját — a fizikai eszközszinttől a felső vezérlőmodulokig — hierarchikus, gráf-alapú és mátrixreprezentációval támogatott, egységes modellbe integrálja.

Első lépésként részletesen feltérképeztem egy egyetemi tulajdonú, teljesen elektromos jármű (Nissan Leaf Z0) valós szenzor- és vezérlőhálózatát, amely több mint ötven eszközből (modulok, egységek, intelligens és egyszerű szenzorok, aktuátorok) álló komplex rendszer. A hálózat elemeit háromszintű hierarchiába rendeztem (szenzor/aktuátor – egység – modul), amely világosan elkülöníti a jelgeneráló, jel-előkészítő és járműszintű vezérlési funkciókat. Ez a struktúra biztosítja, hogy az elektromos jármű irányítási feladatai rendszerszinten áttekinthetők és elemezhetők legyenek.

A hierarchikus mappinget követően gráf-modellt hoztam létre, amely a rendszer összes elemét csomópontként, a közöttük lévő irányított vagy kétirányú kapcsolatokat pedig élekként reprezentálja. A gráfból kiindulva megalkottam az EV vezérlőrendszerének incidencia-mátrixát és adjacencia-mátrixát, amelyek formális matematikai eszközt adnak a hálózat szerkezetének leírására. A modell kiterjed a jelátviteli irányokra, valamint az alrendszerek közötti átfedések kezelésére is. Utóbbi érdekében bevezettem egy rétegzett (hypermátrix) reprezentációt, amely lehetővé teszi a több alrendszerhez egyszerre tartozó egységek és kapcsolatok matematikai elkülönítését és elemzését.

A kommunikációs hálózatok — különösen a LIN és CAN — részletes vizsgálatával integráltam a tényleges T/R (transmit/receive) szerepeket és jelutakat a gráfos és mátrixos modellbe. Ennek eredményeként a kommunikációs hibák (szakadás, rövidzár, hibás jelátvitel, modul-egység inkompatibilitás) már mátrixalapú diagnosztikai módszerekkel azonosíthatók és elemezhetők. A modell képes visszavezetni egy hibás hálózati viselkedést a megfelelő jelútra vagy eszközre.

A kutatás záró lépéseként javaslatot tettem a modul-szintű kapcsolati mátrix kiterjesztésére megbízhatósági mátrixszá, amelyben az élekhez hozzárendelhetők a kapcsolat megbízhatósági együtthatói. Ez a megközelítés új lehetőséget nyit a járműirányítási hálózat hibaterjedési modelljeinek, valamint a rendszer-szintű reliability-vizsgálatainak formalizált, matematikai értékelésére.

Összegzésként megállapítható, hogy a kidolgozott módszer alkalmas:

- az EV-k teljes szenzor- és vezérlőhálózatának szisztematikus feltárására,

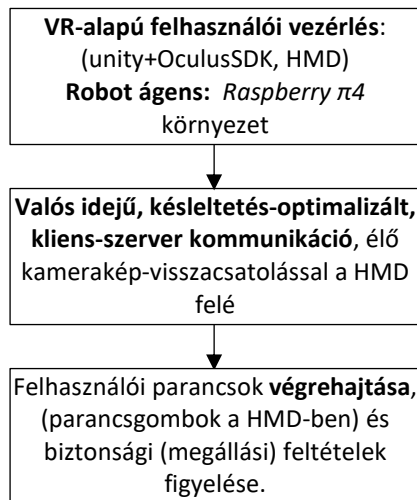
- a fizikai és logikai hálózat összekapcsolt, hierarchikus modelljének létrehozására,
- a jelutak és kommunikációs kapcsolatok formalizált elemzésére,
- az összefüggések és hibaterjedési irányok azonosítására,
- megbízhatósági mátrixok és hálózati modellek alapjának megteremtésére.

A módszer hozzájárul az elektromos járművek irányító rendszereinek mélyebb megértéséhez, hatékonyabb diagnosztikájához és jövőbeli tervezési-, valamint megbízhatósági vizsgálataihoz, egyesítve a járműmérnöki gyakorlatot a gráfelméleti és mátrixos rendszermodellezés eszköztárával.

2.6. A virtuális valóság (VR) bevonása a járművek irányításába

Már nem csak a kutatásokban, de a valós életben is egyre inkább teret hódít magának a virtuális és a kiterjesztett valóság (VR, AR). Ezzel kapcsolatban, szintén egy projekt keretén belül, kialakítottam egy VR alapú távirányítási rendszert mobilrobotokhoz, mely mozoghat olyan környezetben, ahol az emberi jelenlét lehetetlen lenne (mérgező gázos vagy rádióaktív környezetek).

6. **tézis:** Az alábbi egyszerűsített folyamatára szerint kifejleszthető egy valós idejű, VR-alapú távirányítási rendszer a differenciálhajtású ágensekhez, amely rendszer alacsony költségű hardveren, késleltetés-optimalizált, kliens-szerver architektúrával működtethető.



(Acta 2024 [11])

2.6.1. A tézis rövid kifejtése

A virtuális világgal kapcsolatos kutatásaimat valójában egy kamerás drón, arcfelismerő szoftverrel való felszerelése kapcsán, egy kijelölt arc követése volt (SISY 2022). Itt valójában a drón követte a kijelölt célpontot. Ezt fordítottam meg a VR alapú mobil-ágens irányítással, ahol a kamera a mobil ágensen volt elhelyezve, miközben az irányító gombok pedig a *Unity platform* segítségével a VR világban „látszanak”.

A modern ICT (*Information and Communication Technologies*) eszközök, különösen a virtuális és kevert valóság (AR) technológiák, új lehetőséget kínálnak IoT-eszközök valós idejű vezérlésére és felügyeletére. A VR-headsetek (HMD-k) elterjedése a technológiai fejlődésnek és az árak csökkenésének köszönhetően lehetővé teszi, hogy széles körben, költséghatékonyan használhatók legyenek távvezérlési feladatokra. A kutatás egy differenciál hajtású mobil robotot mutat be, amelyet valós időben lehet távolról irányítani VR-környezeten keresztül. A vezérlés *Unity 3D* motorral (*SW motor*) készített virtuális vezérlőfelületen történik. A VR-felület különálló virtuális objektumokat tartalmaz: *kormánykerék*, *irányváltó*, *kijelző-felület* és *vészleállító* gomb – ezek kézmozdulatokkal (3D-s egér) vagy kontrollerekkel is működtethetők.

Az **Oculus Quest** külső kamerái kézmozdulat-felismerést biztosítanak, így a felhasználó ezzel is képes irányítani az ágenszt. Az ágensen elhelyezett „*Raspberry Pi 4*” három alapeladatot lát el:

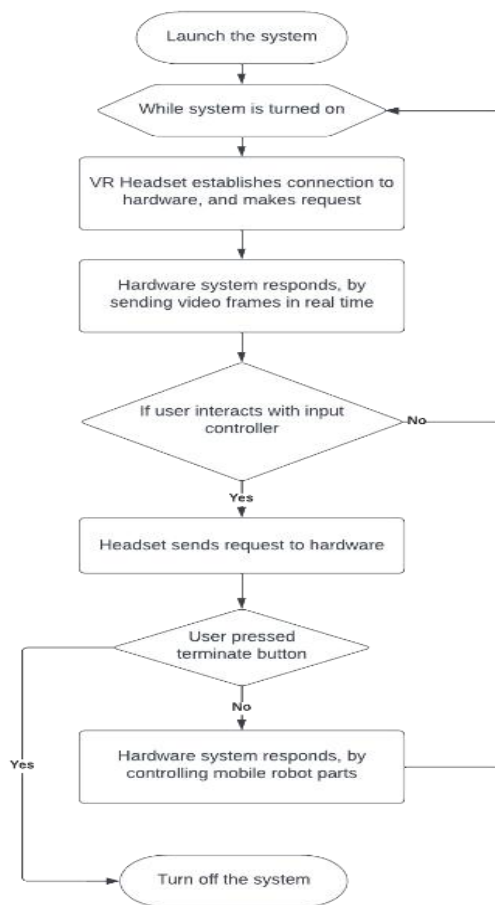
- a DC-motorok vezérlése GPIO-n keresztül,
- a kamera képének továbbítása a VR-headset felé,
- a VR-ből érkező vezérlőparancsok fogadása és végrehajtása

A **kommunikáció** kliens–szerver architektúrával működik:

- a Raspberry Pi a szerver (Flask + OpenCV),
 - a VR-alkalmazás a kliens (Unity + C#),
- és a két eszköz valós idejű videó- és vezérlőadat-cserét folytat.

Az **Oculus SDK** biztosítja az inputrendszerek konfigurálását, így a VR-környezet teljeskörűen integrálható a robot valós

idejű irányításába. A rendszer működésének folyamatábrája látható a lenti ábrán.



8. ábra A VR rendszer működésének részletes folyamatábrája

2.6.2. Eredmények és összefoglalás

A kutatás célja egy olyan, valós időben működő távirányítási környezet létrehozása volt, amelyben a kezelő egy VR-alapú

vezérlőfelületen keresztül, intuitív módon irányíthat egy differenciálhajtású mobil robotot. A rendszer megtervezésekor elsődleges szempont volt a költséghatékonyság és a késleltetés minimalizálása, ezért a teljes architektúra alacsony árú, mégis nagy számítási teljesítményű komponensekre épült (Raspberry Pi 4, L298N motorvezérlő, egyszerű DC-motoros platform, valamint Oculus Quest 2 VR headset).

A vezérlés és a képi információk átvitele egy kliens–szerver modellt követ:

- **A Raspberry Pi tölti be a szerver szerepét**, amely a kamera képét valós időben továbbítja, és fogadja a mozgásparancsokat. A videófeldolgozás OpenCV-vel, míg a hálózati kommunikáció Flask alapú aszinkron webszolgáltatással valósul meg.

- **A VR-szemüveg Unity-alapú kliensalkalmazása** biztosítja a vezérlőelemeket (kormány, irányváltó, vészleállító), illetve továbbítja a felhasználó gesztusait és joystick-mozgásait.

A differenciálhajtású mobil robotot a Raspberry Pi GPIO-kimenetei vezérlik. A két egyedi hajtott kerék fordulatszámának szabályozásával megvalósul a robot előre-hátra mozgása, valamint a helyben történő forgás, ami jól illeszkedik a VR-vezérlés igényeihez. A választott platform egyszerűsége ellenére stabil, pontos és könnyen manőverezhető működést tesz lehetővé.

A valós idejű működést a rendszer optimalizált jelátviteli lánc biztosítja: az adatcsomagok kisméretűek, a kommunikáció késleltetése alacsony, a videófolyam továbbítása pedig folyamatos, ami kritikus a felhasználói élmény és a robot biztonságos irányíthatósága szempontjából.

A felhasználói tesztek igazolták, hogy a VR-alapú interfész gyorsan tanulható, a valóságghú vezérlőelemek (kormány,

karok) javítják az immerziót, és a kezelő térbeli tájékozódását jelentősen segíti az élő videókép. A joystickos irányítás különösen stabilnak bizonyult, míg a kézmozdulat-követés a kamerák látómezőjéből való kikerülés miatt időnként hibára hajlamos volt.

Összességében a kutatás bizonyította, hogy a VR-alapú távvezérlés alacsony költségű, kereskedelmi komponensekből is megvalósítható, és megfelelően optimalizált architektúrával a rendszer valós idejű működése is biztosítható. A létrehozott platform tovább fejlesztésre alkalmas, például autonóm funkciók, érzékelők vagy útvonaltervező algoritmusok integrálásával.

3. Irodalmi hivatkozások listája

[Nagy, 1997-1] I. Nagy, L.Vajta, Cs. Gürtler; „*Vision based navigation for multi-agent mobile robot system in the transport domain*”; Proc. of the INES 97, IEEE Int. Conf. on Intelligent Engineering Systems; ISBN: 0780336275, doi: 10.1109/INES.1997.632387, pp.: 165-170, Budapest, 1997.

[Nagy, 1997-2] I. Nagy, L.Vajta, Cs. Gürtler, „*Trajectory Planning for Mobile Robots Based on Dynamical Models*”; Proc. of the INES 97, IEEE Int. Conf. on Intelligent Engineering Systems; ISBN: 0780336275, doi: 10.1109/INES.1997.632387, pp.: 170-174, Budapest, 1997.

[Nagy, 2009] Nagy István; „*Pozicionálási pontosságon alapuló, közel idő-optimalis pályatervezési eljárás mobilrobotokhoz*”; PhD értekezés, 2009, p.: 130, BME, Budapest.

[Claire, 2020] Claire David; „*Wandering across the Weierstrass function, while revisiting its properties*”; Analysis of PDE-Morningside Lecture notes, In press. fhal-03696196f, 2020.

[Halme, 1993] A. Halme, P. Jakubik, T. Schonberg and M. Vainio, "*The concept of robot society and its utilization*," *Proceedings of 1993 IEEE/Tsukuba International Workshop on Advanced Robotics*, Tsukuba, Japan, 1993, pp. 29-35, doi: 10.1109/ICAR.1993.337222.

[Benotsmane, 2022] Benotsmane, R.; Vásárhelyi, J.; „*Towards Optimization of Energy Consumption of Tello Quad-Rotor with Mpc Model Implementation*". *Energies* 2022, 15, 9207; doi.org/10.3390/en15239207; 2022.

[Sabban, 2013] W. H. Al-Sabban, L. F. Gonzalez and R. N. Smith, "*Wind-energy based path planning for Unmanned Aerial Vehicles using Markov Decision Processes*," *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 784-789, doi: 10.1109/ICRA.2013.6630662.

[Pham, 2022] Pham, K.L.; Leuchter, J.; Bystricky, R.; Andrie, M.; Pham, N.N.; Pham, V.T., „*The Study of Electrical Energy Power Supply System for UAVs Based on the Energy Storage Technology*"; *Aerospace* 2022, 9, 500. <https://doi.org/10.3390/aerospace9090500>

[Desreux, 2019] A. Desreux, A. Bouscayrol, R. Trigui, E. Castex and J. Klein, "*Impact of the Velocity Profile on Energy Consumption of Electric Vehicles*," in *IEEE*

Transactions on Vehicular Technology, vol. 68, no. 12, pp. 11420-11426, Dec. 2019, doi: 10.1109/TVT.2019.2949215.

[Kebriaei, 2015] M. Kebriaei, A. H. Niasar and B. Asaei, "Hybrid electric vehicles: An overview," *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, Shenzhen, China, 2015, pp. 299-305, doi: 10.1109/ICCVE.2015.84.

[Fantin, 2021] Fantin Irudaya Raj, E., Appadurai, M; „*The Hybrid Electric Vehicle (HEV)—An Overview*”; In: Kamaraj, V., Ravishankar, J., Jeevananthan, S. (eds) *Emerging Solutions for e-Mobility and Smart Grids*. Springer Proceedings in Energy. Springer, Singapore; doi.org/10.1007/978-981-16-0719-6_3; 2021.

[Deng, 2025] T. Deng et al., "MNE-SLAM: Multi-Agent Neural SLAM for Mobile Robots"; *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 1485-1494, doi: 10.1109/CVPR52734.2025.00146, 2025.

[Kai-Tang, 2025] Kai-Tang Song, Yuan-Shou Hsieh, „*Energy-Efficient Obstacle Avoidance via Iterative B-Spline Optimization for a Mobile Manipulator in Dynamic Environments*"; *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)* October, Hangzhou, China, 2025.

[Eshtehardian, 2023] Eshtehardian, S.A., Khodaygan, S; „A continuous RRT*-based path planning method for non-holonomic mobile robots using B-spline

curves"; *J Ambient Intell Human Comput* **14**, 8693–8702 (2023); doi.org/10.1007/s12652-021-03625-8; 2023.

[Galarza, 2023] Galarza, B.R.; Ayala, P.; Manzano, S.; Garcia, M.V. „*Virtual Reality Teleoperation System for Mobile Robot Manipulation*”. *Robotics* 2023, *12*,163; doi.org/10.3390/robotics12060163; 2023.

[Alsamhi, 2023] Alsamhi, S.H., Hawbani, A., Kumar, S., Curry, E.; „*FlyVR in the Sky: Virtual Reality Framework for Unmanned Aerial Vehicles in B5G*”; In: Arai, K. (eds) *Proceedings of the Future Technologies Conference, Volume 1. FTC 2023. Lecture Notes in Networks and Systems*, vol 813. Springer, Cham.; doi.org/10.1007/978-3-031-47454-5_41; 2023.

[Nagy, 2002] I. Nagy; „*Marker-Based Mobile Robot Positioning in Respect of Displacement of the Markers*”; in *Proceedings of the 6th International Conference on Intelligent Engineering Systems, INES 2002*, ISBN: 9536071177, pp. 63-68, Opatija, Croatia, 2002.

[Nagy, 2004] I. Nagy; „*Marker-Based Mobile Robot Positioning in Respect of Displacement of the Markers*”; in W Elmenreich ; J T Machado ; I J Rudas (eds); *Intelligent Systems at the Service of the Mankind*; Augsburg: U Books, pp. 99-109, 2004.

4. Tézisekhez kapcsolódó legfontosabb irodalmi hivatkozások listája

- [1] Nagy, I.; Laufer, E. „*Energy-Optimized 3D Path Planning for Unmanned Aerial Vehicles*”; Applied Sciences 2024, 14, 6988, doi:10.3390/app14166988.
- [2] I. Nagy, A. R. Varkonyi-Koczy and A. Dineva, "GA optimization and parameter tuning at the mobile robot map building process"; 2013 IEEE 9th International Conference on Computational Cybernetics (ICCC), Tihany, Hungary, 2013, pp. 333-338, doi: 10.1109/ICCCyb.2013.6617613.
- [3] I. Nagy; „*Genetic Algorithms Applied for Potential Field Building in Multi-Agent Robotic System*”; In: I. J., Rudas; A, Szakál (eds.) Proceeding of the International Conference on Computational Cybernetics, ICCC 2003, Budapest, Hungary, Budapest Polytechnic, pp. 104-108, ISBN: 9637154184, 2003.
- [4] I. Nagy, A.L. Bencsik; „*A simulation system for behaviour based potential field building in multi-agent mobile robot system*”; in R Andonie (eds.), 3rd IASTED International Conference on Computational Intelligence (CI 2007), Banff, Canada, Anaheim (CA): ACTA Press, pp 7-12, Anaheim (CA): ACTA Press, pp 7-12, ISBN: 9780889866713, 2007.
- [5] I. Nagy; „*From Exploring to Optimal Path Planning Considering Navigation's Error in Multi-agent Mobile Robot's Domain*”; ACTA POLYTECHNICA HUNGARICA, (ISSN: 1785-8860 1785-8860): **11** 6 pp 39-55, doi: 1012700/APH.11.06.2014.06.3, 2014.
- [6] I. Nagy and B. Selmei; "Artificial Error Map Simulation in Dynamic Environment on Multi-Agent Domain"; 2022 IEEE

10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC), Reykjavík, Iceland, pp. 000203-000208, doi: 10.1109/ICCC202255925.2022.9922832, 2022.

[7] I. Nagy; „*Hierarchical Mapping of an Electric Vehicle Sensor and Control Network*”; ACTA POLYTECHNICA HUNGARICA (1785-8860 1785-8860): **18** 9 pp 161-180, doi: 10.12700/APH.18.9.2021.9.10, 2021.

[8] I. Nagy; "Hierarchical Organization of Control and Sensory System of an Electric Vehicle"; 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2021, pp. 287-292, doi: 10.1109/SACI51354.2021.9465638.

[9] I. Nagy; „*Matrix representation of an EV's Sensory and Control System*”; in IOP Conference Series: Materials Science and Engineering, Volume 1237, 6th Agria Conference on Innovative Vehicle Technologies and Automation Solutions (InnoVeTAS 2022), doi: 10.1088/1757-899X/1237/1/012003, Eger, Hungary

[10] I Nagy; „*Mathematical Description of an EV's Sensory and Control System*”; in Kassmi Prof. Khalil (eds.) Advances and Challenges in Science and Technology, Vol. 8, London: BP International, pp. 121-136, ISBN 978-81-967198-2-1 (eBook), doi: 10.9734/bpi/acst/v8, 2023.

[11] I. Nagy; Y. Kenzhetayev; „*A Real-time, Remotely Operated Vehicle, using a Raspberry Pi and a Virtual Reality Headset*”; ACTA POLYTECHNICA HUNGARICA (1785-8860 1785-8860): **21** 8 pp 125-146 (2024), doi: 10.12700/APH.21.8.2024.8.7, 2024.

4.1 A témához kapcsolódó további irodalmi hivatkozások listája

[12] I. Nagy; "*Fuzzy set based navigation's area application for mobile robots*" 2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 2011, pp. 507-511, doi: 10.1109/CINTI.2011.6108559.

[13] I. Nagy; "*A complex path planning model based on localization's error for mobile robots*" 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), Lisbon, Portugal, 2012, pp. 55-59, doi: 10.1109/INES.2012.6249899.

[14] Sz. Tuloki, I. Nagy; „*Vehicle Steering Technologies, Direct Adaptive Steering : Gépjármű kanyarodási technológiák, közvetlen adaptív kormányzás*”; in A XXII. Fiatal Műszakiak Tudományos Ülésszaka előadásai, Kolozsvár: Erdélyi Múzeum Egyesület (EME), Óbudai Egyetem, pp 395-398 (2017), ISBN: 9789634490180, (Műszaki Tudományos Közlemények / Papers on Technical Science 2393-1280 ; 7)

[15] I. Nagy, G. Dinev and A. Dineva, "Aggregation operators in accurate potential field building" 2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES), Larnaca, Cyprus, 2017, pp. 000219-000224, doi: 10.1109/INES.2017.8118559.

[16] I. Nagy, Sz. Tuloki; „*Elektromos gépjárművek szenzorhálózatának feltérképezése és biztonsági elemzése: Mapping and Security Analysis of Sensor Network for Electric*

Vehicles", In: Cluj-Napoca, Kolozsvár: Óbudai Egyetem, Erdélyi Múzeum Egyesület (EME), pp 243-247 (2018) (Műszaki Tudományos Közlemények/Papers on Technical Science 2393-1280 ; 9.), doi.: 1033895/mtk-2018.09.56.

[17] I. Nagy, Sz. Tuloki, "*Fault Analysis and System Modelling in Vehicle Engineering*" 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 2018, pp. 000313-000318, doi: 10.1109/CINTI.2018.8928203.

[18] Tuloki Sz., Nagy I.; „*Elektromos gépjárművek szenzortérképének elkészítése és a keletkező szenzorhiba*”; in XI. IFFK 2017 : Innováció és fenntartható felszíni közlekedés; Budapest, Magyarország 2017.08.30. - 2017.09.01., ISBN: 9789638887528, Budapest: Magyar Mérnökakadémia (MMA), p. &. (2017)

[19] I. Nagy and F. B. Vér, "*Practical Analysis of an AMR Wheel Speed Sensor and CAN Bus System for the Communication's Failures and its Implications to the EV's Control System*" 2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 2019, pp. 241-248, doi: 10.1109/SISY47553.2019.9111548.

[20] R. Pesti, P. Sarcevic, D. Csík, I. Nagy and A. Odry, "*Comparison of optimization algorithms for installation error calibration of accelerometer arrays*" 2023 IEEE 23rd International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 2023, pp. 47-52, doi: 10.1109/CINTI59972.2023.10382086.

[21] I. Nagy; „*Behaviour Study of a Multi-agent Mobile Robot System During Potential Field Building*”; ACTA POLYTECHNICA HUNGARICA (1785-8860 1785-8860): **6** (4) pp. 111-136 (2009)

[22] I. Nagy; „*Path Planning Algorithm, Based on User Defined Maximal Localization Error*”; PERIODICA POLYTECHNICA-ELECTRICAL ENGINEERING (0324-6000 1587-3781): 49 1-2 pp 43-57 (2005),

[23] I. Nagy, Y. Kenzhetayev, M. Szakálos; "Facial Recognition and Motion Tracking in Dynamic Environment by Drone Control Software" 2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 2022, pp. 000233-000238, doi: 10.1109/SISY56759.2022.10036301.