

Tudományos Diákköri Konferencia

Hierarchikus trajektóriatervezés autonóm versenyautók számára

Szerző:

Losonczy Dániel



Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedés- és Járműirányítási Tanszék

Konzulens:

Dr. Fehér Árpád

Tartalomjegyzék

1. Bevezetés	1
1.1. Dolgozat felépítése	4
2. Szakirodalmi áttekintés	5
2.1. Hierarchikus járműirányítási struktúra	5
2.2. Globális útvonaltervezés	6
2.3. Viselkedés döntéshozatal	9
2.4. Lokális mozgástervezés	10
2.4.1. Megerősítéses tanulás	10
2.4.2. Model Predictive Control	11
2.4.3. Numerikus optimalizáción alapuló módszerek	13
2.5. Visszacsatolt szabályozás	14
3. Hierarchikus trajektóriatervezés megvalósítása	15
3.1. Szimulációs környezet	15
3.2. Geometria alapú globális útvonaltervezés	16
3.2.1. Kanyar csúcspont és a versenyívek kapcsolata	16
3.2.2. Globális útvonal előállítása	17
3.3. Lokális mozgástervezés	20
3.3.1. Frenet koordinátarendszer	20
3.3.2. Trajektória meghatározása negyed- és ötödfokú polinom illesztéssel	21
3.4. Globális útvonal alapján lokális trajektóriatervezés numerikus optimalizá- cióval	23
4. Eredmények	25
5. Konklúzió	29

Összefoglaló

Az autonóm járművek világa egyre nagyobb figyelmet kap a kutatás és az ipar területén, mivel a jövő közlekedése szempontjából kulcsfontosságúak. Azok a cégek és iparági szereplők, akik sikeresen integrálják az autonóm rendszereket, versenyelőnyre tehetnek szert, vonzóbbá válva a vásárlók számára a csúcstechnológiai megoldások révén. Az ilyen rendszerek biztonságos és hatékony működése érdekében elengedhetetlen, hogy megfelelően reagáljanak a dinamikusan változó környezetre. E kihívások középpontjában áll az optimális trajektória tervezése, különösen olyan környezetekben, ahol a gyors és pontos irányítás alapvető fontosságú, mint például a versenysportokban, ahol a teljesítmény optimalizálása döntő a sikerhez. A dolgozatban egy hierarchikus trajektóriatervezési módszert mutatok be, amely két szinten működik. Az első szinten geometriai alapú módszerrel kerül meghatározásra a globális útvonal, amely közel optimális íveket generál. A második szinten a lokális mozgástervezés numerikus optimalizálással finomítja az útvonalat, alkalmazkodva a környezeti változásokhoz és a jármű aktuális állapotához. Ez lehetővé teszi optimalizált sebességprofilok generálását is, biztosítva a leggyorsabb és legbiztonságosabb irányítást. A módszer egy saját fejlesztésű egyedi szimulációs környezetben került tesztelésre, különböző pályákon, melyek a szabványos úttervezési irányelveknek megfelelően kerültek generálásra. A szimulációs környezet lehetőséget ad az algoritmusok teljesítményének értékelésére és finomhangolására, továbbá cél valós járműves tesztek végrehajtása is.

Abstract

The world of autonomous vehicles is capturing increasing attention from both research and industry, as these technologies hold the key to the future of transportation. Companies and industry leaders who successfully integrate autonomous systems can gain a significant competitive edge, offering customers advanced, high-tech solutions. For these systems to function safely and efficiently, it is crucial that they respond in real-time to dynamically changing environments. Central to these challenges is the design of optimal trajectories, especially in settings where precise and rapid control is critical, such as competitive sports, where performance optimisation is a cornerstone of success. In this paper, I introduce a hierarchical trajectory planning method that operates on two levels. The first level uses a geometry-based approach to define a global trajectory, generating near-optimal paths. At the second level, local trajectory planning refines the path in real time, adjusting to environmental changes and the vehicle's current state. This also enables the generation of optimised speed profiles, ensuring the fastest and safest driving experience. The method was tested using IPG Carmaker software on various tracks, with its performance compared to the capabilities of human drivers. The simulation environment provides an opportunity to evaluate and fine-tune the algorithm's performance, with the ultimate aim of conducting real-world vehicle tests.emonstrate the advantages and limitations of the technologies and methods we have developed.

1. fejezet

Bevezetés

Az autonóm járművek (AV) technológiája napjainkban egyre hangsúlyosabb szerepet játszik társadalmunkban, különösen a fejlett vezetéstámogató rendszerek (ADAS) formájában, amelyek a kutatásban és a kereskedelmi járművekben egyaránt megjelennek. E technológiák célja, hogy csökkentsék a balesetek számát és súlyosságát, növeljék a mobilitást a fogyatékkal élők és az idősek számára, csökkentsék a kibocsátást, valamint hatékonyabban használják az infrastruktúrát. Az autonóm járművek technológiájának fejlődését erőteljesen ösztönzi az a tényező, hogy e rendszerek immunisak az emberi hibákra, például a figyelmetlenségre, a fáradtságra és az érzelmi vezetésre, amelyek az Amerikai Autópálya Közlekedési Biztonsági Igazgatóság (NHTSA) statisztikái szerint a balesetek körülbelül 94%-át okozzák [1].

Azonban autonóm járművek koncepciója már évtizedekkel korábban, az 1920-as évek elején felmerült, de az 1980-as években kapott igazán lendületet, amikor megalkották az automatizált autópálya-rendszerek ötletét. Az ITS rendszerek alkalmazásával a járművek ma már képesek lehetnek kommunikálni az autópálya-infrastruktúrával. Az AV technológia úttörői elsősorban az Egyesült Államokban és Németországban kezdtek el fejleszteni 1980 és 2000 között. Emellett az amerikai DARPA (Fejlett Védelmi Kutatási Projektek Ügynöksége) által végzett kiterjedt kutatások is felgyorsították az AV-technológia fejlődését [2]. A kutatók és ipari vezetők versengtek az első olyan teljesen autonóm jármű kifejlesztéséért, amely elég robusztus, megbízható és biztonságos a valós és nagy sebességű közlekedési környezetekben. Ezek a versenyek lehetőséget biztosítottak az ipar és a kutatók számára, hogy felmérjék az AV-k képességeit és határait különböző közlekedési környezetekben. Szeretném kiemelni, hogy ezen versenyek tárták fel az AV szoftver- és hardvermegoldásokban meglévő jelentős nehézségeket és hiányosságokat, amelyek egy része a mai napig megoldatlan maradt.

Ahogy a kutatás, a tesztelés és az autonóm technológiával felszerelt járművek bevezetése világszerte növekszik, úgy a szabványosított irányelvek és szabályozások fejlesztése is középpontba került, hogy biztosítsák a technológia biztonságos integrációját a társadalomba. Az SAE (Society of Automotive Engineers) nemzetközi szabványát az automatizálási

szintek meghatározására először 2014-ben fogadták el, amelyek az autonóm járműveket a 0. szinttől (az emberi vezető teljes irányítása) az 5. szintig (a jármű teljesen önállóan vezet) definiálják.

Jelenleg, a rendelkezésre álló érzékelők korlátozott mivolta és magas költségeik miatt a legtöbb kereskedelmi jármű csupán az 1-es és 2-es szintet valósítja meg, amely folyamatos vezetői figyelmet és irányítást igényel. Ezek a járművek általában vészfékezési, holtér-érzékelési és/vagy sávban tartási funkciókat kínálnak. Bizonyos modellekben már elérhetőek a 3-as szintű autonóm funkciók is. Ugyanakkor a közelmúltban történt balesetek aggodalmat keltettek a vezetők technológia használatának biztonságos megértése és képességei kapcsán [3].

Napjainkban megoldandó problémák közé tartozik, hogy a különböző vezetési szokások mellett a technológia körültekintően kell, hogy működjön. Ahhoz, hogy az autonóm funkciók és járművek a biztonság, a mobilitás és a hatékonyság javításában jelentős hatással bírjanak, a nyilvánosságnak bizonyos mértékű ismeretséget kell szereznie a technológia képességeit, beleértve annak korlátaival, alkalmazási területeit illetően, amikor a technológia használata ajánlott vagy elvárt.

Egy hierarchikus megközelítés szerint az autonóm járművek működésének egymásra épülő fő összetevői: érzékelés, trajektória tervezés, döntéshozatal és járműirányítás. Az érzékelés során folyamatosan pásztázza és figyeli a környezetet, hasonlóan az emberi látáshoz és egyéb érzékekhez [4]. Az érzékelők adatai alapján lokalizál és térképez meghatározva az ego-jármű globális és lokális helyzetét, és térképet készít a környezetről. A trajektória tervezés meghatározza az ego-jármű számára lehetséges biztonságos útvonalakat. A döntéshozó komponens felelős a jármű aktuális cselekedete meghatározásáért, mint például sávváltás, jobbra kanyarodás vagy más manőver. A szabályozórendszer pedig kiszámítja a megfelelő járműparancsokat (gyorsulás, kormánykerék szög, stb.), hogy követni tudja az optimális trajektóriát.

A hierarchikus irányítási struktúra egy olyan megközelítés, amely jó alapot adhat az autonóm járműrendszerek számára elősegítve, hogy a járművek helyesen navigáljanak a bonyolult közlekedési helyzetekben, és javítsák a biztonságos és hatékony közlekedést. A struktúrát alkalmazó fejlett rendszerek, a meghatározott szinteket követve jól elszeparált szoftvermodulok fejlesztését teszi lehetővé, melyeknek nem csupán a bővítése, de a karbantartása is hatékonyabb tud lenni azokhoz a rendszerekhez képest, ahol ezen rétegeket összemossák.

A hierarchikus trajektóriatervezés lehetővé teszi, hogy a járművek több szinten, eltérő gyakorisággal és részletességgel tervezzék meg a mozgásukat. Ez a több szinten való tervezés hozzájárul a balesetek kockázatának csökkentéséhez, mivel a jármű képes rövid és hosszú távon is biztonságos és hatékony manővereket végrehajtani. Az ilyen rendszerek könnyen alkalmazkodnak dinamikus változó forgalmi környezethez is, ami kulcsfontosságú a zavartalan közlekedés fenntartásához.

A TDK munkámban egy módszert fogok bemutatni, ahol a trajektória tervezése két szinten valósul meg: globális és lokális, mindkettő saját kihívásaival és komplexitásával rendelkezik. A globális trajektóriatervezés célja, hogy a jármű számára egy teljes útvonalat határozzon meg egy versenypálya koordináta rendszerében. A lokális trajektória tervezése a jármű lokális koordinátarendszerében történik és az a rövidtávú részletes mozgásstratégia kidolgozására összpontosít. Ez magában foglalja a sebességprofilok optimalizálását is, amelyek lehetővé teszik a jármű számára, hogy hatékonyan és biztonságosan navigáljon a környezeti feltételeknek megfelelően. A dolgozat egy olyan hierarchikus két szintes trajektóriatervező módszert mutat be, amely versenyautók számára geometriai módon határozza meg a kanyar csúcspontokat érintő leggyorsabb globális útvonalat. S ezen útvonal a lokális mozgástervező révén további pontosításra kerül a hosszirányú sebességprofilok meghatározásával klasszikus algoritmusok és numerikus optimalizáció során.

1.1. Dolgozat felépítése

Az első fejezete ismerteti a dolgozat célkitűzését, amely a hierarchikus trajektóriatervezés alkalmazása autonóm versenyautók számára, és felvázolja a kutatás fő irányvonalát.

A második fejezet áttekintést nyújt a szakirodalomban található releváns módszerekről, amelyek a hierarchikus járműirányítás struktúráját és a trajektóriatervezést foglalják magukban. A fejezet tárgyalja a globális útvonaltervezés és viselkedési döntéshozatal szerepét, továbbá bemutat különböző lokális mozgástervezési megközelítéseket, mint a megerősítéses tanulás, a Model Predictive Control (MPC), és a numerikus optimalizáción alapuló technikák. Végül a visszacsatolt szabályozás jelentősége is bemutatásra kerül.

A harmadik fejezet részletesen bemutatja a dolgozat központi elemét, a hierarchikus trajektóriatervezési rendszert, mely a globális és lokális trajektóriatervezést integrálja. A fejezet tárgyalja: A szimulációs környezet fejlesztését, amely lehetővé teszi a módszer tesztelését. A geometriai alapú globális útvonaltervezést, amely az elméleti versenyívet alakítja ki a pályán található kanyar csúcspontok és versenyívek felhasználásával. A lokális mozgástervezést, amely a trajektóriák pontosítására szolgál a Frenet koordinátarendszer és polinomos illesztések segítségével. A globális útvonal finomítását lokális mozgástervezési optimalizációval, amely a jármű aktuális pályaelemekhez igazodó, finomabb trajektóriáját hozza létre.

A negyedik fejezet részletesen elemzi a szimulációs eredményeket, összehasonlítva több módszer szerint meghatározott versenyív mentén történő navigálás teljesítményét. A tesztek kimutatták, hogy a hierarchikus módszer jobb köridőket, kisebb oldalirányú gyorsulásokat és alacsonyabb csúszási értékeket eredményez, különösen kanyarodás során. Az utolsó fejezet összegzi a kutatás megállapításait, és javaslatot tesz a jövőbeli fejlesztésekre, beleértve a módszer tesztelését valós környezetben vagy IPG CarMaker szimulációs platformon, valamint a valós vezetők által leírt versenyívvel való összehasonlítását.

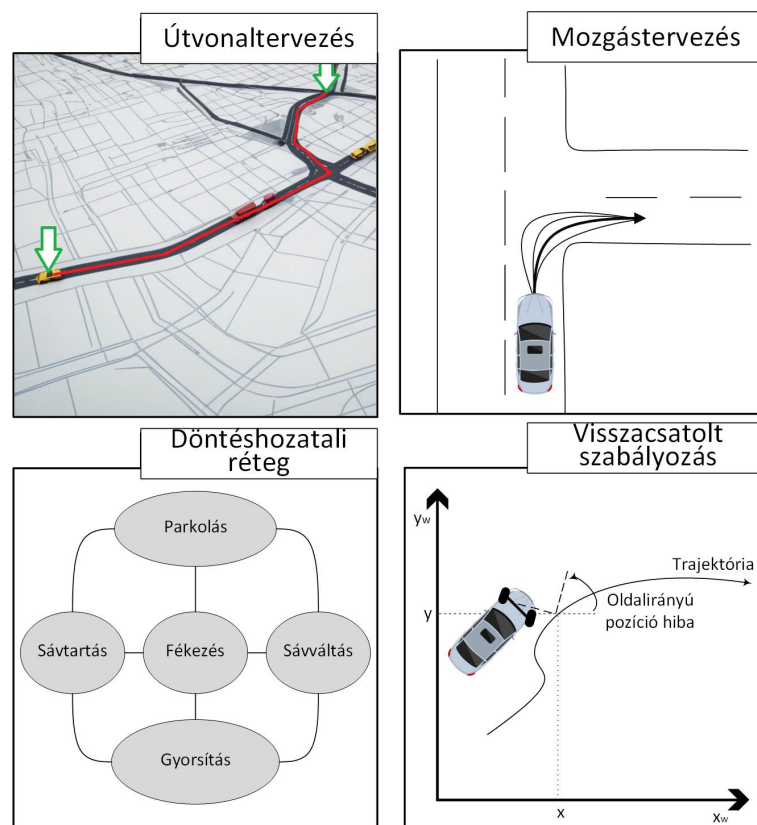
A konklúzió hangsúlyozza, hogy a valós idejű alkalmazások során a jármű dinamikai viselkedésének előrejelzése és integrálása az optimalizációba kulcsfontosságú az autonóm versenyautók trajektória-tervezésének továbbfejlesztése érdekében.

2. fejezet

Szakirodalmi áttekintés

2.1. Hierarchikus járműirányítási struktúra

Az utóbbi időben több módszert is meghatároztak az autonóm járművek irányítási struktúrájára, ezek közül az egyik legelterjedtebb egy hierarchikus struktúra [5] amely négy alapvető szintet határoz meg. Ezek szerepe egymástól jól elkülöníthető 2.1.

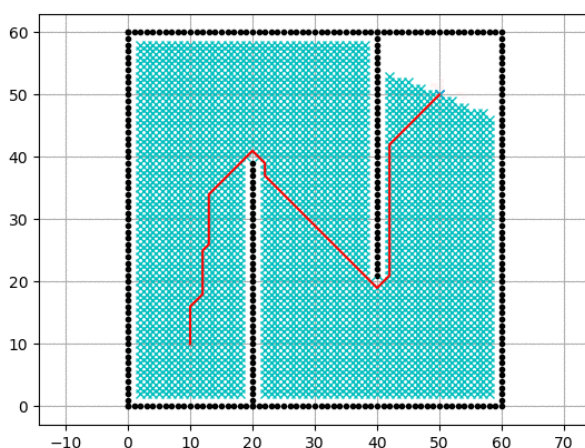


2.1. ábra. Járművek döntéshozatali hierarchiájának szintjei

2.2. Globális útvonaltervezés

Az első szinten az autonóm járművek globális útvonalat terveznek meg egy előre meghatározott úthálózaton belül, hogy elérjék a célállomásukat a jelenlegi helyzetükből kiindulva. Ezen probléma megoldására számos megközelítés létezik, melyeket különböző kategóriákra lebontva megkülönböztetünk geometriai alapú módszereket, heurisztikus megoldásokat és optimális irányításra vagy nemlineáris programozásra épülő algoritmusokat, azonban ezeket az algoritmusokat tipikusan lokális mozgástervezésre alkalmazzák. A geometriai alapú megközelítés jelentős számítási kapacitást igényelhet, ezért a heurisztikus megoldásokkal szokták együtt alkalmazni, melyek során keresési vagy véletlenszerűen mintavételező algoritmust alkalmaznak.

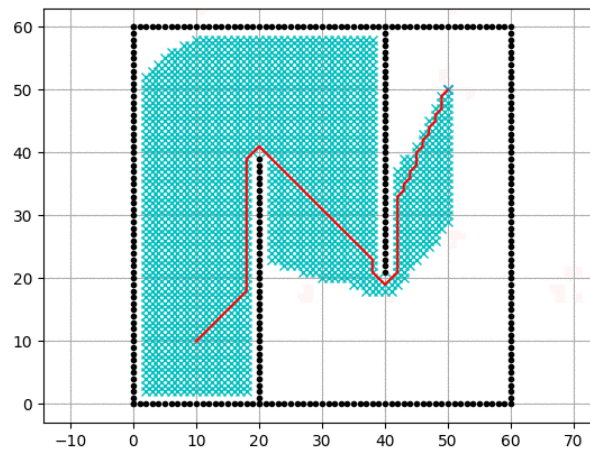
Az első pályatervezési módszert, amely geometriai alapú 1979-ben javasolták először, melyet Láthatósági gráf-nak neveztek el [6]. Ez a módszer egy konvex környezetben való pálya létrehozására alkalmas megoldást ír le. Az akadályokat foglaltsági térképen úgy reprezentálják, hogy azok a területeket definiálják, amelyek nem járhatók a jármű számára. Ezután egy gráfot hoznak létre, ahol az akadályok csúcsai csomópontként szerepelnek. Ez egy láthatósági gráf, amelyen keresési algoritmusokkal végezhető el a pályatervezés. Ezen keresésre a Dijkstra algoritmus [7] az egyik legismertebb, amely garantálja a legkisebb költséggel járó útvonal megtalálását. Keresés során minden csomópontra meghatározza a legkisebb távolságot a kiindulóponttól, majd iteratíván keresi a következő legközelebbi csomópontot, míg el nem éri a célpontot. Emiatt az algoritmus ugyan pontos, de nagy terekben meglehetősen idő- és számításiigényessé válik. A 2.2 ábra jól illusztrálja a módszert.



2.2. ábra. Dijkstra algoritmus [8]

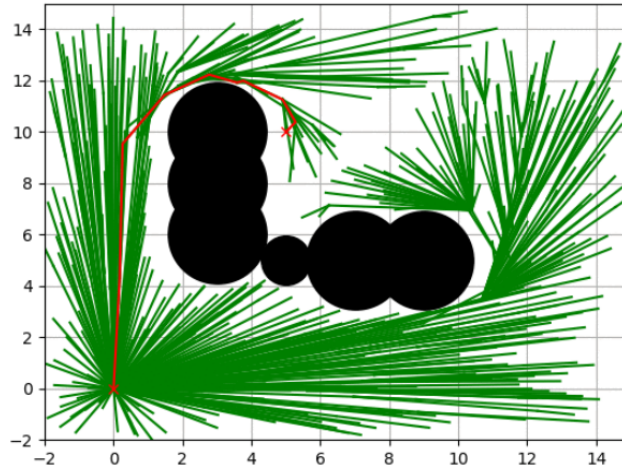
Az A^* algoritmus [9] viszont egy heurisztikus megközelítést alkalmaz a legrövidebb út megtalálására. Az A^* algoritmus hasonlít a Dijkstra algoritmushoz, de a heurisztikát használva gyorsabban el tudja érni a célállapotot, mivel nem vizsgál meg minden csomópontot, hanem figyelembe veszi a korábban jobbnak talált csomópont összeköttetéseket. Ezáltal az A^* algoritmus sokkal hatékonyabb tud lenni, különösen nagyobb vagy komplex környezetekben, de a Dijkstra-hoz hasonlóan ez a módszer is mindig a célpont-hoz legközelebbi pontot priorizálja, ennek köszönhető a(z) 2.3 ábrán látható útvonalnak az akadályokhoz való simulása. Azonban az A^* csupán diszkrét megoldást nyújt, ezért nem alkalmazható közvetlenül a nemholonom járművek útvonaltervezéséhez, ezért ennek kiterjesztéseit használják, mint például a Hibrid A^* algoritmust [10].

◦



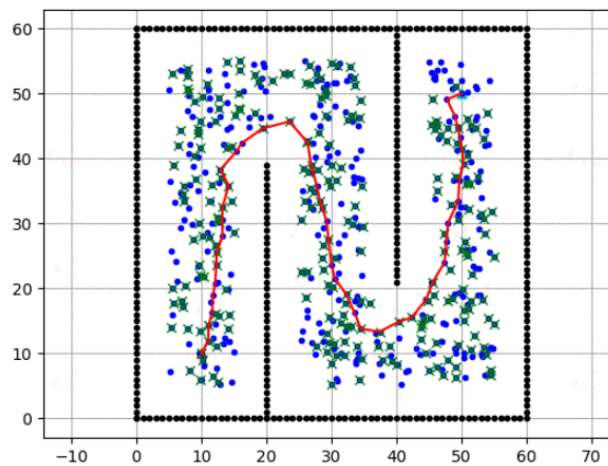
2.3. ábra. A^* algoritmus [8]

A véletlenszerűen mintavételező algoritmusok közül az RRT (Rapidly exploring Random Trees) [11] az egyik legnépszerűbb megközelítés, amely 1998-ban került bevezetésre La Valle által. Az RRT-k fa-struktúrákat hoznak létre, amelyek iteratívan terjeszkednek a keresési térben, kezdve a kiindulóponttól (gyökér) növekedve, amíg egy úgynevezett levele el nem éri a célpontot. Az RRT-k számos előnnyel rendelkeznek, mint például a multidimenzionális terekben való alkalmazhatóság és a nemholonomikus korlátok kezelése. Azonban RRT-k esetében szintén érdemes alkalmazni a keresési algoritmusokat a hatékonyság növelése érdekében, mint az A^* vagy a Dijkstra, hogy optimalizálják a megtalált útvonalat. A 2.4 ábra bemutatja az RRT algoritmus fa-struktúráját.



2.4. ábra. RRT algoritmus [8]

Egy másik hasonló véletlenszerűen mintavételező algoritmus a Probabilistic roadmap (PRM)[12]. Ezen megoldás egy olyan gráfot hoz létre a jármű konfigurációs teréből (lehetőséges pozíciói és orientációi), amely közelíti azokat a mozgásokat, amelyek végrehajthatók az adott környezetben. Ebben az esetben a gráf csomópontjait úgy kapcsolja össze, hogy azok egy meghatározott távolságon belül legyenek. A további újabb lehetséges mozgásokat összekapcsolva válik a gráf egyre sűrűbbé, s a gráf építkezése véget ér, amint az elég sűrű ahhoz, hogy reprezentálja a mozgási lehetőségeket. A gráf elkészültével ebben az esetben is a korábban említett keresési algoritmusok egyikét kell alkalmazni az útvonal előállításához. A lehetséges pozíciókat és útvonalat az alábbi 2.5 ábra szemlélteti.



2.5. ábra. PRM algoritmus [8]

2.3. Viselkedés döntéshozatal

Miután az autonóm jármű számára meghatározták az útvonalat, a járműnek képesnek kell lennie, hogy az útvonalon végig haladva kövesse a közlekedési szabályokat, és megfelelően reagáljon a többi közlekedési résztvevőre. Ez a navigációs feladat nem csupán azt jelenti, hogy a jármű a helyes irányban halad, hanem azt is, hogy folyamatosan alkalmazkodik a forgalmi helyzetekhez, a közlekedési jelzésekhez, valamint az útviszonyokhoz. A viselkedési réteg feladata, hogy az aktuális forgalmi szituáció észlelésére alapozva minden pillanatban kiválassza a legmegfelelőbb vezetési magatartást. Például, ha a jármű egy kereszteződés előtti stopvonalhoz ér, a rendszer megállítja a járművet, figyeli a többi jármű, kerékpáros és gyalogos mozgását, és csak akkor indítja el újra, ha biztonságos a továbbhaladás, és a közlekedési szabályok szerint az következik.

A különböző vezetési helyzetek, például kereszteződéshez közeledés, sávváltás vagy gyalogos átkelő megközelítése, meghatározhatóak konkrét szabályokkal, és ezek a szituációk jól leírhatók véges halmazokkal. A különféle vezetési helyzetek közötti döntési folyamat modellezhető úgy, hogy minden viselkedést egy véges állapotú gép állapotaként képzelünk el, ahol az állapotok közötti váltásokat az adott helyzetben releváns környezeti tényezők, például a tervezett útvonalhoz és a közeli járművekhez viszonyított relatív pozíciók határozzák meg. Ily módon a rendszer automatikusan tudja kezelni a különböző vezetési scenáriókat, és ezek alapján meghozni a helyes döntést.

Az autonóm haladás során, különösen városi környezetben, azonban gyakran bizonytalanságokkal kell szembenézni, különösen ami a többi közlekedési résztvevő, például más járművek, kerékpárosok vagy gyalogosok szándékait illeti. Ezeknek a résztvevőknek a jövőbeli viselkedését, például az előrejelzett mozgásukat és a várható útvonalukat, előre meg kell becsülni a jármű megfelelő döntéseihez. Ennek érdekében különféle megközelítések léteznek, például gépi tanulási technikák, amelyek a múltbeli mozgások alapján képesek előrejelezni a közlekedési résztvevők jövőbeli viselkedését, vagy modellezési alapú módszerek, amelyek közvetlenül az érzékelőkből származó adatokra alapozva próbálják megérteni a résztvevők szándékait.

A viselkedési réteg tehát ezekkel a bizonytalanságokkal dolgozik, és valószínűségi modellek segítségével dönt arról, hogyan reagáljon a jármű a többi közlekedési résztvevő viselkedésére. A döntési folyamat gyakran valószínűségi tervezési formális rendszereket, például Markov-döntési folyamatokat használ, amelyek lehetővé teszik, hogy a rendszer kezelje a nem teljesen megfigyelhető, változó vezetési scenáriókat, például a gyalogosok szándékainak előrejelzését. Ezek a módszerek lehetővé teszik, hogy a jármű nemcsak a jelenlegi helyzet alapján, hanem a jövőbeli eseményeket figyelembe véve is optimálisan viselkedjen, biztosítva ezzel a biztonságos és hatékony közlekedést a forgalomban.

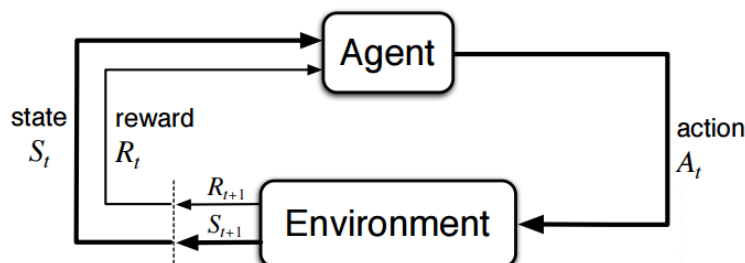
2.4. Lokális mozgástervezés

A lokális mozgástervezési réteg egy adott horizonton a jármű lokális koordinátarendszerében tervez. Feladata, hogy a jármű aktuális helyzetéből a célhelyzetbe egy biztonságos, kényelmes és megvalósítható trajektóriát számítson ki. Ez írja le hogyan változik a jármű állapota az idő függvényében lehetővé téve a jármű mozgásának közvetlen modellezését. A célhelyzet eltérhet a szituációtól függően, amit a döntési hierarchiában a viselkedési réteg határoz meg, lehet például a sáv középpontja pár méterrel előre, egy kereszteződés stopvonala, vagy egy kívánt parkolóhely. A mozgástervezési komponens figyelembe veszi a jármű körüli statikus és dinamikus akadályokat, és olyan ütközésmentes pályát generál, amely megfelel a jármű mozgási és kinematikai korlátaival. Gyakran a mozgástervező egy adott célfüggvényt is optimalizál, például minimalizálja az utazási időt, vagy bünteti a veszélyes vagy kényelmetlen mozgásokat. A mozgástervező által generált trajektóriát végül a visszacsatolós trajektóriakövető szabályozó réteg kapja meg. A hossz és keresztirányú szabályozás feladata, hogy a jármű pontosan végrehajtsa a tervezett mozgást.

Dinamikus környezetben a lokális mozgástervezés rendkívül összetett feladat, és PSPACE-nehéznek tekinthető [5], ami azt jelenti, hogy a megoldáshoz szükséges számítási kapacitás nagyon magas lehet. Ez azt jelenti, hogy az útvonalnak folyamatosan frissülnie kell a legújabb szenzoradatok és a jármű aktuális állapotának függvényében.

2.4.1. Megerősítéses tanulás

Napjainkban kiemelt és fejlett módszertan ezen probléma megoldására a megerősítéses tanulás (RL). A megerősítéses tanulás (RL) [13] a mesterséges intelligencia és gépi tanulás egyik területe, amelynek célja, hogy egy tanuló ágens (Agent) egy adott környezetben (Environment) sajátítsa el a céljai eléréséhez szükséges viselkedést. Ellentétben a felügyelt tanulással, ahol az ágens előre megkapja a jó és rossz viselkedés példáit, az RL ágensnek próbálkozás és hiba útján kell felfedeznie, hogyan érheti el a legnagyobb jutalmat (reward) 2.6.



2.6. ábra. Megerősítéses tanulás működési ábra [14]

Az ágensnek észlelnie kell a környezet állapotát (state), majd ezen információk alap-

ján olyan cselekvéseket (action) kell végrehajtania, amelyek új állapothoz vezetnek. Ezen cselekvései eredményeként az ágens jutalmat kap, ami segíti a jövőbeli viselkedésének kialakítását. A probléma modellezéséhez fontos a környezet állapotátmeneteinek leírása az ágens cselekvései alapján, amit részben egy POMDP-modell (Részlegesen Megfigyelhető Markov Döntési Folyamat) ír le. Egy kutatásban a mély megerősítéses tanulás (deep reinforcement learning) területén autóversenyezésre vonatkozóan mutattak be eredményeket és megoldást [15]. A kutatók egy olyan módszert dolgoztak ki, amely az úgynevezett end-to-end vezetést alkalmazza, vagyis a rendszer közvetlenül a jármű irányítását tanulja meg egy előre néző kamera RGB-képei és a jármű adatai alapján, anélkül, hogy köztes érzékelési lépéseket, például objektumfelismerést alkalmazna. A megoldásuk egy aszinkron aktor-kritikus (A3C) algoritmust használ, amely lehetővé teszi, hogy a jármű különböző pályákon (kanyarok, dombok) tanuljon, valósághű fizikai és grafikai szimulációk segítségével. Egy másik kutatás során [16] kimutatták, hogy a megerősítéses tanulási megközelítés segítségével gyorsan és hatékonyan generálhatók vészhelyzeti pályák, amelyek túlteljesítik az emberi sofőröket, különösen magas dinamikájú vezetési helyzetekben. Ebben a kutatásban a megerősítéses tanulás és a Model Predictive Control (MPC) segítségével megvalósított hierarchikus kitérési mozgástervezésre fókuszáltak, melynek eredményeit emberi sofőrökkel is összehasonlították. A kutatásból is megállapítható, hogy az MPC (Model Predictive Control) és a megerősítéses tanulás (Reinforcement Learning, RL) közti átfedés egyre gyakrabban fordul elő, és gyakran használják őket együtt, mert kiegészítik egymás erősségeit a mozgástervezés területén. Az RL megtanulhat magasabb szintű stratégiai döntéseket, mint például a sávváltás, előzés vagy akadálykerülés. Ebben az esetben az RL az MPC inputjaként szolgál, és segíti a rendszer stratégiájának fejlesztését, míg az MPC biztosítja, hogy a jármű rövid távú mozgása valós idejű és biztonságos legyen, ahol a fizikai korlátozások és a jármű dinamikája fontosak. A kettő módszer jól kombinálható, de önmagukban is helytállnak, ezért a következő alfejezetben bemutatom az MPC ezen területen való további alkalmazásainak lehetőségeit is.

2.4.2. Model Predictive Control

Az utóbbi időben számos tanulmány foglalkozott az MPC-alapú (Model Predictive Control) [17]-[18] mozgástervezési módszerekkel, amelyek a numerikus optimalizáción alapuló módszerek, valamint az adatalapú mozgástervezési módszerek közé tartoznak. Az MPC-alapú mozgástervezési módszerek, amelyek képesek figyelembe venni többféle korlátozást, mint például a saját jármű kinematikai és dinamikai korlátait, a környező közlekedési résztvevőket, az út határait és így tovább. Képesek arra is, hogy a jövőbeli információkat figyelembe vegyék a jelenlegi szabályozás során, és kezeljék a többféle kinematikai és dinamikai korlátozást az optimalizálási folyamatban, ami stabil, pontos és robusztus szabályozási teljesítményt eredményez. Ezen módszerek előnyöket nyújtanak a predikci-

ós információk kihasználásában és a többféle korlátozás figyelembevételében, különösen a dinamikai korlátozások esetében nagy sebességű scenáriókban, amelyek nagyon fontosak az autonóm járművek (AV) biztonsága és stabilitása szempontjából. Az MPC-alapú módszerek fő nehézsége azonban a számítási teher csökkentése, hogy időhatékonyabbá váljanak.

Egy kutatás során [19] olyan autonóm járművek szabályozására fókuszáltak, amelynek célja, hogy a jármű minél gyorsabban teljesítse a pályát, miközben a pálya határain belül marad és figyelembe veszi a fizikai korlátokat, például a jármű sebességét és kanyarodási képességét. A szerzők egy nemlineáris modellprediktív szabályozáson alapuló módszer során egy egyszerűsített bicikli modellt használtak, ahol az MPC felelt a mozgástervezés közben való megfelelő előretekintésért. Az MPC-t átalakították és egy időminimalizálási problémát fogalmaztak meg:

$$T = \int_{t_0}^{t_f} 1 \, dt = \int_{s_0}^{s_f} \frac{1}{\dot{s}(\tau)} \, d\tau \quad (2.1)$$

Egy másik tanulmány [20] a hibrid versenyautókban alkalmazott belső égésű motor és az elektromos motor/generátor teljesítményének megosztásával foglalkozik, amely kulcsfontosságú a legjobb köridő elérése érdekében. A Formula 1-ben alkalmazott hibrid hajtás-láncok esetében ez azért különösen lényeges, mert az elektromos motor fékezéskor energiát nyer vissza, míg gyorsításnál extra elektromos teljesítményt ad a rendszerhez. Ez a kettős funkció lehetőséget biztosít arra, hogy az energiafelhasználás optimalizálásával a belső égésű motor és az elektromos motor teljesítményét dinamikusan, a verseny közbeni helyzethez igazítva osszák meg. Így a versenyautó nemcsak a teljesítményt, hanem az üzemanyag- és energiahatékonyságot is maximalizálhatja, ami különösen fontos a versenyek hosszú távú sikeressége szempontjából.

A tanulmány által bemutatott eljárás egy két szintű Model Predictive Control (MPC) stratégiára épül, amely először egy magas szintű algoritmus segítségével előre kiszámítja a teljes körre vonatkozó optimális energiafelhasználási stratégiát. Ezt követően egy alacsony szintű MPC gondoskodik arról, hogy a jármű valós időben is képes legyen követni ezeket a stratégiai célokat, még váratlan helyzetekben, például hirtelen fékezés vagy tapadási viszonyok változása esetén is.

Az ilyen kutatások jól szemléltetik az MPC sokoldalú alkalmazhatóságát, különösen a mozgástervezési és energiaoptimalizálási problémák megoldásában. Bár a tanulmányok rávilágítanak arra, hogy az MPC milyen hatékony eszköz lehet a modern hibrid hajtás-láncok vezérlésében, én a kutatásomban inkább egy másik megközelítést fogok bemutatni, amely szintén előrejelző lehet ezen a területen. A következő alfejezetben az általam alkalmazott numerikus optimalizáción alapuló módszereket ismertetem.

2.4.3. Numerikus optimalizáción alapuló módszerek

Az ismertett MPC szabályozási módszer, amely előrejelzi a rendszer jövőbeli állapotait egy adott időhorizonton belül, az irányítási problémákat egy folyamatos optimalizációs problémává alakítja át, melyhez olyan numerikus optimalizációs módszerek, mint az SQP, SLSQP és hasonló módszerek rendkívül fontos eszközök, hiszen segítenek az optimális parancsok meghatározásában korlátok és célfüggvények figyelembevételével. Az SQP egy kvadratikus programozási (QP) problémát alkalmaz a nemlineáris programozási (NLP) probléma közelítésére egy iterációnál. Mely esetén x^k a jelenlegi iteráció, és egy keresési irány d meghatározása során, találjuk meg a következő iterációt x^{k+1} :

$$x^{k+1} = x^k + \alpha^k d \quad (2.2)$$

ahol α^k a lépésmagyságot jelöli, amely a következő iteráció optimális megoldásához közelebb vezet. Ennek eredményeként egy iterációs sorozat $\{x^k\}$ jön létre, amely feltehetően konvergál az optimális megoldáshoz x^* [21].

Az SQP módszert először 1963-ban [22] fejlesztették ki, majd más kutatók sorozatos hozzájárulásai népszerűsítették [23]. Az általuk használt algoritmusban a kvázi-Newton módszert alkalmazták a Lagrange-függvény Hesse-mátrixának közelítésére a QP alproblémában, amely a következő formában írható fel:

$$L(x, \lambda) = f(x) + \lambda^T g(x) \quad (2.3)$$

Ahol $f(x)$ a célfüggvény, $g(x)$ pedig a korlátozások vektora. A cél a Hesse-mátrix közelítése:

$$H \approx \nabla^2 L(x, \lambda) \quad (2.4)$$

Az SQP egy változata az SLSQP, melyet először [24]-es kutatásban javasoltak. Az SLSQP ahelyett, hogy minden fő iterációban egy QP részproblémát oldana meg, egy egyenértékű legkisebb négyzetek (LSQ) részproblémát old meg a keresési irány generálásához. Az SLSQP módszer képes a több változóból álló függvény minimalizálására, korlátok, egyenlőségi és egyenlőtlenségi korlátok tetszőleges kombinációjával, ezért az általam megoldani kívánt probléma során az eredetileg [25]-ös kutatásban bemutatott algoritmusra építkeztem.

Az általam kipróbált módszerek közé tartozik még a BFGS (Broyden-Fletcher-Goldfarb-Shanno) egy kvázi-Newton módszer is. A BFGS-módszer előnye, hogy nem igényli a Hesse-mátrix explicit számítását, helyette iteratíván közelíti azt, így gyorsabban konvergálhat, mint egy hagyományos gradient módszer. Azonban a módszer memóriaigényes mivel tárolja a teljes Hesse-mátrix közelítést. Ezen hátrány kiküszöbölése érdekében fejlesztették ki az L-BFGS módszert, mely a BFGS egy memória-optimalizált változata, amely csak

a legutóbbi iterációk adatait tárolja, így jelentősen csökkenti a memóriaigényt. Ezáltal alkalmas nagyobb méretű problémák megoldására is, amelyeknél a teljes Hesse-mátrix tárolása túl költséges lenne. Azonban az SLSQP a kvadratikus programozást (QP) használva jobban illeszkedik az optimális trajektória tervezés probléma megoldása során megkövetelt folyamatos újraszámítási eljárásokhoz is. Illetve jobban használhatónak bizonyult olyan problémák megoldására, ahol egy irányítási feladat korlátjai nagy jelentőséggel bírnak.

2.5. Visszacsatolt szabályozás

A negyedik szinten a tervezett pálya és trajektória végrehajtására kerül sor. E folyamat során a visszacsatoló szabályozó kiszámítja a megfelelő aktuátor bemeneteket a jármű vezetéséhez, kormányzásához és fékezéséhez, így biztosítva a tervezett mozgás végrehajtását, miközben minimalizálja a követési hibákat. Számos módszer van a megtervezett trajektória követésének kivitelezésére. Az egyik legegyszerűbb a Pure Pursuit Controller [26], ez a módszer az egyik legkorábban alkalmazott megoldások közé tartozik és használata is széles körű. Az aktuális helyzet és a definiált pálya alapján határozza meg a jármű mozgását, melyhez geometriai megközelítést alkalmaz, ahol körív illesztéssel történik meg a jármű és egy meghatározott pont között a kívánt kormányászög meghatározása. Egy sokkal bonyolultabb szabályozók közé tartozik: a Lyapunov-alapú szabályozás [27], ahol egy matematikailag megalapozott módszer, amely a rendszer stabilitásának elemzésére és biztosítására épít. Az autonóm járművek irányításában ez a szabályozási technika Lyapunov-függvényeket használ annak érdekében, hogy meghatározza, hogyan alakul a rendszer stabilitása az időben. Ha találunk egy olyan Lyapunov-függvényt, amely minden időpontban csökken, akkor a rendszer stabilnak tekinthető. Ez a megközelítés biztosítja, hogy a jármű folyamatosan a kívánt pálya felé konvergáljon, anélkül, hogy letérne arról vagy instabil mozgást végezne.

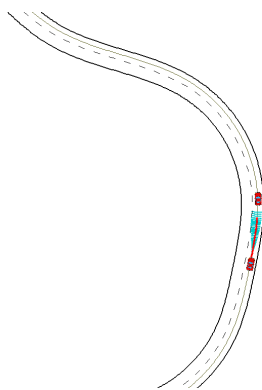
Egy sokkal rugalmasabb megközelítés az LPV szabályozás [28], amely a nemlineáris rendszerek lineáris modelljeinek alkalmazására épít, ahol a modellek paraméterei időben változnak. Az LPV módszer abban különleges, hogy képes valós időben alkalmazkodni a jármű és a környezet változó körülményeihez, például a sebesség, tapadási viszonyok vagy a jármű dinamikai jellemzőinek módosulásaihoz. Ez a szabályozó folyamatosan frissíti a rendszermodellt, figyelembe véve a jármű pillanatnyi állapotát és a környezetből származó visszacsatolásokat, így garantálva, hogy a jármű mindig az optimális irányítójelet alkalmazza. Az MPC, melyet már részletesebben tárgyaltam, mint szabályozási módszer is megjelenik, főként aktuátor szabályozóként. Az MPC egyik legnagyobb előnye a többi módszerrel szemben, hogy több célfüggvényt is minimalizálhat, például az utazási időt vagy az üzemanyag-felhasználást, miközben biztosítja a jármű ütközésmentes haladását. Ezek a visszacsatolt szabályozási módszerek mind biztosítják a jármű pontos és stabil irányítását, működésüket és számítási igényüket tekintve más felhasználási módjuk van.

3. fejezet

Hierarchikus trajektóriatervezés megvalósítása

3.1. Szimulációs környezet

A fejlesztéshez és teszteléshez egy egyedi szimulációs környezet került kifejlesztésre, amely a következő komponensekből áll: Fő eleme egy útgenerátor, mely megfelel a szabványos úttervezési irányelveknek. Az útvonalat egyenesekből, kanyarokból és klotoid átmeneti ívekből építi fel. Különböző környezeteket hozhatunk vele létre, például közúti környezetet két sávval, de jelen tanulmány során versenypályaszerű egy sávos utakat hoztam létre. A pályán végig vezetett jármű leírásáért egy nemlineáris síkbeli járműmodell felel, mely egy dinamikus kerékmodellt is tartalmaz [29]. A jármű oldalirányú irányításáért egy MPC szabályozó felel: Ez a szabályozó egy lineáris dinamikus modellt használ. A görbületi sebesség profilja kényszerfeltételként van meghatározva. Hosszirányban egy PID szabályozó felel a jármű sebességének szabályozásért, mely esetében a sebességreferenciát a mozgástervező sebességprofilja határozza meg. Továbbá különböző transzformációs megoldásokat is rejt a szimulációs környezet, melyek a Frenet koordinátarendszerben való tervezés miatt szükségesek. Maga a vizualizáció egy 2D grafikus felületen valósul meg 3.1.

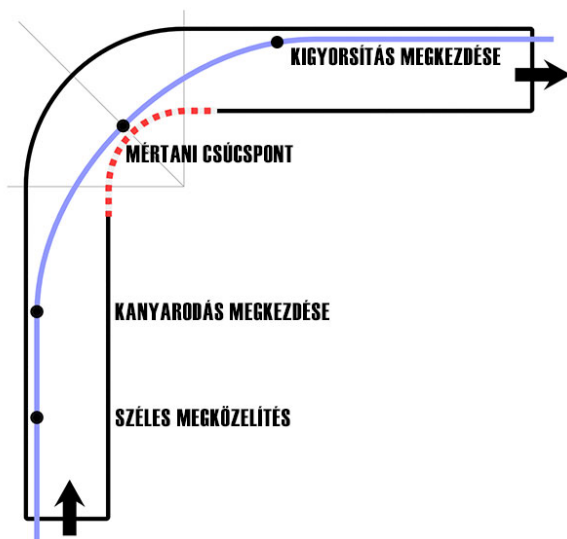


3.1. ábra. 2D grafikus felület

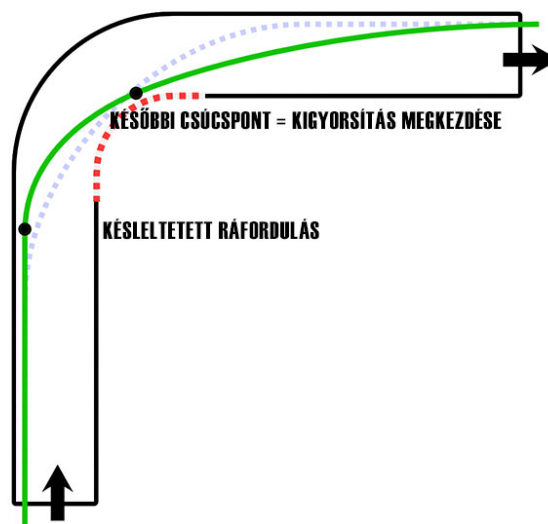
3.2. Geometria alapú globális útvonaltervezés

3.2.1. Kanyar csúcspont és a versenyívek kapcsolata

Az elérhető maximális sebesség a kanyar sugara négyzetgyökével növekszik, így nagyobb sugarú ív esetén a jármű gyorsabban tud átmenni a kanyaron anélkül, hogy kisodródna. Ebből következik, hogy a legjobb időeredmény eléréséhez a járművet a legnagyobb lehetséges íven kell vezetni a kanyarban. Mivel az útgenerátor ideális ívekből és klotoid átmeneti ívekből állítja össze az utat, ahogyan azt a modern pályatervezés megköveteli, ezáltal a jármű konstans sebességgel haladva az optimális pályán, ami ebben az esetben az út középvonala, lehetővé teszi, hogy konstans sebesség esetén ez legyen a leggyorsabb útvonal. Ezáltal a kormányszög és az oldalirányú gyorsulás változása lineáris kapcsolatban lesz az út görbületével. Ennek eredményeképp a jerk minimális vagy közel nulla lesz, ami fontos a stabilitás és az utas komfortja szempontjából is. Azonban egy versenyhelyzetben a sebesség szinte soha nem marad állandó, hiszen a jármű gyakran gyorsít és lassít, például előzési manőverek, kanyarodási technikák vagy akár gumikopás és üzemanyag fogyasztása miatt. Ezért a kanyarokba érkezve lassít, illetve az egyenesekből gyorsítva a versenyautó sebessége folyamatosan változik. Amint a hosszirányú sebesség nem konstans, onnantól kezdve az útgenerátor által előállított ívek már nem lesznek ideálisak, a jármű kormányszög hibája és oldalirányú gyorsulása is megszűnik lineárisnak lenni, ezért egyéb optimális íveket kell meghatározni, ha arra törekszünk, hogy minél jobb köridőt érjünk el. Ezért a kanyar csúcspont meghatározása és pontos eltalálása kritikus fontosságú a versenyautózásban, mivel közvetlen hatással van a kanyarvételi sebességre és a kanyarból való gyorsulásra, amelyek együttesen meghatározzák a pályán elért időeredményt [30]. Jobban megvizsgálva, a kanyar csúcspont (vagy apex), a kanyar mértani középpontjában helyezkedik el, itt halad a jármű legközelebb a kanyar belső széléhez 3.2-3.3.



3.2. ábra. Elméleti kanyarív



3.3. ábra. Verseny kanyarív

Ez az autóversenyzésben rendkívül fontos szerepet játszik, mivel az apex helyes megválasztása döntő szerepet játszik, hogy a jármű milyen sebességgel tud kilépni a kanyarból és milyen ívet tud lekövetni a kanyar teljes szakaszán. Az ideális versenyvonal (racing line) az elméleti kanyarívet némileg módosítja. A klasszikus versenyvonalnál, avagy verseny kanyarív esetén az apex a geometriai középpont után helyezkedik el, azaz később érhető el. Ez a „késői apex” segít előkészíteni a kanyarból való gyorsabb kilépést. Ennek az oka, hogy a nagyobb sugarú ív kisebb oldalgyorsulást (centripetális erőt) eredményez, így a jármű magasabb sebességgel tud áthaladni a kanyaron. Az ív ennek megfelelően úgy alakul, hogy a kanyar második felében a sugár növekszik, lehetővé téve a gyorsítást, míg az első felében kissé szűkül az ív, ami elősegíti a precízebb kanyarbejáratot.

3.2.2. Globális útvonal előállítása

A globális útvonal előállításához első sorban az egyes útszakaszokat szegmentáltam, és minden szakaszhoz hozzárendeltem az út középvonalának koordinátáit, az állásszöget, a görbületet, illetve az adott szakasz típusát (egyenes vagy kanyar). A kanyarok esetében az ív irányát is rögzítettem, amely biztosítja, hogy a pálya minden pontján pontos ismeretünk legyen az út tulajdonságairól és geometriájáról.

A kanyarok geometriai alapú optimális ívének előállítása során a pálya középvonalát követő alapív generálása után meghatároztam a kanyarszegmensekhez kapcsolódó kulcsponthoz tartoznak a kanyarok csúspontjai, valamint a kanyarba való belépési és kilépési ívek külső pontjai. Mivel a kanyarokat klotoid alapú átmeneti ívek kapcsolják az egyenes szakaszokhoz, a görbület változásai a következő módon jelennek meg: az átmeneti ívek görbülete lineárisan növekszik vagy csökken, míg a kanyar középső szakaszában a görbület állandó. Ennek köszönhetően a kanyar középső, állandó görbületű részéből

meghatározható a kanyar csúcspontja, míg a klotoid átmeneti ívek segítségével pontosan kijelölhetőek a be- és kilépési ívek pontjai.

Az optimális ív kialakítása érdekében ezeket a pontokat a pálya következő vagy előző kanyarjának geometriája alapján módosítottam, hogy a jármű a legideálisabb pozícióból közelíthesse meg a kanyart, így a kilépési szakaszokon maximális gyorsítást érhet el. A cél az, hogy különböző sebességek mellett a legkisebb görbületű ívet határozzam meg, amely a stabilitás és a gyorsítás szempontjából előnyös a jármű számára. A geometrikus elméleti kanyarívet a 3.2 ábra szemlélteti, mely során a ív az egyenesek külső íveit köti össze a mértani csúcsponttal. Ezen globális útvonaltervező ezt valósítja meg az említett pontokra való harmadfokú (kübös) spline illesztésével. A spline-illesztés célja az volt, hogy sima, folytonos ívet biztosítson a jármű számára, amely követi az elméleti optimális kanyarok ívét és az egyenes szakaszokat. A spline-illesztést az alábbi módon definiáltam [31]. Tekintsük a $P_i = (x_i, y_i)$ pontokat, ahol $i = 1, 2, \dots, n$ jelöli az egymást követő útpontokat. Ebben az esetben a kübös spline függvény $S(x)$ minden szakaszra egy harmadfokú polinomot használ az egyes pontok közötti sima átmenet biztosítására a következő formában:

$$S(x) = \begin{cases} s_1(x) & \text{ha } x_1 \leq x < x_2 \\ s_2(x) & \text{ha } x_2 \leq x < x_3 \\ \vdots & \vdots \\ s_{n-1}(x) & \text{ha } x_{n-1} \leq x < x_n \end{cases} \quad (3.1)$$

ahol s_i egy harmadfokú polinom, amelyet az alábbiak szerint definiálunk:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad x \in [x_i, x_{i+1}] \quad (3.2)$$

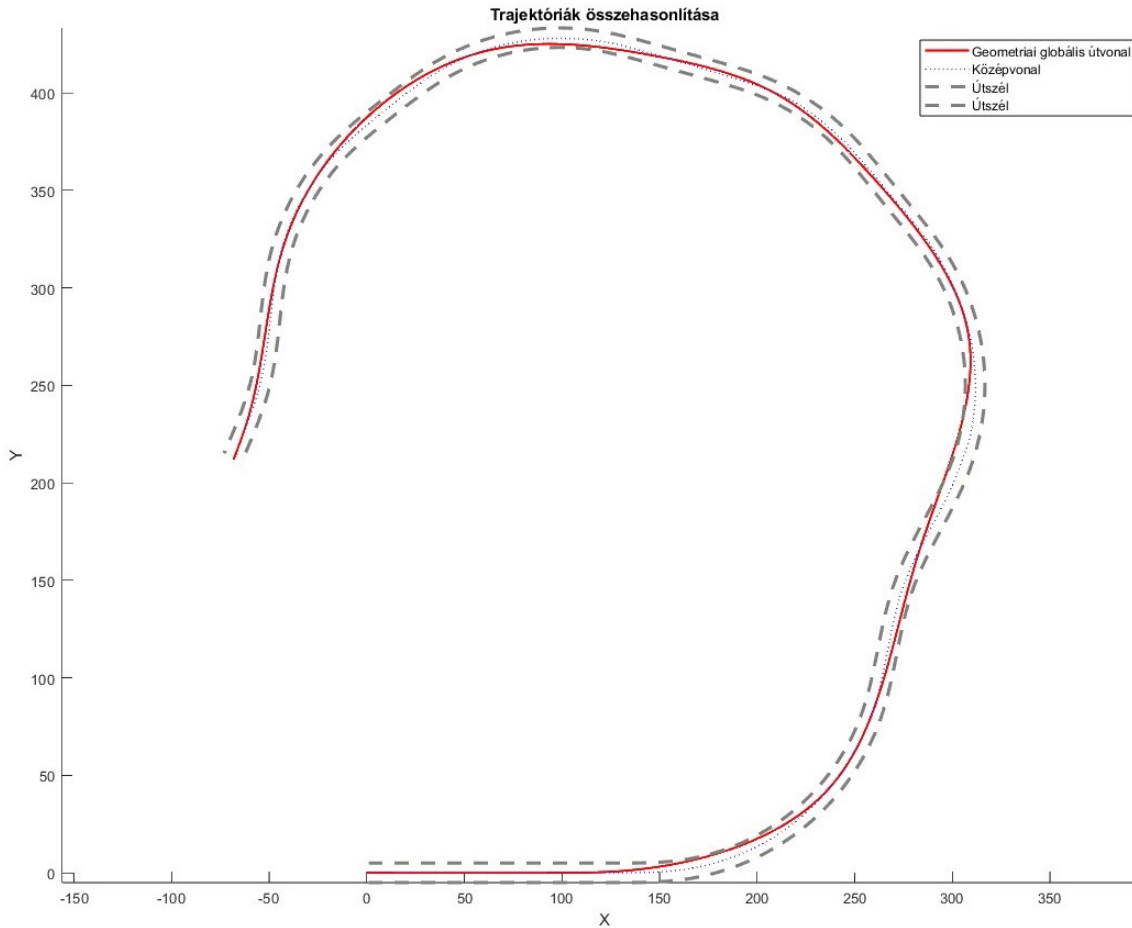
,ahol $i = 1, 2, \dots, n - 1$. Az első és második deriváltjai ennek az $n - 1$ egyenletnek alapvető fontosságúak a folyamat szempontjából, és ezek a következők:

$$s'_i(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \quad (3.3)$$

$$s''_i(x) = 6a_i(x - x_i) + 2b_i \quad (3.4)$$

ahol $i = 1, 2, \dots, n - 1$.

Az így előállított globális útvonalat, mely az elméleti kanyaríveket valósítja meg $a(z)$ 3.4 ábra szemlélteti. Ezt geometrikus globális útvonalat lokális mozgástervezővel $a(z)$ 3.3 ábrának megfelelően kívántam tovább finomítani, melyet a későbbiekben részletesen bemutatok.

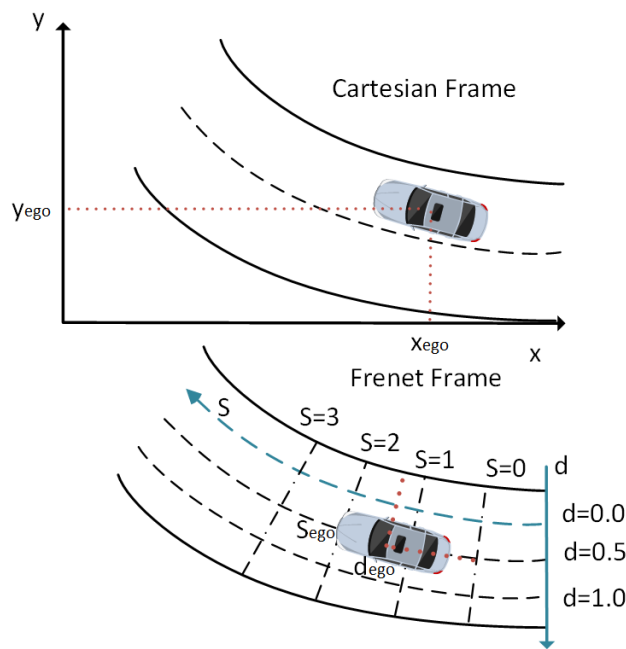


3.4. ábra. Globális útvonal az elméleti kanyarívnek megfelelően

3.3. Lokális mozgástervezés

3.3.1. Frenet koordinátarendszer

A Frenet Frame koordinátarendszerben való autonóm járművek útvonaltervezésének megközelítése gyakran alkalmazott módszer a járművek mozgásának leírására az útvonal mentén [32]. Ezen módszertan segítségével egy járművel mozgó vonatkoztatási rendszer határozható meg, mely normál és érintőleges vektorokkal írja le a görbe egy-egy pontját, melyet a koordinátarendszer referenciavonalának nevezünk.



3.5. ábra. Frenet koordinátarendszer

Ezen referenciavonal tetszőlegesen megválasztható a problémánktól függően, lehet akár egy megtervezett ideális globális útvonal, vagy az úttest középvonala. A kutatásom során ezen referenciavonalként a globális trajektóriatervező kimenete szolgál. A járművet $a(z)$ 3.5 ábra alapján pozicionáltam az s tengely mentén, a referenciavonallal párhuzamosan, míg a d tengely esetén erre merőlegesen. Ez a megközelítés pontosan meghatározza a jármű pozícióját, mert a lokális vonatkoztatási rendszerrel együtt mozog a jármű, így az akadályok, sávhatárok és különböző korlátok jobban kezelhetők. A Frenet használata esetén negyed- és ötödfokú polinomok segítségével határoztam meg a kívánt trajektóriát.

Azonban a mozgó koordinátarendszer alkalmazása hátránnyal is jár, számottevő transzformációs költségeket von maga után, mert a jármű aktuális orientációját és pozícióját folyamatosan oda-vissza kell transzformálni a globális Descartes-féle derékszögű koordinátarendszerről. Annak ellenére, hogy ez hátrányt jelentett, számomra pontosabbnak és jobban alkalmazhatónak bizonyult, ezért a globális útvonal követése közben a lokális útvonalat Frenet koordinátarendszerben számítottam ki.

3.3.2. Trajektória meghatározása negyed- és ötödfokú polinom illesztéssel

A lokális trajektóriatervezésnek különböző követelményeknek kell megfelelnie. A tervezés végrehajtási idejének elég rövidnek kell lennie ahhoz, hogy valós időben irányíthassa a járművet. A jelen tanulmány során nem valós idejű megoldás kerül bemutatásra, de ezen követelmény betartására törekszem. Továbbá az útvonal görbületének és sebességének időben folytonosnak kell lennie, ezért a módszernek olyan trajektóriát kell előállítania, amely a sebességváltozást is képes kezelni. A probléma megoldására ezért olyan interpolációs megoldások, mint a spline és klotoid alapú módszerek nem optimálisak, mivel ezek csak állandó sebességnél biztosítják a görbület folytonosságát. Az említett követelmények teljesítése érdekében egy olyan mozgástervezési módszert alkalmaztam, amely a trajektória pozícióját és sebességét koordináltan tervezi.

Egy olyan általam numerikus optimalizációval továbbfejlesztett negyed- és ötödfokú polinomillesztési eljárást alkalmaztam, amely képes az útvonal síkbeli leírására, lehetővé téve többféle feltétel, például a sebesség, gyorsulás vagy a globális útvonal fenntartásának figyelembevételét [33].

Az alap algoritmus működésének alapelvei a következők:

Az alap trajektóriatervező algoritmusom a Jerk költségének minimalizálására törekszik, mivel a módszer célja, hogy a lokális útvonalat a hirtelen gyorsulásváltozások korlátozásával alakítsa ki. A költségfüggvény a következőképpen írható fel:

$$C = \frac{1}{2} \int_0^T L, dt \quad (3.5)$$

ahol a teljesítmény index :

$$L = \left(\frac{d^3 x}{dt^3} \right)^2 + \left(\frac{d^3 y}{dt^3} \right)^2 \quad (3.6)$$

Ebben az esetben az (x) és (y) az aktuális jármű pozícióját határozza meg az (x-y) fix koordinátarendszerben. A cél a lokális útvonal $(x(t), y(t))$ során fellépő költségfüggvény minimalizálása. A polinomok az Euler-egyenletekből vezethetők le, amelyeket részletebben tárgyal [33]:

$$d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (3.7)$$

$$s(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 \quad (3.8)$$

Az útvonalat a fenti polinomokkal írtam le a Frenet koordinátarendszerben bemutatott d- és s-tengelyek mentén ötödfokú és negyedfokú polinomokat használva, amelyek a globális pályától való eltéréseket tükrözik. Az oldalirányú jellemzők meghatározására ötödfokú polinomillesztést használtam. Ez a polinom leírja az oldalirányú helyzet válto-

zását az idő függvényében, ahol az 1., 2., 3. és 4. deriváltak határozzák meg a jármű mozgási tulajdonságait minden egyes időlépésnél. Míg az együtthatók a peremfeltételek alapján kerülnek meghatározásra, például a kezdeti és véghelyzet, sebességek, valamint gyorsulások figyelembevételével. Az első tag, a konstans, a polinom kezdeti állapotát jelzi, vagyis meghatározza, hogy az idő (t) pillanatában hol kezdődik az útvonal. Az elsőfokú tag a sebességet képviseli, a második a gyorsulást, a harmadik pedig a jerk-et, azaz a gyorsulás változásának ütemét. A negyedfokú tag a jolt-hoz kapcsolódik, amely a jerk változásának sebességét mutatja. Végül az ötödik tag a jounce-ot képviseli, ami a jolt változásának sebessége. A hosszirányú jellemzők meghatározása negyedfokú polinom segítségével történik, hasonlóan az oldalirányhoz.

Az algoritmus iteratív módon működik az útvonal követése során. A kívánt sebesség a következő képlettel számítható:

$$v_s = \sqrt{\dot{x}^2 + \dot{y}^2} \quad (3.9)$$

Az útvonal görbületét pedig az alábbi képlet alapján lehet meghatározni:

$$\kappa = \frac{a_d}{v_s^2} \quad (3.10)$$

ahol a_d a jármű oldalgyorsulása. Az útvonaltervezés során megadhatók különböző paraméterek, mint például a kezdő- és végpozíció, sebesség vagy gyorsulás.

Ezen módszer egyik hátránya, hogy az MPC-vel (Model Predictive Control) és az RL-lel (Reinforcement Learning) ellentétben nem képes előre látni és figyelembe venni a jövőbeli információkat. Ez azért van, mert csak egyetlen lokális trajektóriát tervez meg a globális útvonal mentén. Azonban a tanulmány során ezt a módszert numerikus optimalizációval kiegészítettem, amely lehetővé teszi több egymást követő, összefűzött lokális trajektória meghatározását. Ezek a trajektóriák a globális útvonalat alapul véve ideálisabb íveket írnak le, és meghatározzák a jármű sebességprofilját is.

Az algoritmus ezek figyelembevételével képes a lokális útvonal megtervezésére, miközben minimalizálja a polinomok együtthatóinak kombinációját a kényszerek teljesítése érdekében. Az együtthatók révén lehetőség nyílik a különböző kényszerek beállítására és a lokális útvonal meghatározására.

3.4. Globális útvonal alapján lokális trajektóriatervezés numerikus optimalizációval

A geometriai módszerrel meghatározott globális útvonal (racing line) lokális trajektóriával való optimalizálásának célja, hogy a jármű a pálya egyenes szakaszain a lehető legnagyobb sebességgel haladjon, míg a kanyarokban a belső íven történő gyors irányváltás és a kanyarból való kilépés minimális sebességvesztéssel történjen. A globális útvonal az elméleti kanyaríveket követi, azonban az optimalizáció során ezeket az íveket a lokális trajektóriák finomítják, miközben meghatározzák az ideális sebességprofil is.

Ez úgy valósul meg, hogy a bemutatott lokális trajektória tervező polinomjainak együtthatói révén lehetőség nyílik a különböző kényszerek beállítására és a lokális útvonal pontos meghatározására, akár úgy is, hogy jelentősen eltérjen a globális útvonaltól. Az optimalizáció során a polinom együtthatói, illetve a trajektória végén elérni kívánt sebesség kerül meghatározásra. A trajektóriák tervezési időhorizontjának 5 másodpercet választottam. Ebben az esetben a trajektóriák az adott sebességgel ezen idő alatt megtehető úttal egyenlő hosszúságúak lesznek. Az optimalizáció során kettő egymáskövető trajektória paramétereit határoztam meg, a következő költségfüggvény segítségével, ahol a jelöléseket a frenet koordinátarendszernek megfelelően tüntettem fel, $s(t)$ hosszirányú, $d(t)$ oldalirányú relativ pozíciókat jelentik:

Az oldalirányú eltérési költség célja a jármű eltérésének minimalizálása a referenciaútvonaltól. Az eltérést az oldalirányú pozíció ($d(t)$) alapján számítottam ki.

$$J_{\text{lat}} = \text{cost_lat_error} \cdot \sum_t |d(t)| \quad (3.11)$$

A hosszanti sebesség költség a célsebességtől való eltérések büntetésére szolgál. Különösen akkor van nagyobb súlya, ha a tényleges sebesség alacsonyabb, mint a célsebesség.

$$J_{\text{speed}} = \frac{1}{2} \cdot \text{cost_lon_v} \cdot \sum_t (\text{desired_speed} - s'(t))^2 \quad (3.12)$$

A jerk költség minimalizálja a hirtelen gyorsulásokat, hogy sima vezetési élményt biztosítson. Itt az oldalirányú pozíció ($d(t)$) harmadik deriváltját ($d'''(t)$) vettem figyelembe.

$$J_{\text{jerk}} = \text{cost_lat_ddd} \cdot \sum_t |d'''(t)|^2 \quad (3.13)$$

A maximális hosszanti gyorsulás költség a hirtelen gyorsítások és lassítások korlátozását szolgálja, amit az $s(t)$ második deriváltja, vagyis a gyorsulás ($s''(t)$) maximalizálásának büntetése révén értem el.

$$J_{\text{accel_max}} = 0.05 \cdot \max(s''(t)) \quad (3.14)$$

Az összesített költségfüggvény az egyes részkomponensek összegeként adódik, és célja az egymást követő trajektóriák teljes költségének minimalizálása. Az optimalizáció ennek megfelelően nem csupán az egyes trajektóriák költségét veszi figyelembe önállóan, hanem összetett, előretekintő módon értékeli az egymást követő ívek és szakaszok összköltségét.

Ezáltal a polinom együtthatók és a sebesség is olyan paraméterekként kerülnek meghatározásra, amelyek figyelembe veszik az útszakasz előrejelzett jellegét. Ha például egyenes szakasz után kanyar következik, az optimalizáció úgy alakítja ki a jármű ívét és sebességét, hogy megfeleljen a versenyívekre jellemző optimális költségfüggvénynek.

Ez azt eredményezi, hogy a jármű lassít a kanyarba való belépéskor, majd a kanyarból kifelé haladva kisebb görbületű trajektóriát követ, nagyobb sebességprofillal, ahogy azt a gyakorlati versenyívek diktálják. Az így kialakított trajektória nemcsak a leggyorsabb köridőt célozza meg, hanem a teljes pályaszakasz összhatékonny és folyamatos haladását is biztosítja.

$$J = J_{\text{lat}} + J_{\text{speed}} + J_{\text{jerk}} + J_{\text{accel_max}} \quad (3.15)$$

A költségfüggvény minimalizálásához a Python programozási nyelv SciPy könyvtárát használtam, amely a tudományos számítások és optimalizációk széles körét támogatja. A korábban bemutatott SLSQP (Sequential Least Squares Quadratic Programming) módszert alkalmaztam, amely többváltozós függvények minimalizálására szolgál, és lehetőséget nyújt tetszőleges korlátok, egyenlőségi és egyenlőtlenségi feltételek kezelésére [25].

A korlátok definiálásakor megadtam a polinomiális együtthatók és a sebesség alsó és felső határait, ezzel gyorsítva az optimalizálási folyamatot. Mivel a jármű dinamikai jellemzőit a modell nem veszi figyelembe, a sebesség felső határát 90 km/h -ban maximalizáltam, így biztosítva, hogy az optimális megoldás megfeleljen a valósághoz közelítő sebességparamétereknek. Az alsó sebességhatárt 60 km/h -ban állítottam be, figyelembe véve, hogy az útgenerátor által létrehozott kanyarok szabványosítása ehhez a sebességhez készült.

4. fejezet

Eredmények

	Középvonalon haladás				Hierarchikus tervezés				Geometriai globális útvonalon haladás			
	Max slip	Átlag old.gyorsulás	Átlag sebesség	köridő	Max slip	Átlag old.gyorsulás	Átlag sebesség	köridő	Max slip	Átlag old.gyorsulás	Átlag sebesség	köridő
1	0.042	0.25	81.37	44.55	0.084	0.527	84.025	42.75	0.047	0.172	77.044	46.8
2	0.03	0.124	77.678	17.55	0.028	0.169	79.298	17.1	0.019	0.086	77.31	17.55
3	0.043	0.262	80.124	35.7	0.041	0.298	83.484	34.2	0.041	0.178	79.614	35.85
4	0.03	0.135	76.583	31.8	0.05	0.211	81.872	29.55	0.036	0.094	76.167	31.8
5	0.041	0.216	79.333	32.1	0.073	0.326	82.589	30.6	0.054	0.136	78.045	32.4
6	0.028	0.137	77.503	24.75	0.036	0.217	78.715	24.45	0.037	0.184	79.614	24.15
7	0.031	0.143	80.946	22.65	0.041	0.283	80.991	22.65	0.035	0.13	79.227	23.1
8	0.043	0.254	80.532	32.7	0.096	0.445	81.17	32.4	0.052	0.116	76.128	34.5
9	0.042	0.189	77.73	46.35	0.063	0.45	82.814	43.35	0.045	0.18	78.696	45.6
10	0.03	0.133	77.891	26.25	0.033	0.161	81.887	24.9	0.026	0.085	75.12	27.15

4.1. ábra. Jármű karakterisztikák véletlenszerű pályákon különböző random seedekkel

Az optimalizáció teszteléséhez a módszert véletlenszerűen generált pályákon hasonlítottam össze a geometriai alapú globális útvonalon és a középvonalon történő haladással. Mindegyik vezetési stílus esetében meghatároztam a sebességprofil, de előretekintést nem alkalmaztam, az egyes trajektóriák által definiált sebességprofil volt a mérvadó.

A 4.1 táblázatban látható eredmények szerint az optimalizáció révén elért legjobb köridő egyértelműen mutatja, hogy ez az előretekintést alkalmazó korrekció a globális útvonalon kiemelkedő teljesítményt nyújt, anélkül, hogy jelentős kompromisszumot kötné a stabilitásban. A hierarchikus tervezés valamivel magasabb oldalirányú gyorsulást és slip értéket vesz fel, de ezek a különbségek mérsékeltek, és nem térnek el lényegesen a geometriai alapú vagy középvonalas megközelítéstől. Az optimalizáció során alkalmazott globális útvonaltervezés és előretekintés úgy növeli a csúszást és gyorsulást, hogy éppen csak annyira módosítja őket, amennyire szükséges a teljesítmény fokozásához, anélkül, hogy a stabilitás és kontroll rovására menne.

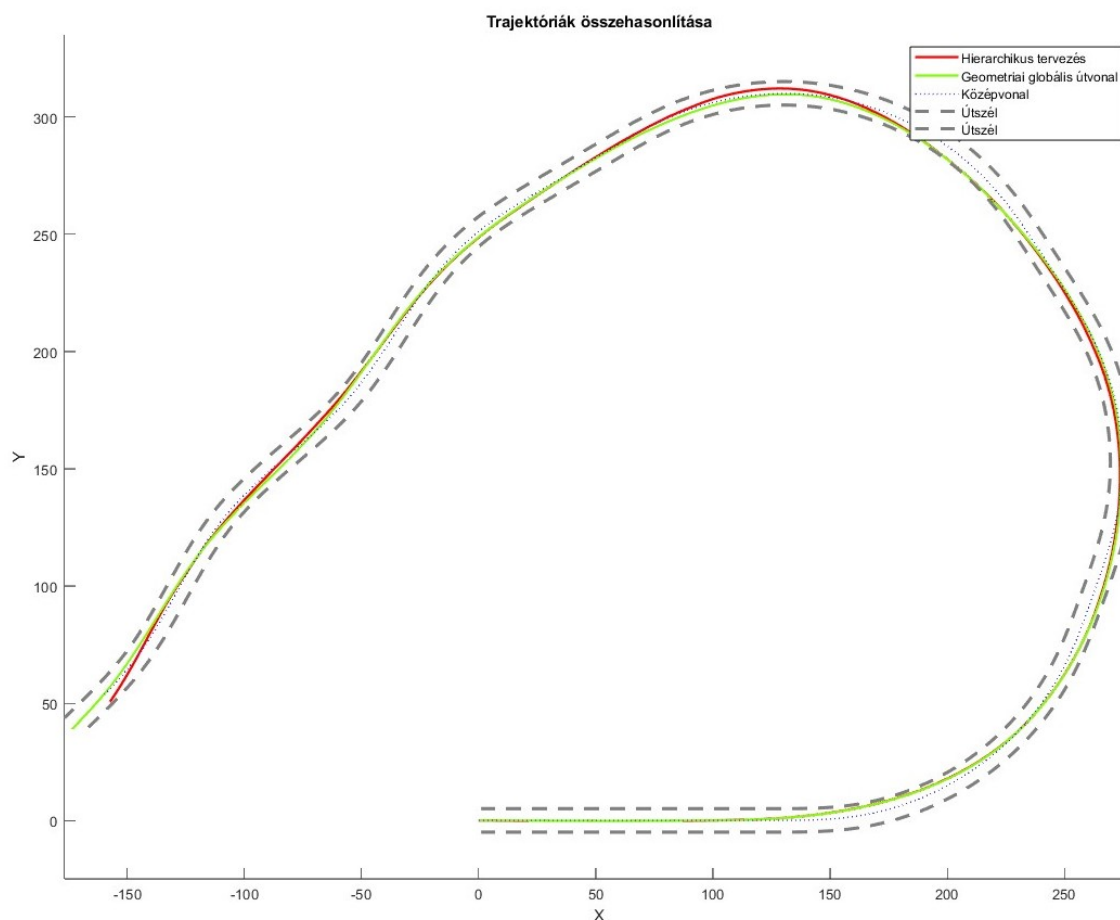
Amikor a geometriai globális útvonalat a középvonalon való haladással hasonlítom össze, megfigyelhető, hogy önmagában a geometriai trajektória – ha a sebességprofil tervezése során nem alkalmazunk előretekintést a numerikus optimalizáció révén – nem teljesít

jobban. Sőt, bizonyos esetekben még kedvezőtlenebb értékeket eredményezhet.

A 4.2 ábrán látható különböző útvonalak közül a pirossal jelölt optimalizált útvonal jól mutatja, hogyan sikerül tovább csökkenteni a megtervezett, zölddel jelölt geometriai útvonal görbületét, ezáltal közelebb hozva azt a gyakorlati versenyívek optimális íveihez. Ezzel szemben a kék pontozott vonallal jelölt középvonal egyszerűbb, a pálya görbületét veszi fel, ami azonban csak állandó sebesség esetén optimális, és nem alkalmaz előretekintést vagy finomhangolást.

Az optimalizált útvonal egyik fő előnye, hogy finomítja a kanyarodási szögeket a sebesség változásoknak megfelelően, így a jármű simább, kiszámíthatóbb ívet követ a pályán. Az optimalizált útvonal kevesebb hirtelen irányváltást igényel, ami nemcsak gyorsabb köridőket eredményez, hanem növeli a jármű stabilitását és csökkentheti a gumik kopását is.

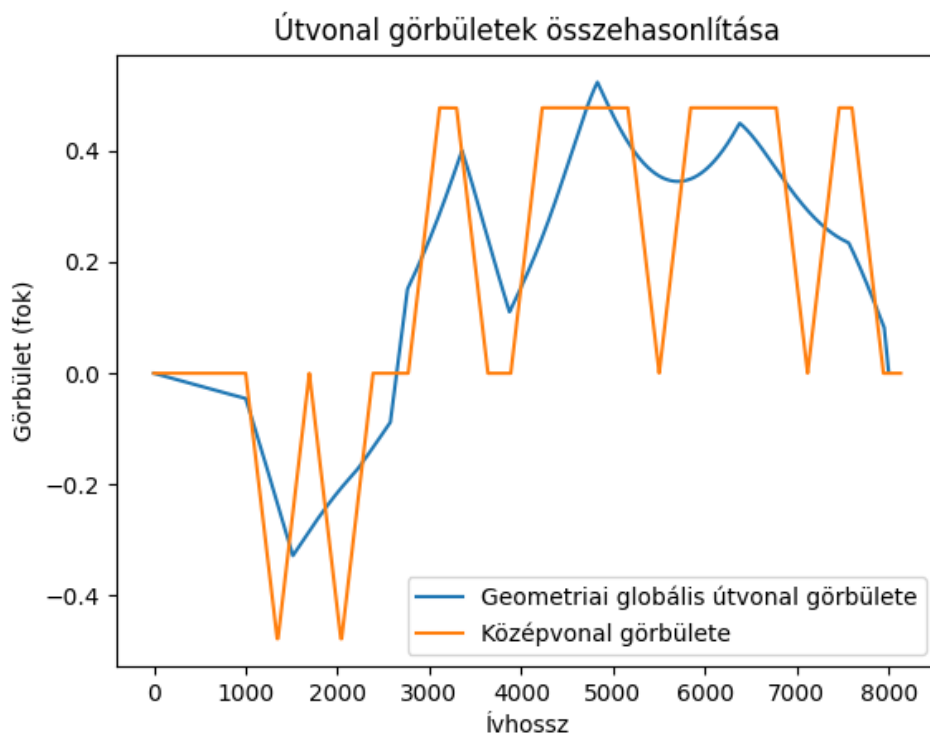
Ez az optimalizált útvonal tehát nemcsak a sebességet maximalizálja, hanem a versenyhelyzetek során ideális ívet követ, közelítve a legjobb gyakorlati versenytechnikákhoz. Az előretekintéssel és globális útvonaltervezéssel elért, finomított pályaívek olyan vezetési útvonalat biztosítanak, amely egyszerre növeli az irányíthatóságot és az optimális teljesítményt.



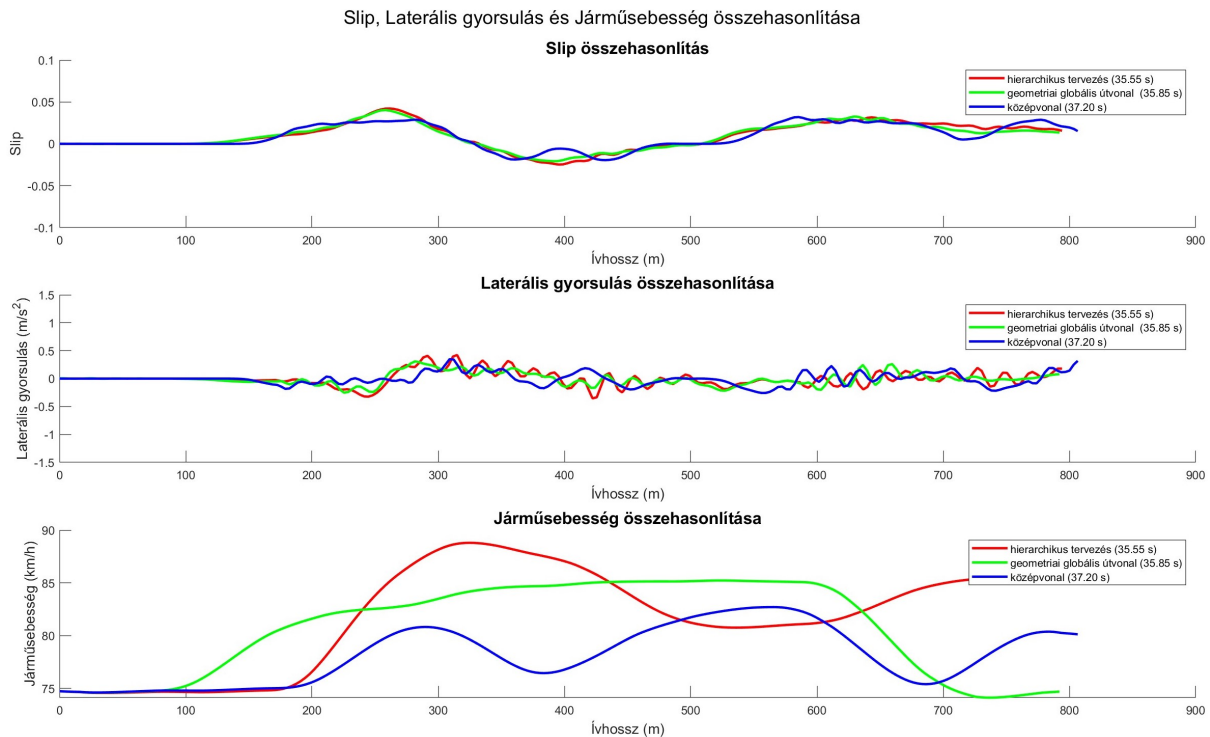
4.2. ábra. Optimalizált trajektória példa

A 4.3 ábrán a két különböző útvonal görbülete összehasonlítása látható. A kék görbe a megtervezett, geometriai alapú globális útvonalat mutatja, amelyet a jármű számára terveztünk, figyelembe véve a pálya görbületeinek csökkentését. A sárga görbe pedig a pálya középvonalának görbületét ábrázolja.

Az ábrán látható, hogy a sárga görbe klotoid átmeneti ívekből és állandó görbületű kanyarokból áll. Ez az útvonal állandó sebesség esetén optimális járműkarakterisztikát eredményez, ám versenypályán nem ideális. A kék színű, tervezett globális útvonal sikeresen simítja a középvonal kanyarjait: jól megfigyelhetők rajta a kanyarok csúcspontjai, ahol a görbület csökken. A kanyarhoz érkeve a görbület növekedése lassítást igényel, míg a csúcspontot elhagyva a görbület hirtelen csökken, lehetővé téve a jármű stabil és erőteljes gyorsulását, amely elősegíti a jármű hatékonyabb haladását.



4.3. ábra. Optimalizált trajektória görbülete



4.4. ábra. Jármű karakterisztika összehasonlítás

A 4.4 ábrán a különböző módszerekkel kapott jármű karakterisztikák – hierarchikus tervezés (piros), geometriai globális útvonal (zöld), és középvonal (kék) – összehasonlítása látható a slip, a laterális gyorsulás és a járműsebesség függvényében. A slip ábrán a hierarchikus tervezés jellemzően a geometriai esethez hasonló szélső értékeket vesz fel, azonban enyhébb ingadozásokat mutat. Ez stabilabb útfogást eredményez, mivel az ívhossz nagy részén magasabb sebességgel haladt a hierarchikus tervezés során a jármű.

A hierarchikus tervezés laterális gyorsulása a zöld geometriai globális útvonalhoz képest kevésbé ingadozik, avagy hasonlóan ingadozik magasabb sebesség értékek mellett is és jellemzően nem éri el a szélsőséges értékeket.

A járműsebesség ábrán látható, hogy a piros hierarchikus tervezés magasabb csúcsebességeket tesz lehetővé, különösen az egyenes szakaszokon. Ez azért van, mert az útvonal optimalizálásakor figyelembe veszi a kanyarokat, így a jármű optimális sebességgel közelítheti meg azokat, és gyorsabban érheti el a maximális sebességet a kanyar után.

Az ábrán feltüntetett köridők alapján a hierarchikus tervezés a leggyorsabb (35,55 s), míg a geometriai globális útvonal (35,85 s) és a középvonal (37,20 s) ennél lassabb. A rövidebb köridő bemutatja, hogy a hierarchikus tervezés optimálisabban használja ki az íveket és a gyorsulási lehetőségeket, így jóval hatékonyabb haladást biztosít.

5. fejezet

Konklúzió

A dolgozatban bemutatott hierarchikus trajektóriatervezési módszer sikeresen kombinálja a globális, geometriai alapú versenyív megközelítést egy lokális, optimalizáció-alapú finomítással, amely a trajektória és a sebességprofil dinamikus hangolására szolgál. Az elméleti versenyívet először globális útvonaltervező segítségével határoztam meg, majd a lokális tervezési folyamatban a pálya aktuális szakaszainak és kanyarjainak jellemzőihez igazítottam. A tervezett trajektóriákat és sebességprofilokat saját fejlesztésű szimulációs környezetben teszteltem és hasonlítottam össze két megközelítésben: a középvonalon való haladással, illetve a geometriai alapú versenyív mentén történő navigálással.

Az eredmények alapján a hierarchikus tervezési módszer jelentősen jobb köridőket, kisebb oldalirányú gyorsulásokat és alacsonyabb csúszási értékeket eredményezett. Ez különösen a kanyarokban mutatkozott meg, ahol az optimalizált versenyív a jármű stabilabb, és alacsonyabb görbülettel rendelkező ív haladását tette lehetővé, ezzel javítva a teljes pályaszakaszon a jármű irányíthatóságát és teljesítményét.

A módszer hatékonysága és a szimulációban nyújtott eredmények alapján ígéretes megoldást kínál autonóm versenyautók trajektóriatervezésére. Azonban a valós idejű alkalmazások során számításba kell venni a jármű fizikai karakterisztikáit és a vezetési környezet hatásait is. Ez azt jelenti, hogy az optimalizációs folyamatban előrejelzéseket szükséges beépíteni a jármű mozgásdinamikájára, például a tapadási jellemzők és az erőátviteli rendszer teljesítményére vonatkozóan, hogy a tervezett trajektória a valós körülmények között is optimálisan teljesítsen.

A jövőbeli célkitűzések közé tartozik a módszer validálása egy valós tesztkörnyezetben, illetve az IPG CarMaker szimulációs platformon, valamint a valós vezetők által leírt versenyívekkel történő összehasonlítás. Ezenkívül a jármű karakterisztikáinak pontos előrejelzése és ezek alapján történő trajektória-optimalizáció kulcsfontosságú lépést jelent a módszer valós alkalmazhatóságának további növeléséhez, amely végső soron a leggyorsabb köridők elérését és a biztonságos, folyamatos autonóm versenyzést célozza.

Továbbfejlesztési lehetőségként érdemes megvizsgálni az összkerék-kormányzás előnyeit is versenypályás környezetben, mivel a szimulációban használt jármű erre fel van ké-

szítve. Alacsony sebességnél a hátsó kerekek az elsőkel ellentétes irányú kormányzása javíthatja a manőverezhetőséget és csökkentheti a fordulókört, míg nagyobb sebességnél az elsőkel megegyező irányba forduló hátsó kerekek fokozhatják a stabilitást.

Köszönetnyilvánítás

A kutatás az Európai Unió támogatásával valósult meg, az Autonóm Rendszerek Nemzeti Laboratórium keretében. (RRF-2.3.1-21-2022-00002)

Irodalom

- [1] Gerald Singh és tsai. *Baleset statisztika*. 2015. szept.
- [2] Guna Seetharaman, Arun Lakhotia és Erik Philip Blasch. „Unmanned vehicles come of age: The DARPA grand challenge”. *Computer* 39.12 (2006), 26–29. old.
- [3] Jessica Van Brummelen és tsai. „Autonomous vehicle perception: The technology of today and tomorrow”. *Transportation Research Part C: Emerging Technologies* 89 (2018), 384–406. old. ISSN: 0968-090X. DOI: <https://doi.org/10.1016/j.trc.2018.02.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X18302134>.
- [4] Markus Maurer és tsai. *Autonomous Driving. Technical, Legal and Social Aspects*. 2016. máj. DOI: 10.1007/978-3-662-48847-8.
- [5] Brian Paden és tsai. „A survey of motion planning and control techniques for self-driving urban vehicles”. *IEEE Transactions on intelligent vehicles* 1.1 (2016), 33–55. old.
- [6] Tomás Lozano-Pérez és Michael A Wesley. „An algorithm for planning collision-free paths among polyhedral obstacles”. *Communications of the ACM* 22.10 (1979), 560–570. old.
- [7] Huijuan Wang, Yuan Yu és Quanbo Yuan. „Application of Dijkstra algorithm in robot path-planning”. *2011 second international conference on mechanic automation and control engineering*. IEEE. 2011, 1067–1069. old.
- [8] *Python Robotics git repository, Megtekintve: 2023.11.06*. URL: <https://github.com/AtsushiSakai/PythonRobotics>.
- [9] Mohammadreza Radmanesh és tsai. „Overview of path-planning and obstacle avoidance algorithms for UAVs: A comparative study”. *Unmanned systems* 6.02 (2018), 95–118. old.
- [10] Michael Montemerlo és tsai. „Junior: The stanford entry in the urban challenge”. *Journal of field Robotics* 25.9 (2008), 569–597. old.
- [11] Steven LaValle. „Rapidly-exploring random trees: A new tool for path planning”. *Research Report 9811* (1998).

- [12] Lydia E Kavraki és tsai. „Probabilistic roadmaps for path planning in high-dimensional configuration spaces”. *IEEE transactions on Robotics and Automation* 12.4 (1996), 566–580. old.
- [13] Szilárd Aradi. „Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles”. *IEEE Transactions on Intelligent Transportation Systems* 23.2 (2022), 740–759. old. DOI: 10.1109/TITS.2020.3024655.
- [14] Richard S Sutton. „Reinforcement learning: An introduction”. *A Bradford Book* (2018).
- [15] Maximilian Jaritz és tsai. „End-to-end race driving with deep reinforcement learning”. *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, 2070–2075. old.
- [16] Árpád Fehér, Szilárd Aradi és Tamás Bécsi. „Hierarchical Evasive Path Planning Using Reinforcement Learning and Model Predictive Control”. *IEEE Access* 8 (2020), 187470–187482. old. DOI: 10.1109/ACCESS.2020.3031037.
- [17] Carlos E Garcia, David M Prett és Manfred Morari. „Model predictive control: Theory and practice—A survey”. *Automatica* 25.3 (1989), 335–348. old.
- [18] Kang Yuan és tsai. „Mixed local motion planning and tracking control framework for autonomous vehicles based on model predictive control”. *IET Intelligent Transport Systems* 13.6 (2019), 950–959. old.
- [19] Robin Verschueren és tsai. „Towards time-optimal race car driving using nonlinear MPC in real-time”. *53rd IEEE conference on decision and control*. IEEE. 2014, 2505–2510. old.
- [20] Mauro Salazar és tsai. „Real-time control algorithms for a hybrid electric race car using a two-level model predictive control scheme”. *IEEE Transactions on Vehicular Technology* 66.12 (2017), 10911–10922. old.
- [21] Paul T Boggs és Jon W Tolle. „Sequential quadratic programming”. *Acta numerica* 4 (1995), 1–51. old.
- [22] Robert B Wilson. „A simplicial algorithm for concave programming”. *Ph. D. Dissertation, Graduate School of Business Administration* (1963).
- [23] Shih-Ping Han. „Superlinearly convergent variable metric algorithms for general nonlinear programming problems”. *Mathematical Programming* 11.1 (1976), 263–282. old.
- [24] Klaus Schittkowski. „The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function: Part 1: Convergence analysis”. *Numerische Mathematik* 38 (1982), 83–114. old.

- [25] Dieter Kraft. „A software package for sequential quadratic programming”. *Forschungsbericht-Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt* (1988).
- [26] Richard S Wallace és tsai. „First Results in Robot Road-Following.” *IJCAI*. 2. köt. 1985, 1089–1095. old.
- [27] Y. Kanayama és tsai. „A stable tracking control method for an autonomous mobile robot”. *Proceedings., IEEE International Conference on Robotics and Automation*. 1990, 384–389 vol.1. DOI: 10.1109/ROBOT.1990.126006.
- [28] Péter Gáspár, Zoltán Szabó és József Bokor. „LPV design of fault-tolerant control for road vehicles”. *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE. 2010, 807–812. old.
- [29] Tamás Hegedűs, Balázs Németh és Péter Gáspár. „Design of a low-complexity graph-based motion-planning algorithm for autonomous vehicles”. *Applied Sciences* 10.21 (2020), 7716. old.
- [30] Brian Beckman és No Bucks Racing Club. „The physics of racing, part 5: Introduction to the racing line”. *online*] <http://www.esbconsult.com.au/ogden/locust/phors/phors05.htm> (1991).
- [31] Sky McKinley és Megan Levine. „Cubic spline interpolation”. *College of the Redwoods* 45.1 (1998), 1049–1060. old.
- [32] Moritz Werling és tsai. „Optimal trajectory generation for dynamic street scenarios in a frenet frame”. *2010 IEEE international conference on robotics and automation*. IEEE. 2010, 987–993. old.
- [33] A. Takahashi és tsai. „Local Path Planning And Motion Control For Agv In Positioning”. *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems '. (IROS '89) 'The Autonomous Mobile Robots and Its Applications*. 1989, 392–397. old. DOI: 10.1109/IROS.1989.637936.

Ábrák jegyzéke

2.1. Járművek döntéshozatali hierarchiájának szintjei	5
2.2. Dijkstra algoritmus [8]	6
2.3. A* algoritmus [8]	7
2.4. RRT algoritmus [8]	8
2.5. PRM algoritmus [8]	8
2.6. Megerősítéses tanulás működési ábra [14]	10
3.1. 2D grafikus felület	15
3.2. Elméleti kanyarív	17
3.3. Verseny kanyarív	17
3.4. Globális útvonal az elméleti kanyarívnek megfelelően	19
3.5. Frenet koordináta-rendszer	20
4.1. Jármű karakterisztikák véletlenszerű pályákon különböző random seedekkel	25
4.2. Optimalizált trajektória példa	26
4.3. Optimalizált trajektória görbülete	27
4.4. Jármű karakterisztika összehasonlítás	28