



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Automatizálási és Alkalmazott Informatikai Tanszék

# Valósídejű kockázatelemzés, szituációosztályozás és viselkedéstervezés autonóm járművek számára

TDK DOLGOZAT

*Készítette*

Dávid Bence

Hunyady Gergely

Láncz Gergő

*Vállalati konzulens*

Gyurkó Zoltán

Straub Kornél

*Tanszéki konzulens*

Csorvási Gábor

Kiss Domokos

2018. október 29.

# Tartalomjegyzék

<b>Kivonat</b>	<b>4</b>
<b>Abstract</b>	<b>5</b>
<b>Bevezető</b>	<b>6</b>
<b>1. Autonóm járművek</b>	<b>7</b>
1.1. Történelmi áttekintés . . . . .	7
1.2. Motiváció a gépjárművek automatizálására . . . . .	10
1.3. Autonóm járművek osztályozása . . . . .	12
1.4. Automatizálható részfeladatok . . . . .	14
1.5. Technikai kihívások . . . . .	14
1.6. Jogi akadályok . . . . .	16
1.7. Morális kérdések . . . . .	17
<b>2. A projekt bemutatása</b>	<b>19</b>
2.1. A tesztjármű architektúrája . . . . .	19
2.2. Rendszerarchitektúra . . . . .	20
2.3. Környezetérzékelés . . . . .	21
2.4. Döntéshozás . . . . .	23
2.5. Szimulációs környezet . . . . .	24
<b>3. Valósídejű kockázatelemzés</b>	<b>27</b>
3.1. Feladat megfogalmazása . . . . .	27
3.2. Saját rendszer felépítése . . . . .	28
3.2.1. Kockázat definiálása . . . . .	28
3.2.2. Konceptió . . . . .	29
3.2.3. Részfeladatok áttekintése . . . . .	33
3.3. Tanító adatok létrehozása . . . . .	33
3.3.1. Algoritmus . . . . .	33
3.3.2. Implementált számítás . . . . .	47
3.3.3. Grafikus megjelenítő . . . . .	49
3.3.4. Adatgyűjtés . . . . .	50
3.4. Hálótervezés és tanítás . . . . .	50

3.5. Fejlesztési lehetőségek . . . . .	52
<b>4. Forgalmi szituációk generálása, elemzése</b>	<b>53</b>
4.1. Forgalmi szituációk osztályozása . . . . .	53
4.1.1. Végrehajtott manőver szerinti felosztás . . . . .	53
4.1.2. Speciális helyzetek felismerése . . . . .	54
4.1.3. Forgalom sűrűségének osztályozása . . . . .	54
4.1.4. Osztályozás az érzékelt veszély mértéke szerint . . . . .	55
4.1.5. Érzékelés minőségét befolyásoló környezeti tényezők felismerése . . . . .	55
4.2. Konceptió . . . . .	56
4.2.1. Gépi tanulás . . . . .	57
4.2.2. Osztályozó algoritmusok bemutatása . . . . .	59
4.2.3. Tanító adatok gyűjtése . . . . .	62
4.2.4. A tanítási folyamat és az eredmények értékelése . . . . .	62
<b>5. Valósídejű viselkedéstervezés</b>	<b>64</b>
5.1. A feladat körvonalazása . . . . .	64
5.2. Valószínűségi hálók elmélete . . . . .	65
5.2.1. A választás okai . . . . .	65
5.2.2. Valószínűségi alapfogalmak . . . . .	67
5.2.3. Bayes-hálók felépítése . . . . .	69
5.2.4. Döntési hálók . . . . .	72
5.3. Az algoritmus megtervezése . . . . .	74
5.3.1. Az algoritmus bemenetei . . . . .	74
5.3.2. Sávváltási lehetőségek meghatározása . . . . .	75
5.3.3. Sávváltási igény meghatározása . . . . .	76
5.3.4. A valószínűségi háló felépítése . . . . .	78
5.3.5. Az algoritmus működése különböző szituációkban . . . . .	79
5.3.6. A valószínűségi háló tanítása . . . . .	80
5.4. Összegzés . . . . .	80
<b>Összefoglalás</b>	<b>81</b>
<b>Köszönetnyilvánítás</b>	<b>82</b>
<b>Ábrák jegyzéke</b>	<b>84</b>
<b>Táblázatok jegyzéke</b>	<b>85</b>
<b>Irodalomjegyzék</b>	<b>90</b>

## HALLGATÓI NYILATKOZAT

Alulírott *Dávid Bence*, *Hunyady Gergely* és *Láncz Gergő* hallgatók kijelentjük, hogy ezt a dolgozatot meg nem engedett segítség nélkül, saját magunk készítettük, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtuk fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettünk, egyértelműen, a forrás megadásával megjelöltünk.

Hozzájárulunk, hogy a jelen munkánk alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentjük, hogy a benyújtott munka és annak elektronikus verziója megegyezik.

Budapest, 2018. október 29.

---

*Dávid Bence*  
hallgató

---

*Hunyady Gergely*  
hallgató

---

*Láncz Gergő*  
hallgató

# Kivonat

Napjainkban nem találkozhatunk olyan jelentős szereplővel az autóiparban, amelynek ne lennének autonóm járművekkel kapcsolatos fejlesztései. Dolgozatunkban a Knorr-Bremse Fékrendszerek Kft-vel együttműködésben végzett kutatásunk eredményeit fogjuk ismertetni. A vállalat profiljából adódóan a témánk elsősorban haszongépjárművek automatizálását tűzte ki céljául, de a tárgyalt algoritmusok nem feltétlenül teherautó-specifikusak.

A fejlesztési munkánkhoz rendelkezésre álltak környezeti információk (objektumok, sávok, útparaméterek egy bizonyos horizonton, forgalmi szituáció, stb.) absztrakt formában, amely információk figyelembevételével kellett autópályás környezetben magasabb szintű problémák kezelésére megoldásokat keresnünk. A dolgozat három eltérő, de egymáshoz szorosan kapcsolódó témával foglalkozik: kockázatelemzés, szituációosztályozás és viselkedéstervezés. Mindhárom feladat valós időben kezelendő, ezért az algoritmusok futásideje, a megoldások optimalizálása kritikus pontja a specifikációknak.

Egy-egy manőver elvégzése előtt tudni szeretnénk, hogy az mekkora kockázattal jár. Valós időben nincs lehetőségünk nagyszámú trajektóriák szimulációjára, ezért ennél hatékonyabb megoldásra van szükségünk. Kutatásaink során neurális hálók alkalmazását vizsgáltuk a problémára, így csökkentve az algoritmus számításidejét.

A környezetérzékelésből származó absztrakt adatok nem tartalmaznak információt a forgalmi szituációról, a forgalom sűrűségéről és hektikusságáról. A magasabb szintű döntések meghozatalához ezek lényeges információk, mivel máshogy vezetünk például dugóban, gyorsan haladó, de sűrű forgalomban, vagy egy üres autópályán. Ezzel kapcsolatban forgalmi szituációk generálásával, osztályozásával és egyéb számszerűsíthető elemzésével foglalkoztunk.

Az autonóm járművek viselkedéstervezése komplex és többértékű feladat. Mind stratégiai mind taktikai szempontból az adott forgalmi szituációk és járműállapotok figyelembevételével kell megtalálni a legbiztonságosabb és optimális manőversorozatot, a körülmények folyamatos változására adaptívan reagálva. Valószínűségi következtetések és döntések alkalmazásával a korábban említett két témával szoros összefüggésben a jármű viselkedését meghatározó algoritmussal foglalkoztunk.

# Abstract

Considering the major participants in the automotive industry, it is hard to find one with no development connected to autonomous driving. This study provides insight to our work we did in collaboration with Knorr-Bremse Fékrendszerek Kft. Given the profile of the company, our topic focused on heavy duty commercial vehicles, although the discussed algorithms are not truck-specific.

As an input a predefined abstract structure of data is given, describing the environment and the situation that the ego vehicle is in. Using these information we tried to find feasible solutions to some problems which could improve high level decision making in highway scenarios. Our documentation joins three different but closely related projects, namely maneuver risk assessment, classification of traffic situations and behaviour planning.

We use a discretized maneuver space to handle the infinite amount of possibilities. For these chosen points of our space it is a necessity to assign risk values which can exclude maneuvers and help for further decision algorithms at the same time. Analyzing heaps of trajectories is an appealing idea, as long as we don't meet the real-time requirements. Our study dealt with the use of neural networks as a risk estimator.

The abstract data provided by the environment perception module of the system do not give information like the actual traffic scenario, its density or its hecticity. However, all of them are relevant in case of high level decision making. For instance, drivers behave differently in a stop-and-go traffic jam, in dense traffic (which goes ahead with almost full speed) or on an empty highway. We investigated the generation, classification and other quantifiable analysis of traffic situations.

Behavior planning for autonomous vehicles is complicated and versatile task. From both strategic and tactical aspects, the most secure and optimal series of maneuver must be found taking into account the given traffic scenario and vehicle state, while adapting continuously to the constant change of circumstances. During our research, we worked on an algorithm determining the behavior of the vehicle in connection with the above-mentioned other two subproblems.

# Bevezető

A Knorr-Bremse Fékrendszerek Kft. a világon elsőként mutatta be két éve az Autonomous Yard Maneuvering elnevezésű telepi önvezető rendszerét, idén pedig prezentálta legújabb fejlesztését, a Highway Pilot nevű projektjét. A cég előfejlesztési osztályán lehetőségünk nyílt ezen úttörő téma három részfeladatához csatlakoznunk és kutatásaink eredményeit ebben a dolgozatban összefoglalnunk.

A dokumentumot öt fejezetre tagoltuk. Az elsőben az autonóm járművekről fogunk általánosságban beszélni, egy kis történelmi bevezetést követően röviden taglaljuk, hogy miért is érdemes haszongépjárművek automatizálásával foglalkozni. Ebben a fejezetben szót ejtünk a közismert SAE szintekről, illetve a környezet komplexitása és a haladási sebesség szerinti csoportosításban helyezzük el a különböző automatizálható feladatokat. Végezetül kitérünk még az önvezető autókát övező jogi és morális kérdésekre.

A második fejezetben az előfejlesztési osztályon található tesztjárműről lesz szó, röviden ismertetjük az autó architekturális felépítését, a rajta lévő szenzorokat, a feldolgozott objektuminformációkat. Ezen kívül szót ejtünk még az autó vonatkoztatási rendszerében definiált diszkrét manővertérről, illetve a szimulációs eszközökről is.

A harmadik fejezetben a döntéshozás egyik legelső és leginkább kritikus részéről, a kockázati analízisről olvashatunk. Első lépésként megvizsgáljuk, hogy a második fejezetben megismert projekthez hogyan kapcsolódik a kockázatszámítás témaköre. Ezt követően meghatározzuk a kockázat általunk használt definícióját, amin elindulva bemutatunk egy általunk megalkotott valósidejű kockázatbecslésre alkalmas rendszertervet és annak inspirációját. A fejezetben továbbá bemutatjuk ezen rendszer megalkotásához vezető munkánk aktuális állapotát és eredményeit, valamint a továbblépés lehetőségeit.

A negyedik fejezetben megvizsgáljuk a forgalmi szituációk osztályozási kérdéseit, illetve motivációját. Ismertetjük a problémakörben gyakran alkalmazott algoritmusokat, majd részletezzük a kiválasztott megközelítést.

Az ötödik, egyben utolsó fejezetben a viselkedéstervező modulról lesz szó. A feladat megfogalmazását követően a valószínűségi hálók elméletét érintjük, egy orvosi példán keresztül bemutatjuk a Bayes-hálók működését, illetve ismertetjük az egyszerű döntések mechanizmusát felhasználásuk segítségével. Ezt követően áttérünk a hálók viselkedéstervezésre történő alkalmazására, definiáljuk az algoritmus bemeneteit, ismertetjük a felépítését és szót ejtünk a tanításáról. A fejezet a rendszerrel kapcsolatos jövőbeli teendőkkal zárul.

# 1. fejezet

## Autonóm járművek

Napjainkban az önvezető járművek a mérnöki tudományok legfrekvenciáltabb területeinek egyikét jelentik. Számos őket övező hírrel, jóváindulással találkozhatunk nap mint nap. Vannak jóslatok, amelyek szerint néhány év, vannak, amelyek szerint több évtized múlva fogják elárasztani az utakat, de egy biztos, hogy egy gyorsan fejlődő területről van szó.

E fejezet az autonóm járművek világába nyújt átfogó betekintést, hogy az olvasó szélesebb rálátás birtokában szemlélhesse a dolgot további részeit. Kezdve a XX. század első felétől, nagy lépésekkel eljutunk korunk aktualitásaihoz és pár szóban megvizsgáljuk az önvezető járművek fejlődésének technikai, jogi valamint morális kihívásait is.

### 1.1. Történelmi áttekintés

Az emberi tényező kiküszöbölése és a járművek vezetőinek helyettesítése nem tekinthető forradalmian új gondolatnak, már az első tömeggyártásban készülő autó – a Ford T-modell – idejéből is olvashatunk próbálkozásokról, mint például az 1926-os rádióvezérlésű *Phantom Auto*-ról [1] Milwaukee utcáin. Gyakorlatilag a gépjárművek szélesebb körben való elterjedésével egy időben merült fel az igény, hogy a vezető által végzett feladatokat gépekkel váltsuk ki.

Egy amerikai ipari formatervező, Norman Bel Geddes, 1940-ben a *Magic Motorways* [2] című könyvében olyan járműveket vizionált a 60-as évekre, amelyek az úthálózatba integrált eszközökkel kommunikálva kijavítják a vezetők által vétett hibákat. Az 50/60/70-es években valóban több ígéretes projekt [3], [4] is futott, amelyek az útburkolatba ágyazott jeladóknál láttak komoly potenciált. A lehetőséget azóta sem vetette el a mérnöktársadalom, többek között a platooning<sup>1</sup> megoldások [5] kifejlesztéséhez ma is számba veszik az útba épített mágnesek lehetőségét, amelyek egy Automated Highway System (AHS)<sup>2</sup> részét képezhetik.

---

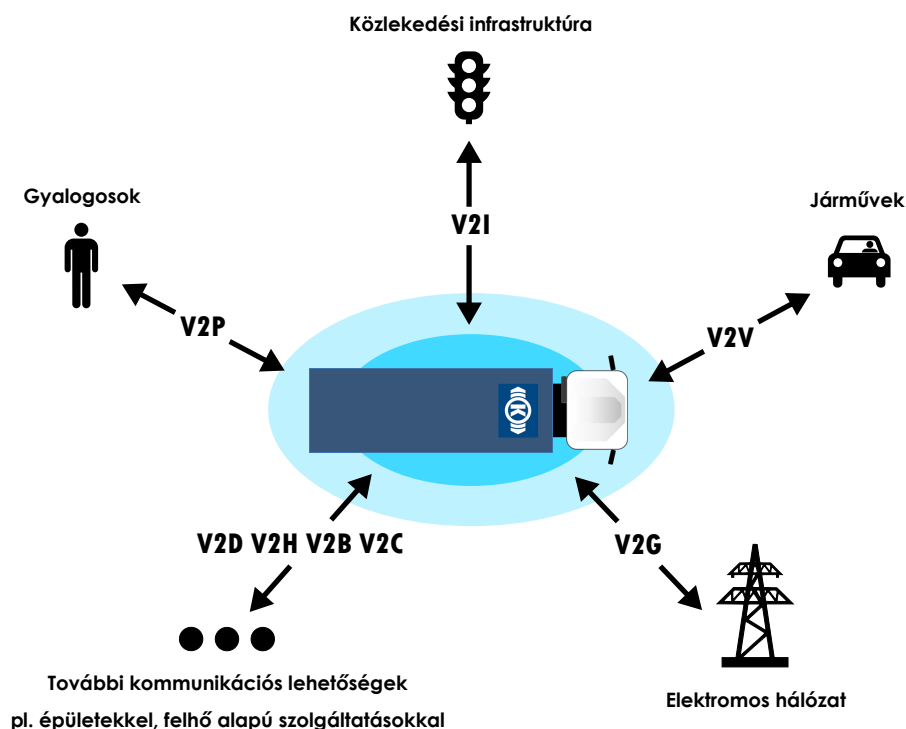
<sup>1</sup> Autópálya környezetre kitalált közlekedési megoldás, amely a járművek csoportokba rendezésével hivatott megnövelni az úthálózat kapacitását, valamint csökkenteni a résztvevők fogyasztását.

<sup>2</sup> Automated Highway System: Intelligens úthálózat rendszer önvezető járművek támogatására.



Ugyanakkor ezen megoldások több komoly problémát is felvetnek. Egyrészt hatalmas beruházási költséget igényel az infrastruktúra átalakítása, így kérdéses, hogy ennek finanszírozását hogyan oldanánk meg, ugyanis a cégek és az állami szereplők egymástól várják a lépéseket. Továbbá amennyiben egyes régiókban meg is valósulnak a szükséges fejlesztések, globálisan nem teszik lehetővé az autonóm járművek használatát. Másik fontos probléma, hogy egy ilyen rendszer használatához magas szintű bizalomra van szükség a gépjárművek gyártóitól, hiszen olyan adatok alapján hoznák a döntéseiket, amelyek részben tőlük függetlenül keletkeznek és ezáltal nehéz azok megbízhatóságát ellenőrizni.

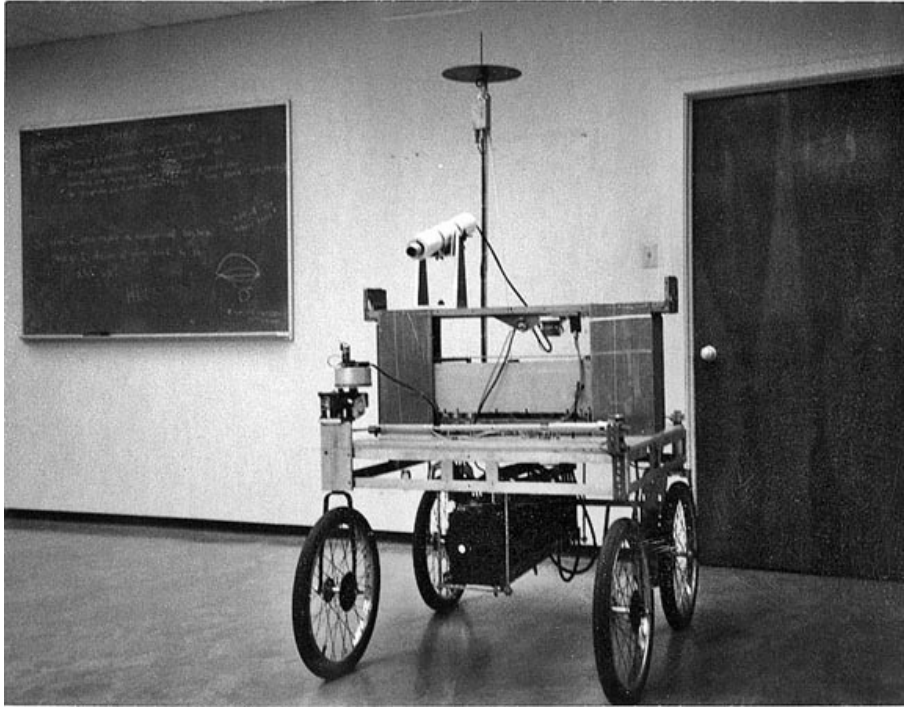
Hasonló kérdéseket vet fel az 1.1. ábrán látható V2X kommunikációs rendszerek<sup>3</sup> használata is [6], melyekről egyre több szó esik napjainkban a hálózati lefedettség és sebesség növekedésének köszönhetően. Ezen ötleteket természetesen közlekedésről sem tartjuk elvetendőnek, az általuk nyújtott lehetőségek biztonságosabbá, gyorsabbá és kellemesebbé tehetik a közlekedésünk jövőjét, ugyanakkor kizárólagos alkalmazásuk nem nyújt általános érvényű megoldást.



1.1. ábra. V2X kommunikációs protokollok

Az 1980-as évektől kezdve már találkozhatunk olyan projektekkel, amelyek igyekeznek úgy autonóm funkciókat megvalósítani, hogy a jármű a lehető legkevesebb külső rendszerre támaszkodjon, a döntésekhez szükséges adatokat mind be tudja gyűjteni saját érzékelőkkel. Az egyik első ilyen nagy hatású kutatás a The Stanford Cart [7] volt, a Stanford Egyetem mesterséges intelligenciát kutató laboratóriumának fejlesztése (1.2. ábra). A jármű gépi látásra támaszkodva méterenkénti megállással térképezte fel a környezetét, majd útvonalat tervezett az akadályok elkerülésére és végrehajtotta azt.

<sup>3</sup> V2X: Vehicle to everything kommunikációs technológiák gyűjtő elnevezése.



1.2. ábra. *The Stanford Cart 1980* (Forrás: Hans Moravec [8])

A korszak másik fontos fejleménye volt, hogy ekkor alapította az Egyesült Államok Védelmi Minisztériumának kutatásokért felelős részlege (DARPA) az AVL projektet, amely később a széles körben ismert DARPA Grand Challenge-é nőtte ki magát, melynek 2004-es és 2005-ös megrendezésekor még sivatagos területen zajlott a verseny, 2007-ben viszont már városi környezetben. A futamoknak köszönhetően számos technikai jelentés áll rendelkezésünkre a csapatok tapasztalatai alapján.

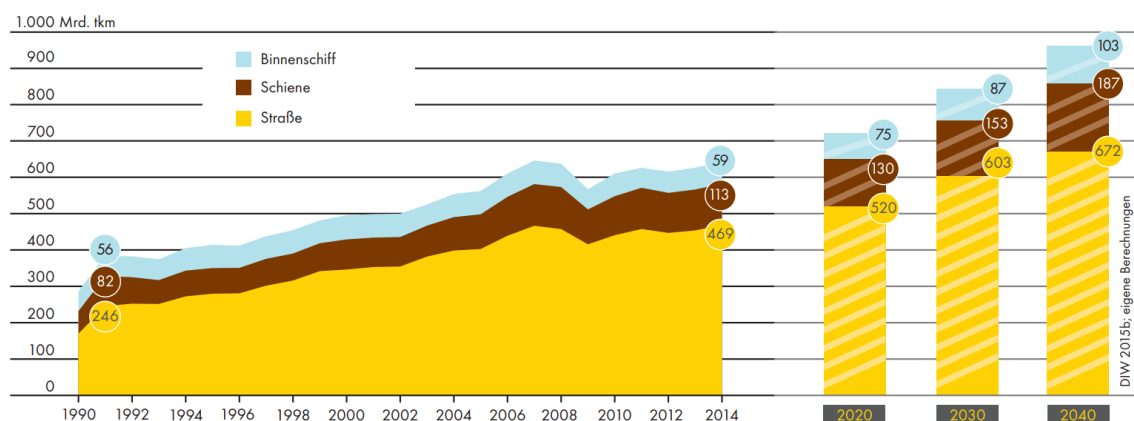
Az önvezető járművek fejlődésének történetében fontos továbbá következő lépésként megemlíteni a Carnegie Mellon Egyetem 1989-es kutatását [9], melynek során itt alkalmaztak elsőként neurális hálót a kívánt haladási irány meghatározására. Megoldásuk a nyers képi információból egyetlen rejtett réteg felhasználásával egyenesen a kívánt haladási irányt határozta meg (gyakorlatilag megkereste, merre található az útburkolat).

Innentől kezdve az autonóm autók fejlesztése egyre szélesebb köröket mozgatott meg, a hagyományos gépjárműgyártókon túl számos más területről induló vállalat is beszállt a versenybe, rájöttek ugyanis, hogy a rendelkezésükre álló hatalmas adatmennyiség kiváló alapot nyújt a mesterséges intelligencia tanításához. Napjaink úttörői, mint például a Tesla, a Google, az Uber, az Audi vagy a Volvo már olyan megoldásokkal rendelkeznek, amelyek valós forgalomban, meghatározott körülmények esetén a kormányzás és a gyorsítás-lassítás feladatát is átveszik a vezetőtől. Közös jellemzőjük ezeknek a rendszereknek, hogy olyan technológiákat alkalmaznak, amelyek még csak prémium áron elérhetőek, ezért is lenne nagy eredmény egy olyan megoldás megalkotása, amely relatív olcsó érzékelőket és technológiákat alkalmaz, ezáltal egy olyan rendszer jöhetne létre, amelynek költségei drasztikusan elmaradnak a versenytársakétól.

Elsősorban a személygépjárművekkel kapcsolatban hallunk a közúti közlekedés automatizálásáról, hiszen a mainstream médiát főként ez foglalkoztatja, ugyanakkor a haszongépjárművek gyártói is hatalmas energiát fektetnek a fejlesztésbe. Jelenleg két fő irányvonala van ezeknek a projekteknek, egyrészt a telephelyen történő manőverezést, másrészt az autópályán való közlekedést próbálják az ezzel foglalkozó cégek megoldani. Attól kitüntetettek ezek az esetek, mivel ezekben számos könnyebbségünk van a teljes automatizáláshoz képest (pl. lassabb haladási sebesség, speciális infrastruktúra igénye könnyebben megvalósítható, jól strukturált környezet, stb.), ugyanakkor hosszan tartó és sokszor monoton vezetési feladatot tudnánk ezzel kiváltani. Dolgozatunkban az autópályán történő automatizált vezetés problémáinak egy szűk részhalmazát fogjuk tárgyalni.

## 1.2. Motiváció a gépjárművek automatizálására

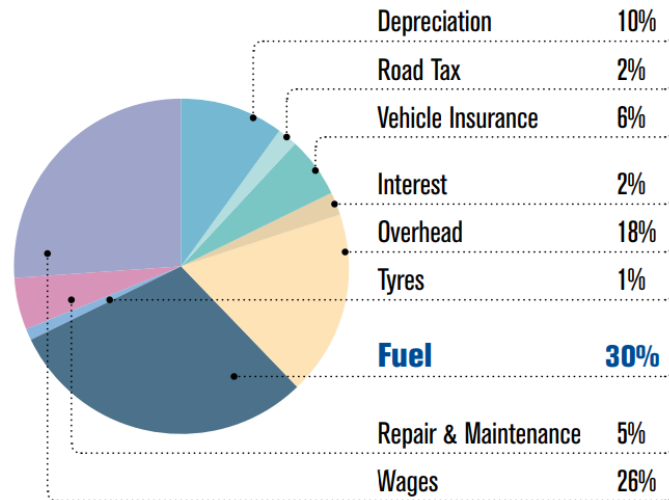
Az eddigiekben röviden láthattuk azt, hogy már az autózás hajnalán foglalkoztatta a mérnököket az automatizált vezetés, azonban ennek okait még nem tárgyaltuk. Nézzük meg, hogy milyen tényezők motiválják ezeket a fejlesztéseket. Dolgozatunkban elsősorban haszongépjárművek számára fejlesztett algoritmusokról lesz szó, ezért maradjunk ennél a kategóriánál. A Schell egy 2016-os tanulmányában [10] az 1.3 ábrán látható mértékű növekedést jósol a németországi teherfuvarozásban. Ez a tendencia nyugodtan általánosítható a többi fejlett nyugat-európai országra is.



1.3. ábra. Az áruszállítás várható növekedése Németországban [10]

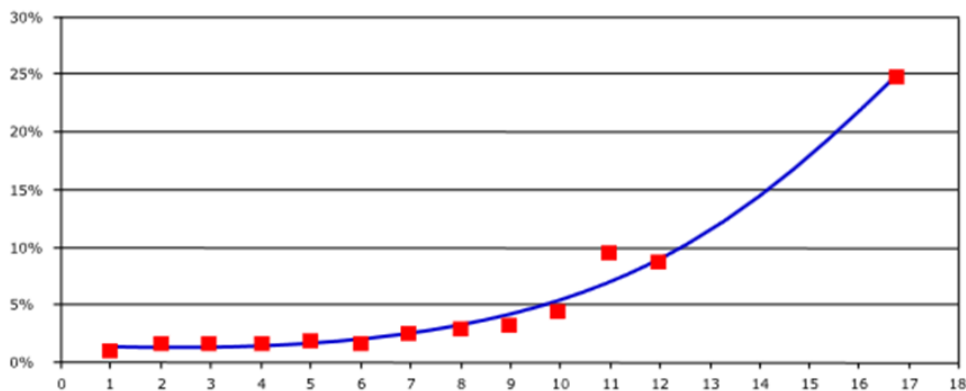
Az ábrán látott forgalomművekedéssel az úthálózat áteresztőképessége várhatóan nem tud majd lépést tartani. Európa közúthálózata napjainkban is meglehetősen jól kiépített, az utak számának jelentős növekedése nem várható, bővítésük is körülményes. A forgalom élénkítése további közlekedési dugókat, az utazási idők növekedését fogja eredményezni, várhatóan a baleseti statisztikákra sem lesz kedvező hatása.

A dugóban töltött idő csökkenti a jármű kihasználtságát és növeli a sofőr jelentett költséget, amely napjainkban is jelentős részét teszi ki a fuvardíjnak. Egy 2010-es tanulmány [11] szerint egy negyven tonnás szerelvény esetében az összes költség több, mint egynegyed részét a sofőr munkabére jelenti.



1.4. ábra. A közúti árufuvarozás költségei [11]

Az ember által vezetett haszongépjárművek esetén még egy jelentős korlátozásunk van, ez pedig a sofőrök által betartandó szabályok miatti pihenőidők, amikor az autó órákig áll. Ezen szabályozások indokoltak, a fáradtság jelentősen növeli a balesetek kockázatát [12].



1.5. ábra. A baleset valószínűsége ( $y$  tengely) a vezetési idő túlteljesítésének ( $x$  tengely) függvényében [13]

A fejlett országokban megfigyelhető forgalomművekedési tendenciák és a költségek leszorítása iránti mindenkori igény megfelelő motivációt biztosít a haszongépjárművek bizonyos feladatainak, távoli célként pedig a teljes körű automatizálására.

Személygépjárművek esetén cél lehet a kényelem növelése, a balesetek számának csökkentése, hosszú távon pedig a teljesen önvezető bérautó-hálózatok kiépítése, amellyel a járművek kihasználtsága a saját tulajdonban lévőkhöz képest nagyságrendekkel növelhető. Ez számuk jelentős csökkenésével, a szabad parkolóhelyek számának növekedésével járna. Vannak olyan tanulmányok [14], amelyek szerint a jövőben már csak a felsőbb társadalmi rétegek kiváltsága lesz saját tulajdonú autóval járni, lényegesen olcsóbb lesz az önvezető autók bérlése. A magas fokú automatizáltságtól a balesetek számának jelentős csökkenését is várják.

### 1.3. Autonóm járművek osztályozása

Az autonóm járművek osztályozását a SAE J3016 [15] szabvány által definiált szintek szerint szokás elvégezni. Ez a dokumentum az automatizáltság fokát öt csoportra osztja, a nulladik szint pedig annak teljes hiányát jelöli. A teljesség igénye miatt fontos megemlíteni, hogy léteznek más osztályozások is, mint például az NHTSA által 2013-ban közzétett [16], azonban ezek nagyon hasonlóak az itt bemutatotthoz, jellemzően könnyedén megfeleltethetőek egymásnak az összetartozó elemek. A szinteket az 1.1. táblázatban foglaltuk össze, melynek megértéséhez szükséges néhány fogalom és definíció tisztázása:

**Laterális irányítás:** A jármű oldalirányú dinamikus viselkedésének irányítása, melybe a kormányval való beavatkozáson túl az aszimmetrikus fékezés/gyorsítás is beletartozik, hiszen ezzel is el tudjuk érni a szerelvény laterális irányváltoztatását.

**Longitudinális irányítás:** A jármű hosszirányú dinamikájának irányítása, azaz a fékezés és gyorsítás feladatát foglalja össze.

**OEDR:** Object and Event Detection and Response – A vezetési környezet monitorozásának folyamatos feladata, a peer<sup>4</sup> járművek és más objektumok valamint események felismerése és értelmezése, továbbá megfelelő reakció ezekre az információkra.

**DDT:** Dynamic Driving Task – A dinamikus vezetés minden részfeladatát összefoglaló fogalom, ez praktikusán a laterális és longitudinális irányítást, valamint az OEDR-t foglalja magába.

**DDT fallback:** A DDT-t ellátó alrendszer hibája esetén vagy a vezetőnek, vagy a rendszernek kell gondoskodnia a DDT további elvégzéséről, lényegében a DDT-re vonatkozó tartalék stratégiát jelenti.

**ODD:** Operational Design Domain – A rendszer tervezett működési tartománya, azaz, hogy milyen környezetben képes a jármű önvezető módban működni.

Érdemes megfigyelni, hogy a táblázaton belül található egy lépcsőzetes elválasztóvonal, amely szemléletesen mutatja meg a gép és az ember közti határt, az egyre magasabb szintű autonóm járművek egyre több részfeladatot vállalnak át az embertől.

Megjegyzendő továbbá, hogy a második szintről a harmadikra ugorva egy nagyon éles váltás következik be. Míg a level 0 - level 2 fokozatig a vezető végzi a DDT legalább egy részét, addig level 3 szinttől fölfelé már a DDT teljes egésze a rendszer feladatát képezi. Pont ennek a markáns lépcsőnek a megugrásán dolgozik a legtöbb gyártó napjainkban.

A dolgozatunk témáját adó autópályán történő automatizált vezetés a sofőr meglététől függően a harmadik vagy negyedik szintbe sorolható.

---

<sup>4</sup> A peerek a környezetünkben közlekedő forgalmi résztvevők, amelyek kellően közel vannak hozzánk, emiatt relevánsak lehetnek a döntéseink meghozatalánál.

**1.1. táblázat.** Autonóm járművek fokozatai a SAE szabványa alapján.

(a táblázatban a vezető által végzett feladokat V, míg a rendszer által R jelöli)

SAE szint	Név	Definíció	Laterális, longitudinális irányítás	OEDR	DDT fallback	ODD
0	Vezetés automatizálás nélküli	A jármű dinamikus irányításának minden részfeladatát mindig a vezető végzi, de a biztonságos vezetést elősegíthetik különböző érzékelők adatai alapján generált figyelmeztető jelzések.	V	V	V	n/a
1	Vezetési asszisztens	Egyes vezetési módokban egy asszisztens rendszer végzi a kormányzás és gyorsítás/lassítás közül az egyiket, feltételezve, hogy minden más vezetési feladatot ellát az ember.	V + R	V	V	Korlátozott
2	Részleges vezetés automatizálás	Egyes vezetési módokban egy vagy több asszisztens rendszer végzi a kormányzás és gyorsítás/lassítás feladatait, feltételezve, hogy minden más vezetési feladatot ellát az ember.	R	V	V	Korlátozott
3	Feltételes vezetés automatizálás	Egyes vezetési módokban a jármű dinamikus irányításának minden részfeladatát a rendszer végzi, feltételezve, hogy szükség esetén meghatározott időn belül rendelkezésre áll és megfelelően beavatkozik az ember.	R	R	V	Korlátozott
4	Magas szintű vezetés automatizálás	Egyes vezetési módokban a jármű dinamikus irányításának minden részfeladatát a rendszer végzi, még akkor is, ha szükség esetén sem áll rendelkezésre megfelelő emberi beavatkozás.	R	R	R	Korlátozott
5	Teljes vezetés automatizálás	Minden olyan helyzet és környezet esetén képes a jármű dinamikus irányításának minden részfeladatát a rendszer elvégezni, amit egy ember képes lenne megoldani.	R	R	R	Korlátlan

## 1.4. Automatizálható részfeladatok

Az automatizáltság szintjeinek szabvány szerinti ismertetése után vizsgáljuk meg azokat a részfeladatokat, ahol autonóm járművekre lehet szükség. Nézzük meg, hogy a jármű sebessége és a környezet komplexitása alapján milyen feladatok képzelhetőek el.

1. *Strukturált környezet, alacsony sebesség:* Strukturált környezetre kiváló példa az autópálya, ahol az alacsony sebesség forgalmi dugót jelent. Itt level 2 szintű termékek már forgalomban vannak, amelyek megkönnyítik a sofőrnek az araszolást, de folyamatos felügyeletet igényelnek. Fejlesztés alatt állnak level 3 és level 4 besorolásba tartozó rendszerek, ezek már teljes mértékben átveszik a sofőr feladatát ebben a jól definiált helyzetben.
2. *Strukturált környezet, magas sebesség:* Strukturált környezetben való nagy sebességű haladásra kiváló példa a dolgozatunk témáját adó Highway Pilot rendszer. A nagy sebesség miatt a beavatkozó szervek irányításának stabilitása erősen biztonságkritikus tényező, ez jelentősen megnehezíti a feladatot. Elméleti megvalósítása józan ésszel átlátható léptékű probléma, azonban a jármű és környezetének folyamatos biztonságát garantálni már korán sem triviális. Kereskedelmi forgalomban kaphatók már olyan személyautók és vontatók, amelyek level 2 szinten automatizáltak ebben a környezetben. A fejlesztések level 3 és level 4 szintnél tartanak.
3. *Komplex környezet, alacsony sebesség:* Komplex környezetnek tekinthető egy telephely, ahol alacsony sebességgel közlekednek a járművek. Ilyen rendszerek segíthetnek a nagy holtterű járműveknek egy utcasarkon való befordulásnál, vagy akár egy zárt udvaron való teljesen önálló közlekedésben is. A fejlesztések itt is level 4 szintnél tartanak, a piacon level 2 szintű termékek már elérhetőek.
4. *Komplex környezet, magas sebesség:* A komplex környezetben nagy sebességgel történő közlekedés gyakorlatilag a legkomplikáltabb városi forgalmat jelenti. Ezen esetre a kutatások már zajlanak, de a feladat összetettsége miatt konkrét termékek fejlesztése még várat magára. A nagy sebesség lényegesen növeli a feladattal járó biztonságkritikus elvárásokat, illetve a strukturálatlan környezet, a városi forgalom összetettsége elméleti síkon is nehezíti a problémát.

## 1.5. Technikai kihívások

Jelenleg a level 2-es autonóm járművek dominálják az önvezető autók piacát, még a Tesla is ide sorolható, bár a gyártó ezt nem szereti hangoztatni (talán ennek is köszönhető, hogy egyesek túlságosan megbíznak a rendszerben [17]). A level 3-as járművek sorozatgyártásban való megjelenésének a küszöbén jár a technika, sőt az Audi legújabb A8-as szériamodellje a gyártó állítása szerint már ebbe a kategóriába tartozik<sup>5</sup>. A jog és a közlekedés szabályozói

<sup>5</sup> A 2017-ben bemutatott Audi A8 [18] már képes a harmadik szintbe sorolható önvezetésre autópályán, 60 km/óra sebesség alatt, azonban ezen funkció engedélyezése jelenleg jogi akadályokkal küzd.

azonban nem állnak erre még készen, emiatt nem igazán lehet kihasználni egy ilyen rendszer előnyeit tulajdonosként, többek közt erről is olvashatunk egy rövid áttekintést az 1.6. fejezetrészben.

Egy másik komoly kihívást jelent az önvezető autók költségének alacsonyan tartása, ugyanis egy-egy nélkülözhetetlen rendszerkomponens akár önmagában megduplázhathatja a teljes árat. A most létező autonóm járművek szinte kivétel nélkül mindegyike fel van szerelve LIDAR<sup>6</sup>-okkal is, a RADAR-ok és kamerák ugyanis önmagukban nem biztosítják a szükséges távolság információt minden helyzetben. Egy ilyen szenzor ára azonban jelenleg 75 000 \$ körül mozog, amely első sorban a mozgó alkatrészek igényelte precizitásnak köszönhető. Ezt megoldandó egy új technológia - a solid state LIDAR [19] - van feltűnően, amely mechanikus kialakítás helyett több fénysugár közötti interferencián alapuló elektromos nyaláb vezérléssel tapogatja le a környezetet.

Szintén érdekes és előremutató fejlesztéssel találkozhatunk a térképek terén is, melynek célja, hogy ugrásszerűen növekedjen a pontosság és részletesség. A GPS technológia ma már pár centiméteres pontosságú pozicionálást képes nyújtani, így megfelelő felbontású térképek segítségével például a képi információkat kiegészítő adatokat kaphatnánk a sávkövetéshez. Ez a lehetőség egy újabb redundanciát vihet a rendszerbe, ami annak megbízhatóságát tudná tovább növelni.

Egy másik probléma, aminek áthidalásában a már korábban is említett V2X kommunikációs megoldások jelenthetnek egy előrelépést, hogy a szenzorainkkal egyrészt csak a szűk<sup>7</sup> környezetünkről kapunk képet, másrészt a peer objektumok jövőbeli viselkedéséről csak feltételezéseink<sup>8</sup> lehetnek. Az 5G<sup>9</sup> hálózat fejlesztésével egy olyan infrastruktúrához juthatunk amelynek a sáv szélessége és lefedettsége lehetővé tenné a valós idejű kommunikációt a forgalom résztvevői között, így első kézből szerezhetnénk tudást a minket körülvevő járművek szándékairól, továbbá előre értesülhetünk egy az útvonalon jóval előttünk bekövetkezett balesetről vagy elterelésről és ennek az információnak a birtokában alternatív opciókat választhatunk.

Folyamatosan fejlődnek a gépi tanuláson alapuló algoritmusok is, melyekkel kapcsolatban kiemelten fontos feladat a megfelelő tanulóminták<sup>10</sup> előállítása, mivel minél több különböző eseten lett egy jármű rendszere betanítva, annál valószínűbb, hogy egy véletlenszerűnek tekinthető forgalmi helyzetben megfelelő döntést fog tudni hozni az automatika. Az iparág nagyobb szereplőinél külön munkacsoportok foglalkoznak azzal, hogy olyan szituációkat találjanak ki, amelyekre korábban még senki sem gondolt. Ez egy olyan feladat, amivel nekünk is meg kell birkóznunk a munkánk során.

---

<sup>6</sup> Lézer alapú RADAR technológia, nagy pontosságú és felbontású mélységképek készítésére.

<sup>7</sup> A rendelkezésre álló kamerák radarok és lidarok segítségével nagyságrendileg száz méterre lát a jármű.

<sup>8</sup> Komoly fejlesztések irányulnak a peer járművek viselkedésének jóslására, habár a tanuló algoritmusok egészen jól teljesítenek, a közvetlenül szerzett információkat nem tudják helyettesíteni.

<sup>9</sup> Következő generációs vezeték nélküli kommunikációs szabvány, az eddiginél nagyobb sáv szélességgel és kisebb válaszidővel.

<sup>10</sup> A cél a hatalmas és ismeretlen állapotér feltérképezése, valamint a mintákon keresztüli reprezentálása.



## 1.6. Jogi akadályok

Fontos felismernünk, hogy az autonóm járművek fejlődésének és elterjedésének a technika kihívásain túl más tényezők is határokat szabnak, komoly problémát jelent az egyes országokban adott jogszabályi környezet is.

Egyrészt sajnálatos módon a jogalkotók jellemzően nem rendelkeznek megfelelő kompetenciákkal a szükséges témakörökben, így általában kétféle helyzet tud kialakulni: vagy nem megfelelő szabályozás születik, vagy nem születik semmiféle szabályozás. Másrészt viszont nem feledkezhetünk meg arról sem, hogy az autonóm közlekedési formák elterjedése új típusú veszélyeket hordoz magában, így a felmerülő félelem részben megalapozott, és a megfelelő jogi keretrendszer megalkotása közelről sem triviális feladat. A legnehezebb talán azt meghatározni, hogy milyen biztonsági kritériumoknak és teszteknek kell megfelelnie egy önvezető járműnek ahhoz, hogy kiengedjük az utakra. Gondoljunk csak bele az önvezető és a hagyományos járművek vegyítése milyen kiszámíthatatlan forgalmi helyzetet eredményez, és hogy az esetlegesen bekövetkező balesetekért valakinek vállalnia kell a felelősséget.

Ezzel meg is érkeztünk az önvezető járművekkel kapcsolatos egyik közkedvelt vitatémájához, ami azt taglalja, hogy a vezető vagy a jármű tekintendő hibásnak, ha az autónk balesetet okoz. Az 1968-ban hetven ország által aláírt közúti forgalomról szóló bécsi megállapodás [20] kimondja, hogy a vezető mindig teljes felelősséggel tartozik a járműve viselkedéséért. Ez az az elv amely idejétmúlttá válik napjainkban és a szituáció azt sejteti, hogy az alapokat is szükséges újraírni. Visszatekintve az 1.1. táblázatra láthatjuk, hogy level 0-tól level 2-ig minden felelősség a vezetőre hárul, hiszen ő monitorozza a környezetet, bármikor át tudja venni a DDT teljes egészét. Level 5 járművek esetében szintén viszonylag egyszerű választ adhatunk a kérdésre, amennyiben az embernek nincs is lehetősége belenyúlni a DDT egyetlen részfeladatába sem, akkor a felelősség egyértelműen a járműre és rajta keresztül annak gyártójára hárul. Sokkal izgalmasabb azt eldönteni, kinek róható fel az incidens egy level 3 vagy level 4 kategóriás jármű esetén, ahol a DDT-t olykor a vezető, máskor a rendszer látja el. Előfordulhat például, hogy a gyártók kötelesek lesznek egy a repülőgépeken alkalmazott fekete dobozhoz hasonló megoldást beépíteni járműveikbe. Érdekes kérdés továbbá, amennyiben bebizonyosodik, hogy a rendszer hibázott, megtalálható és bizonyíthatóan javítható lesz-e a hiba oka. Sajnos a konkrét hibát megtalálni valószínűleg gyakran esélytelen lesz a mesterséges intelligencia útvesztőjében.

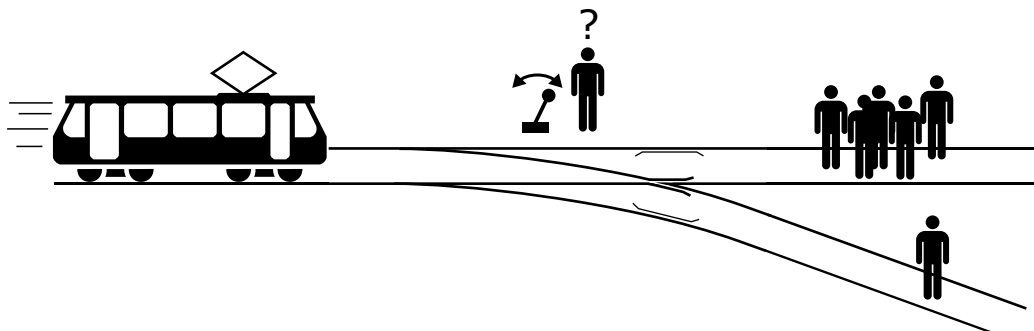
A szabályozásban lévő lyukak kritikus mivoltát jelzi például az Uber egyik tesztjárműve által a közelmúltban elszenvedett halálos kimenetelű baleset [21], ahol egyrészt a járművet felügyelő ember egyértelműen hibázott, ugyanakkor a felvételek alapján nagy valószínűséggel bárki más is tehetetlen lett volna. Az eset kivizsgálása azóta is folyik, hivatalos döntés még nem született az ügyben. A hiányosságok betömése tehát sürgető lenne, hiszen a technika már meghaladta az aktuális jogi környezetet, és ez a helyzet veszélyesnek bizonyul. Egyelőre mindenki a saját belátása szerint cselekszik a témában, egységes irányvonal nincs még kilátásban.

Érdekes terület még a jogi kérdéseken belül a biztosítások piacának adaptálódása. Már ma találkozhatunk olyan biztosítótársaságokkal, amelyek kedvezőbb termékeket kínálnak például a Tesla tulajdonosoknak. A balesetek gyakoriságának csökkenésével akár meg is szűnhet a kötelező jellegű biztosításkötés a tulajdonosok részéről. A felelősség áthelyeződése miatt valószínű forgatókönyv, hogy a jövőben a gyártók lesznek kötelesek biztosítási díjat fizetni minden eladott jármű után, de ez a díj reálisan sokkal kisebb lehet mint a mai összegek. Ami szinte biztos, hogy ennek a költségeit ugyanúgy a vásárlókkal fogják végső soron megfizettetni, vagy egy autóhoz tartozó kötelező éves díjjal ellátott szolgáltatás keretében, vagy a vételárba kalkulálva a jármű megvásárlásakor. A biztosítási cégek tehát komoly energiával keresik a lehetőségeiket, számukra élet-halál kérdése, hogy megtalálják-e a helyüket az átalakulás után is, ugyanis a ma ismert szolgáltatásaik piaca egyre csak zsugorodni fog.

### 1.7. Morális kérdések

Az autonóm járművek revolúciójával a közúti szerencsétlenségek gyakorisága minden bizonnyal csökkenni fog, ugyanakkor teljességgel eltűnni nem fognak a balesetek. Felmerül hát a kérdés, kit védelmezzon a jármű viselkedésével egy elkerülhetetlen ütközés alkalmával? Minden esetben az autóban ülőket? Esetleg válassza azt a megoldást amivel a legtöbb ember menekül meg? És mi a helyzet, ha egy életmentő kutya és egy köztörvényes bűnöző elgázolása közt kell döntenet? Olyan kérdések ezek, amelyekre nem feltétlenül létezik "jó" válasz, megválaszolásuk mégis szükségszerű az autonóm járművek magasabb szintjeinek eléréséhez.

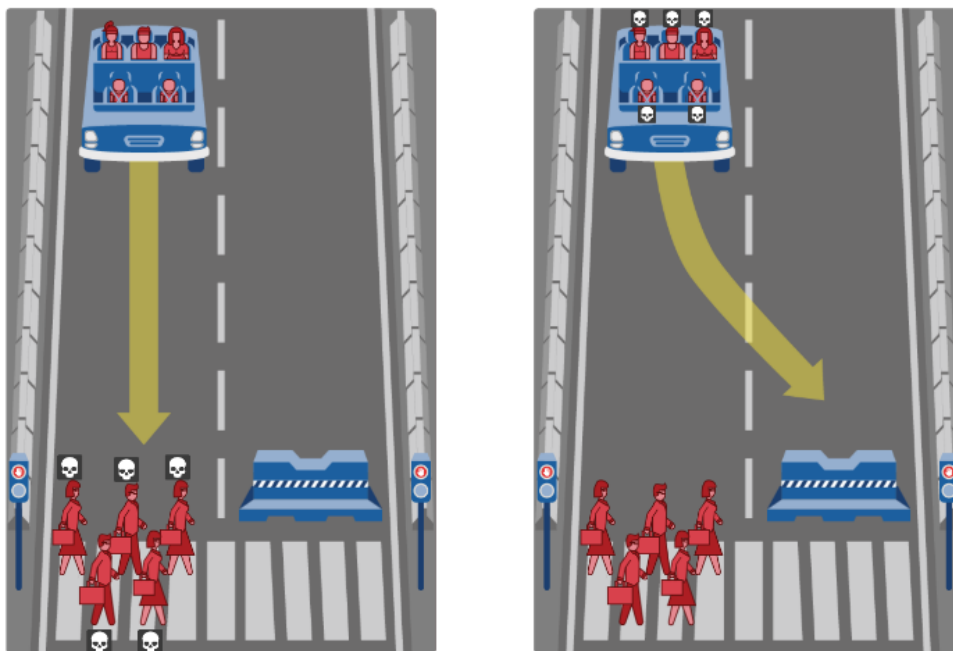
Ehhez hasonló problémák felvetésével igyekeznek információt gyűjteni például a Massachusettsi Műszaki Egyetem által működtetett, Moral Machine nevű weboldal [22] arról, hogy az emberek milyen szempontokat mérlegelve és hogyan döntenének az egyes helyzetekben. Az itt megtalálható kérdések gyakorlatilag a már az 1900-as évek elejétől ismeretes Trolley-problémák közé tartoznak, melyek jellemzően arra a kérdésre világítanak rá, hogy helyénvaló-e cselekvőleg gátolni egy egyén hasznosságát, amennyiben ez a cselekedet más szereplők számára nagyobb hasznosságot eredményez.



1.6. ábra. Az eredeti Trolley-dilemma vázlata

Az önvezető járműveket övező morális probléma témakörének másik fele abból származik, hogy érthető okokból kifolyólag nagy mértékben csökkenne a vevői köre egy olyan autonóm közlekedési eszköznek, amiről kiderülne, hogy egyes helyzetekben nem a benne ülők védelmét tekinti első számú szempontnak.

A fentiek alapján tehát a következő komoly ellentmondás rajzolódik ki, melynek megoldása hosszas társadalmi és jogi vitákkal teli folyamatként jelentkezhethet: Egyszerre kell megfelelni az egymással láthatóan gyakran ellentétes vásárlói és a társadalmi elvárásoknak úgy, hogy közben a vállalatokat a profit motiválja.



1.7. ábra. *Önvezető autókkal kapcsolatos morális kérdések [22]*

Mindemellett felmerül egy másik szempont is, amit szintén itt a morális kérdéseknél említenék meg. Sokak kifejezetten az élvezetért ülnek autóba, nem pedig az A-ból B-be eljutás céljával. Kérdés hát, hogy helyes lenne-e elvenni tőlük a vezetés lehetőségét, amennyiben az autonóm járművek biztonságosabb közlekedést nyújtanak. Láthatjuk tehát, hogy az önvezetés megvalósulása és elterjedése rengeteg komoly akadállyal küzd, melyeket egy csapásra megoldani nem lehetséges.

A jogi és morális problémák ismertetésekor egyértelművé válhatott az olvasó számára, hogy az automatizáltság szintjének növekedésével egyre inkább felelős döntéseket bízunk egy számítógép által vezérelt rendszerre, amelyet programozásakor akarva-akaratlanul felkészítünk olyan döntésekre, ahol károkozásokról, állatok vagy akár emberek életéről van szó. Dolgozatunk témája egy előfejlesztés alatt álló koncepció, ezért nem célja ezen problémák további tárgyalása, de mindenképpen meg kívántuk említeni ezt a tényezőt is.

## 2. fejezet

# A projekt bemutatása

Ezen fejezet a Knorr-Bremse Fékrendszerek Kft-nél futó Highway Pilot projekt rövid bemutatását hivatott ellátni, amellyel egy autópályára szánt level 3-as autonóm rendszer előfejlesztését tűzte ki célul a cég. A dolgozat fő témája a kockázatelemzés, a környező forgalmi szituáció osztályozása és a jármű viselkedésének megtervezése a rendelkezésre álló szenzoradatok és globális információk alapján, ezek előállításának menetét nem célunk tárgyalni, bemenetként rendelkezésre állnak.

### 2.1. A tesztjármű architektúrája

Az előfejlesztési osztály rendelkezik egy saját tesztjárművel, amely sokrétű szenzorrendszerrel és különböző számítási egységekkel van felszerelve. A járművön kiépített architektúra lehetővé teszi a Matlab [23] Simulink toolbox [24] segítségével készített modellekből generált algoritmusok futtatását, gyors prototípus tervezésre lett összeállítva. A rendszer elemei funkcionalitásukat tekintve a következők:

- Infotainment ECU<sup>1</sup>
  - globális helymeghatározás
  - útvonaltervezés és HD térképezés
- Connectivity ECU
  - V2X járműkommunikációs adatok gyűjtése
- Environment Detection ECU
  - objektumok detektálása, osztályozása
  - szenzorfüzió
  - globális útvonalkövetés

---

<sup>1</sup> ECU: electronic control unit, elektronikus vezérlőegység

- lokális szenzoradatok és globális térképadatok fúziója
- Decision Making ECU
  - forgalmi szituációk elemzése
  - viselkedéstervezés
  - trajektória tervezése
  - jármű állapotának figyelése és dinamikai korlátok becslése
  - jármű mozgásának szabályozása
  - beavatkozó szervek kezelése és összehangolása

A felsorolásban látható feladatok jelentős részét nem célhardverek, hanem gyors prototípus-tervezésre [25] fejlesztett egységek végzik. Ez természetes, mivel a tesztjárművön kipróbált fejlesztések egy-egy koncepció működőképességét hivatottak bizonyítani, előfejlesztési munkálatok során nem cél késztermékeket előállítása.

Az architektúra részletesebb bemutatása, az előbbieken említett egységek száma és összetetésük módja a dolgozat számára kitűzött feladat szempontjából lényegtelen, további részletezésük nem szükséges.

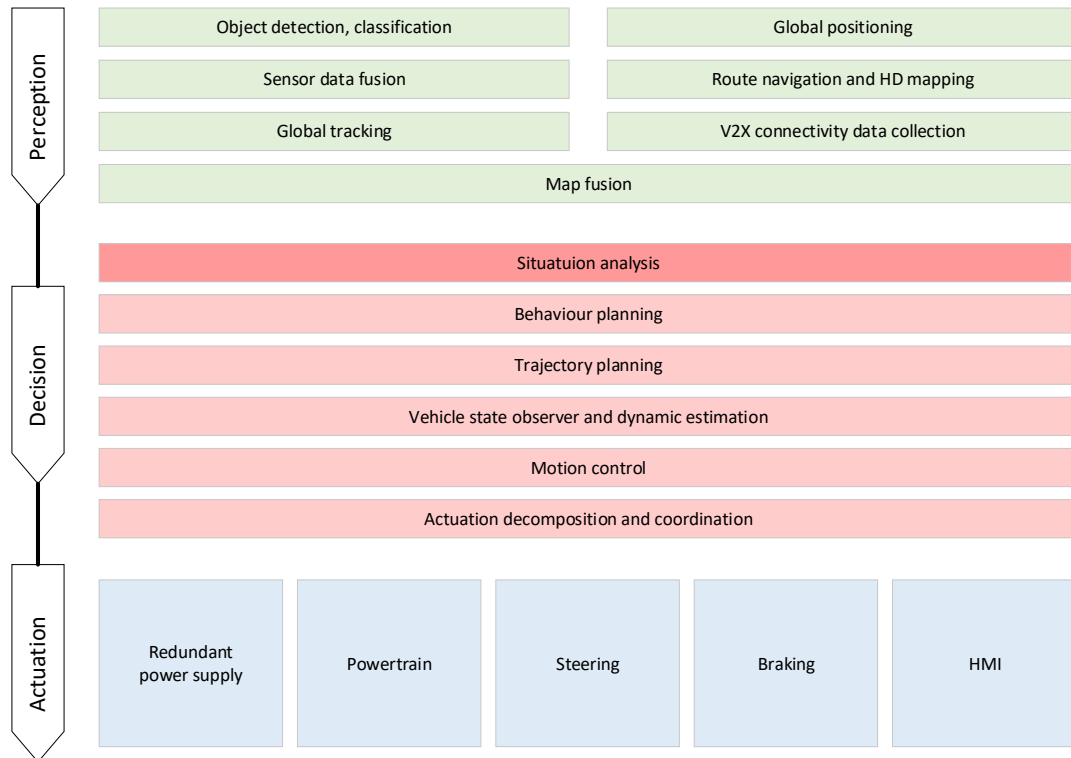


**2.1. ábra.** Az előfejlesztési osztály tesztautója [13]

## 2.2. Rendszerarchitektúra

A 2.2. ábra egy általános felépítést mutat, amely gyakorlatilag minden önvezető járműre, így a miénkre is könnyedén alkalmazható.

Három nagy egységet, a környezetérzékelést (zöld), a döntéshozást (piros) és a beavatkozást (kék) szokás elkülöníteni egymástól. Ezek közül talán a döntéshozás feladatköre a legnehezebben megfogható és talán ettől a legizgalmasabb, hiszen nem létezik még olyan megoldás



2.2. ábra. Highway Pilot (HP) funkcionális architektúrája [26]

az itt lévő problémákra, amelyre azt mondhatnánk, hogy mindenképpen az a követendő út. A diagramon élénkebb piros emeli ki a *Situation analysis* részegységet, melynek része a kockázatelemzés és a forgalmi szituációk osztályozása.

### 2.3. Környezetérzékelés

A tesztjármű a környezetét számos szenzorral detektálja. A vontatón minden irányban radarok helyezkednek el, illetve az ezek által lefedett területeket kamerák is figyelik. Az érzékelők elhelyezésekor fontos szempont volt az, hogy minden adat szenzorfüzió<sup>2</sup> segítségével álljon rendelkezésre. Kamerákkal találkozhatunk a pótkocsin is, illetve annak végére egy lidar is került.

A szenzorok jeleinek feldolgozása nem témája a dolgozatunknak, adataik a fejlesztendő algoritmus számára absztrakt formában rendelkezésre állnak. A teljesség igénye nélkül ezen adatok a következők:

- *Sávok adatai:* pozíció, típus, sebességkorlát, navigációs szempontból megfelelő-e, stb.
- *Elválasztó vonalak adatai:* harmadrendű polinom együtthatói, típus, stb.

<sup>2</sup> Szenzorfüzió: Egy adat kettő vagy több, egymástól eltérő mérési elven alapuló érzékelő jeleinek alapján történő mérése. A füzió megvalósításakor az adott szenzorok kedvező tulajdonságait használjuk ki, eltérő működésük miatt hibájuk is eltérő jellegű és mértékű lesz, így összességében pontosabb következtetéseket tudunk levonni.

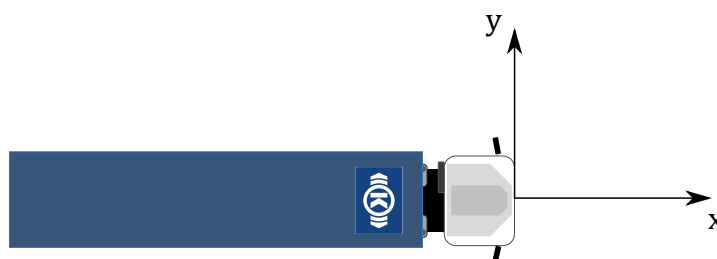
- *Környező objektumok adatai:* sebesség, pozíció, osztály (autó, motoros, gyalogos, stb.), használt sáv, stb.



2.3. ábra. A tesztjármű szenzorrendszere [13]

A jármű közvetlen környezetét figyelő szenzorokon kívül természetesen további adatforrások is vannak, ilyenek például a GPS jelei, a V2X, WLAN és LTE hálózati kapcsolatok, illetve az autó beavatkozó szerveit figyelő érzékelők is szolgáltatnak értékes információkat.

A környezetérzékelés feladataért az LSA (Local Situation Awareness) elnevezésű csoport felel, az általuk létrehozott adatok az *LSA interfészen* keresztül érhetőek el, mi is az itt érkező adatok alapján elemezzük a mindenkori szituációt. Fontos megjegyezni továbbá, hogy minden adat a saját (ego) jármű rendszámátlájának koordináta rendszerében értendő, melyet a 2.4. ábrán láthatunk.

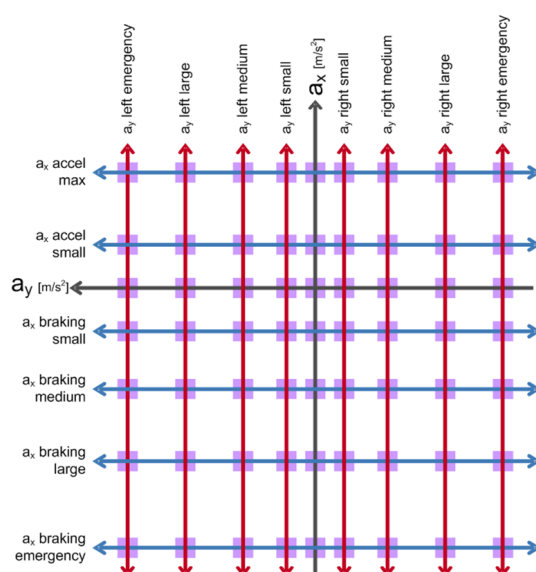


2.4. ábra. Rendszámátlá koordináta rendszer

Az *LSA interfész* pontos adatstruktúrája a dolgozat szempontjából irreleváns, ezért nem is célunk tárgyalni. A későbbiekben a problémák konkrét tárgyalásakor a megértéshez feltétlenül szükséges részei még említésre kerülnek.

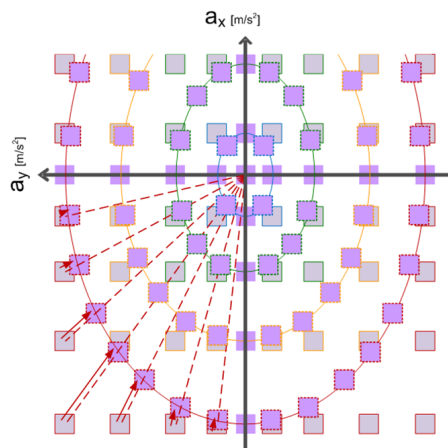
## 2.4. Döntéshozás

A környezetérzékelésből kapott absztrakt információk feldolgozása után a viselkedéstervező modul feladata lesz egy stratégiai jellegű manőversorozat számítása. Egy manővert az autó vonatkoztatási rendszerében definiált sík egy pontjaként adunk meg, amely a jármű longitudinális és laterális gyorsulásai szerinti koordináta-rendszerben helyezkedik el. A sík origójában a sebesség- és sávtartáshoz szükséges gyorsulások találhatók. Az origótól távolodva egyre nagyobb dinamikával jellemezhető manőverek találhatók. A függőleges tengely sávtartó manővereket jelöl, ettől jobbra az ebbe az irányba eső sávváltások, valamint értelemszerűen a másik oldalon a balra sávváltások helyezkednek el. Ezen manővertér diszkretizálásával a 2.5. ábrán is látható módon  $7 \times 9 = 63$  manőverpontot kezelünk.



2.5. ábra. Az autó vonatkoztatási rendszerében definiált manővertér [26]

Az autó kerekének dinamikai jellemzői nagy longitudinális irányú gyorsulások/lassulások esetén kisebb oldalgyorsulást engednek meg, ezért a tér szélein lévő pontok az origó felé tolnak. [27]



2.6. ábra. Dinamikai korlátok figyelembevétele a manővertérben [26]

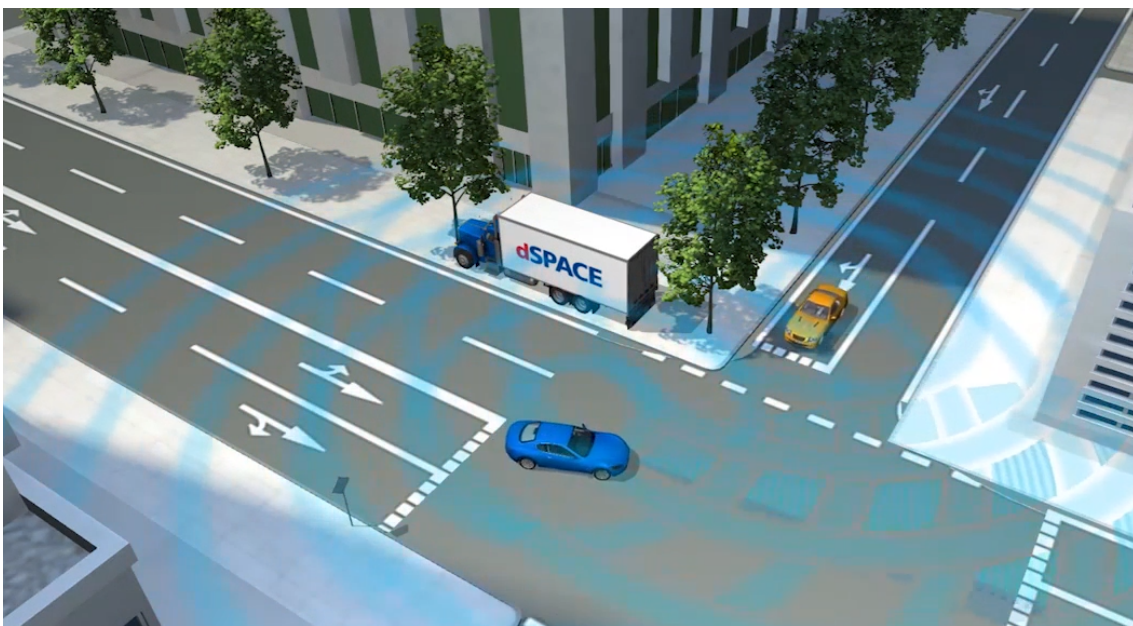


A manővertér pontjai közül valós időben kell választanunk, ezért tetszőlegesen finom kvantálása nem megengedett. A pontok számának növelésével ugyanazt az algoritmust nagyobb adathalmazra kell lefuttatnunk, amely a korlátos számítási kapacitás miatt egy bizonyos ponton túl nem összeegyeztethető a futásidő felé támasztott elvárásokkal. Dolgozatunk egyik témája, a viselkedéstervező modul közvetlen kimenete egy, az előbb ismertetett módon definiált manőverpont kiválasztása. A rendszer a kockázatokat szintén ezen manőverpontokhoz rendeli.

## 2.5. Szimulációs környezet

A tesztjárművön futtatandó algoritmusok kipróbálására egy szimulációs környezet is biztosított, ehhez a dSPACE [28] termékei nyújtanak segítséget:

- *ModelDesk*: Grafikus felhasználói felület online vagy offline szimulációk futtatására, modellek parametrizálására, útszakaszok készítésére, manőverek szerkesztésére, forgalmi szituációk készítésére, motorok modellezésére, stb. [29]
- *ControlDesk*: Járművek elektronikus vezérlőegységeinek tervezését segítő program, alkalmas a szimuláció és a valós jármű adatainak kezelésére is. [30]
- *MotionDesk*: A szimuláció vizualizálását megvalósító program, segítségével háromdimenziós környezetben láthatjuk a jármű mozgását. [31]
- *VEOS*: PC-alapú szimulációs környezet virtuális vezérlőegységek és buszrendszerek szimulálására. Számunkra a Simulink Coder által generált C kódok integrálásának támogatása miatt fontos ez a program. [32]



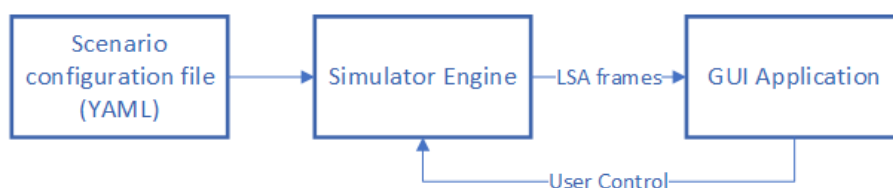
2.7. ábra. dSPACE MotionDesk [28]

Az eddig említett környezet a témák szempontjából az algoritmusok előrehaladott teszteléskor fog fontos szerepet játszani. Forgalmi szituációk generálására és egyszerűbb modell alapján történő szimulációjára alkottunk egy saját programot, amely kétdimenziós nézetben csak az LSA interfész adataihoz hasonló absztrakciós szinten üzemel. Ennek segítségével algoritmusainkat a kezdeti stádiumban sokkal gyorsabban tudtuk fejleszteni és tesztelni.

Az általunk írt szimulátor Python programozási nyelven készült. A Python egy magas szintű, általános célú programozási nyelv, melyhez kiterjedt és széleskörű standard könyvtár, illetve egyéb nyilvános modulok gyűjteménye érhető el bárki számára, ami nagyban megkönnyíti a gyors fejlesztést.

A program egyik fő komponense a szimulációs motor, amely a 3.3.1. szakaszban leírt modell alapján, a fejezetben már említett LSA interfésszel leírható időpillanatokot generál a beállított szimulációs intervallumnak megfelelően.

A szimulátor másik fő komponense egy grafikus megjelenítő, amely a motor által létrehozott keretek tartalmát jeleníti meg felülnézetben. Az alkalmazás e részének az elkészítéséhez PyQt grafikus keretrendszert használtuk fel, mely a platformfüggetlen Qt framework alkalmazását teszi lehetővé a Python előnyeit kihasználva.



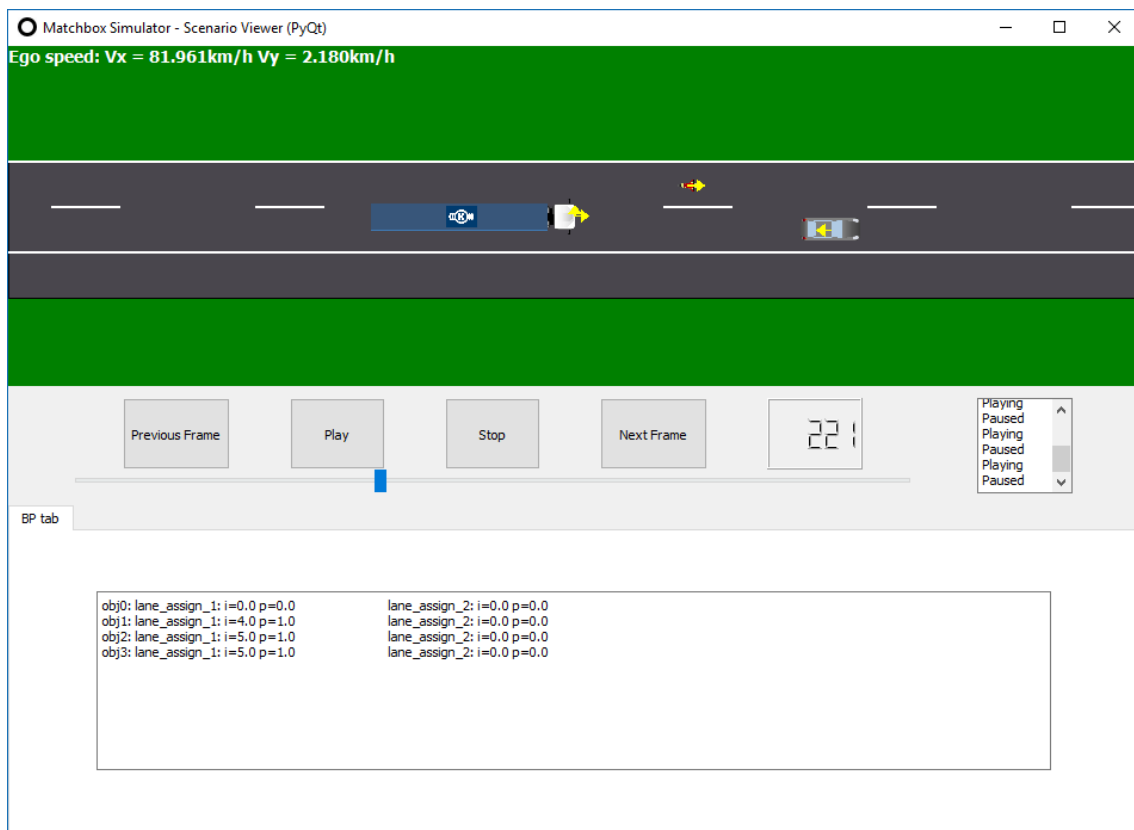
**2.8. ábra.** A szimulátor sematikus felépítése

Ahogy az a 2.8. ábrán is látható, a GUI felület elválzik a motortól, megkönnyítve a program későbbi bővíthetőségét. A szimulálandó jelenetsort egy YAML fájl írja le, amelyben következő paramétereket állíthatjuk be:

- a szimuláció lépésköze,
- globális kiindulási információk, például:
  - létező forgalmi sávok típusa,
  - és domborzati viszonya,
  - a forgalmi sávokat elválasztó vonalak paraméterei,
  - az ego jármű sebesség- és gyorsulásvektora,
  - az ego jármű méretei,
- a peer járművek kiindulási adatai, pl.:
  - az ego járműhöz képesti relatív pozíciójuk,
  - sebességvektoruk,

- gyorsulásvektoruk,
- a jármű osztálya, méretei,
- az ego és a peer járművek viselkedése, ami magába foglalja:
  - az egyes manőverek (opcionális) belépési- vagy kilépési feltételeit,
  - az akcióhoz tartozó gyorsulási profilt,
  - a manőver céljaként meghatározott sebességet,
  - és forgalmi sávot.

A struktúra úgy lett megalkotva, hogy a későbbiekben igény szerint bővíthető legyen további funkciókkal. A grafikus felület jelenlegi állapota a 2.9. ábrán látható.



**2.9. ábra.** A szimulátor GUI felülete

## 3. fejezet

# Valósídejű kockázatelemzés

Ezen fejezet a döntéshozás egyik első lépését és a hozzá szükséges általunk megalkotott módszert mutatja be. Írásunkban első lépésként kitérünk a funkció fontosságának és szerepének hangsúlyozására, ezt követően bemutatjuk az rendszerünk elméleti háttérét a döntések okára és a felhasznált irodalmi forrásokra is kitérve, majd bemutatásra kerülnek az implementált megoldás eredményei és a továbblépés lehetőségei.

### 3.1. Feladat megfogalmazása

A döntéshozó logikai blokk, több különálló elemet foglal magába, melyek közül az első az előfeldolgozást végző rész. Ennek a modulnak is számos feladata van, ezek közül az egyik legfontosabb a kockázatszámítás, amely minden korábban bemutatott manőverponthoz egy konkrét kockázat értéket rendel.

A *Maneuver Risk Calculation* egy többlépéses algoritmus, melynek során különböző kockázati komponensek kerülnek kiszámításra a manőverekhez, melyekből végül súlyozott összegként kapjuk a *Total risk*-et:

- A **base risk** két részből tevődik össze, az ego jármű dinamikus állapotából (mennyire vagyunk közel a manőver során a tapadásvesztéshez), valamint a pozíciójából (sávváltás közben csak egy felfestett vonalat érzékelünk pontosan, mivel a többi távolabb helyezkedik el, ezt a helyzetet kockázatosnak tekintjük, illetve egy sávváltás már önmagában nagyobb kockázattal jár, mint a sávban haladás).
- A **restriction risk** az ego jármű különböző lehetséges hibáiból adódik, például egy felismert bal oldalon lévő szenzorhiba következtében minden balra sávváltás maximális kockázatot jelent.
- Az **occupancy risk** a peerek pozícióját vizsgálja, így például a közvetlenül ego melletti peerek nagyon megemelik a feljűk történő bármiféle sávváltás kockázatát.

- A **detailed risk** pedig egy részletes trajektória analízis eredményeit tükrözi.

A részletes kockázat számítás még nem létezett se implementálva se elméletben amikor a Highway Pilot (HP) projekthez csatlakoztunk, csupán elképzelések voltak rá nézve, így munkánk egyik része ennek a hiányosságnak a megszüntetését célozza meg.

További elképzelés még, hogy a megalkotott rendszer a későbbiekben párhuzamos működésre legyen képes más megközelítésű kockázati kalkulációkkal, ez akkor lehetséges, ha kellően fejlett és gyors lesz a módszer minden típusú veszély felderítésében. Mindez segítene a redundáns működés megvalósításában, ami a *functional safety* szempontjából mindenképpen előnyös lépés lenne, továbbá úgy gondoljuk, hogy rugalmasabb, az emberhez hasonlóbb vezetést érhetnék el így.

## 3.2. Saját rendszer felépítése

A rendszer tervezése során, elsődlegesen a kockázat jelentését volt szükséges lefektetni, ezt követően számba vettük a használható bemeneti és szükséges kimeneti lehetőségeket és követelményeket, majd ezek alapján az alábbiakban ismertetett megoldás mellett döntöttünk.

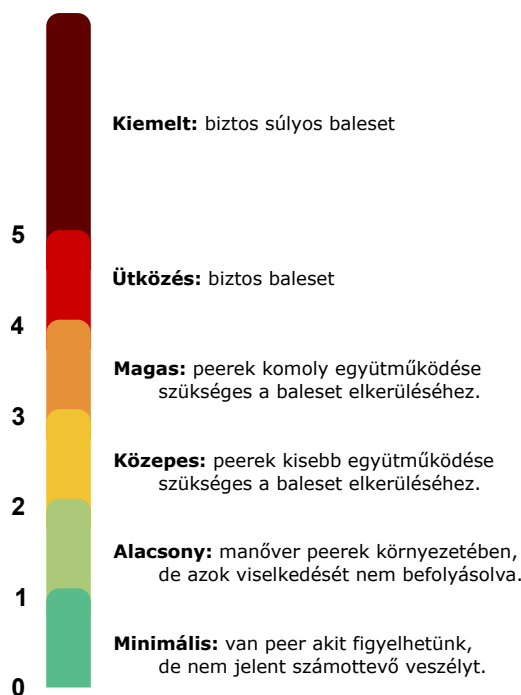
### 3.2.1. Kockázat definiálása

Egy kockázatbecslő tervezéséhez első lépésként szükséges meghatároznunk, hogy mit tekintünk kockázatnak. Autonóm járműves környezetben praktikus az azt jelenti, hogy milyen helyzeteket mennyire szeretnénk elkerülni.

Kézenfekvőnek tűnhet a megközelítés, hogy a kockázatot a forgalmi szituációban résztvevő személyek fizikai sérülésének valószínűségével és súlyosságával tegyük arányossá. Ekkor azonban a kis sebességű ütközések gyakorlatilag kockázatmentesnek adódnának, miközben egy ilyen esemény is megállást, papírmunkát, komoly idővesztést és javítási költségeket követel. Egy vezető nélküli járműnél a járulékos teendők még ennél is drasztikusabbak lehetnek, előfordulhat, hogy egy csekély baleset következtében a jármű nem tudja önerőből eldönteni, hogy biztonsággal továbbhaladhat vagy sem, ilyenkor az út szélén várakozna mindaddig, amíg egy ember a segítségére nem érkezne. Ezen szempontokat is figyelembe véve praktikusabb, ha az ütközés tényét már önmagában büntetjük, és annak súlyossága ezt a kockázatot csak növelni tudja.

Minden konkrét manőver esetén egy nullánál nem kisebb valós számmal szeretnénk jellemezni a kockázatot. Nullával jelöljük azt a helyzetet, amikor minden objektumtól távol, egyenesen haladunk az autópálya egy folyamatosan használható sávjában. A számértékek további értelmezéséhez a 3.1. ábra nyújt támpontot. Elvi felső határ nincs definiálva a kockázathoz, mivel mint azt később a 3.3. fejezetben látni fogjuk, a gyakorlati felső korlátot az algoritmus paraméterei nagyban befolyásolják. Ugyanakkor ennek komoly jelentősége

nincs, az öt fölötti értékek minden esetben szinte teljesen biztos és egyben súlyos balesetet jeleznek előre. Ebben a tartományban az eltéréseket a balesetben érintett peerek száma és az ütközések relatív sebességei befolyásolják, ez olyan kiugró esetek észrevételét teszi lehetővé, mint például egy tömegbaleset vagy egy szemből érkező jármű az autópályán.



3.1. ábra. Risk értékek értelmezése

### 3.2.2. Konceptió

**Célok meghatározása** A projekt célja egy olyan rendszer koncepciójának megalkotása és megvalósítása, amely képes valós időben – azaz 33 fps sebességgel – ellátni a részletes kockázat analízis feladatát minden lehetséges manőverre nézve. Előny továbbá, ha a rendszer számításának részletessége a későbbiekben könnyen bővíthető marad a valós idejű követelmények betartása mellett.

**Bemenetek** A kalkuláció a környezetfelismerés által szolgáltatott interfésztől (valamint az ebből számolt, kiegyenesített sávokba transzformált adatokból) és a jármű állapotvektorából veszi az bemeneteket képkockánként, azaz mindig csak az aktuális információkat használja fel, a korábbiakat nem. Ugyanakkor közvetetten ez az állítás már nem igaz, hiszen például egy objektum aktuális gyorsulásának a meghatározásához értelemszerűen több egymást követő pillanat adatait használjuk fel, így tekintve a részletes kockázat analízis bemenete az objektumok tulajdonságainak időbeli változásáról is tartalmaz információt. A jelenleg használt bemenetek még bővíthetnek majd a későbbiekben a járműmodell komplexebbre cserélésével. A következő adatok a már említett ego jármű rendszám-tábla koordináta-rendszerében értendők.

- Felhasznált bemenetek a jármű állapotvektorából:
  - `vx` - longitudinális ego sebesség
  - `vy` - laterális ego sebesség
  - `ax` - longitudinális ego gyorsulás
  - `ay` - laterális ego gyorsulás
  - `depth` - ego járműszerelvény hossza
  - `width` - ego járműszerelvény szélessége
- Felhasznált bemenetek közvetlenül a környezetfelismerés interfészéről:
  - Sáv információk:
    - \* `type` - sáv típusa
    - \* `slope` - sáv meredeksége
    - \* `type_next` - sáv következő típusa
    - \* `dist_type_change` - sáv típusváltozásának előre látható távolsága
  - Vonal információk:
    - \* `a0` - a négy legközelebbi vonalról
    - \* `a2` - a két legközelebbi vonalról
    - \* Megjegyzés: A környezetfelismerés harmadrendű polinomként kezeli a felismert vonalakat, amelyek lehetnek felfestések, de az útburkolat széle is. Az  $a_0$  együttható a kiegyenesített vonal  $y$  irányú távolságát jelöli, míg az  $a_2$  az eredeti görbe vonal  $x = 0$ -ban vett görbülete. A járműmodellből adódóan az objektumok elfordulása nem lehetséges, így nincs szükség a vonalak  $a_1$  együtthatóira.
- Felhasznált bemenetek a kiegyenesített sávokra transzformált objektumlistából:
  - `dx` - relatív longitudinális távolság az ego-tól
  - `dy` - relatív laterális távolság
  - `vx` - relatív longitudinális sebesség
  - `vy` - relatív laterális sebesség
  - `ax` - relatív longitudinális gyorsulás
  - `ay` - relatív laterális gyorsulás
  - `width` - objektum szélessége
  - `depth` - objektum hossza
  - `obj_class` - objektum típusa
  - `det_side` - objektum érzékelt oldala

} 16 objektumhoz

**Kimenetek** A blokk kimenete az a  $7 \times 9 = 63$  darab nullánál nagyobb vagy egyenlő valós szám mátrixa, amely az egyes manőverek potenciális végrehajtásának kockázatát jellemzik az adott pillanatban, ezeket egy ún. *Maneuver Library*-ben tároljuk a kiszámítást követően.

A számítás módjára a rendelkezésre álló bemeneti adatok és az elvégzett irodalomkutatás alapján egy nagyszámú trajektória mentén való szimuláció lefuttatása adódott az egyetlen olyan megoldásnak, amely valóban részletesnek tekinthető, explicit képletekkel ugyanis komoly nehézséget jelent minden fontos részlet megragadása. A fő probléma ezzel az iránnyal az, hogy már az ebben a dokumentumban bemutatott első, legegyszerűbb járműmodellt alkalmazó szimuláció is igen komoly számítási igényeket támaszt. A távolabbi elképzelések pedig sokkal részletesebb verziókat remélnék, így viszont a rendszer még akkor sem lenne bővíthető, ha komoly munka árán sikerülne egy optimalizált kóddal beleférni a valósidejű határok közé. Egy árérzékeny piacra, haszongépjárművekhez fejlesztett rendszer esetében pedig komoly szempont a hardware-költségek alacsonyan tartása, így ha van rá lehetőség, akkor a szükséges számítási kapacitást alacsonyan kell tartani.

Az itt felvázolt probléma leküzdésére merült fel az ötlet, hogy a gépi tanulást, azon belül is a neurális hálókat próbáljuk alkalmazni a kimerítő szimuláció kiváltására. A koncepció lényege, hogy nem engedjük el a számításigényes kalkuláció adta előnyöket, csupán az nem valós időben és nem is a járművön lévő hardveren futna le, hanem már a fejlesztés közben az irodában egy nagy és reprezentatív adathalmazon, majd ennek az eredménye alapján egy megfelelő neurális háló olyan gyorsabban számítható kapcsolatot találhat a bemeneti és a kimeneti adatok között, amelynek az explicit matematikai felírására kevés esély mutatkozik.

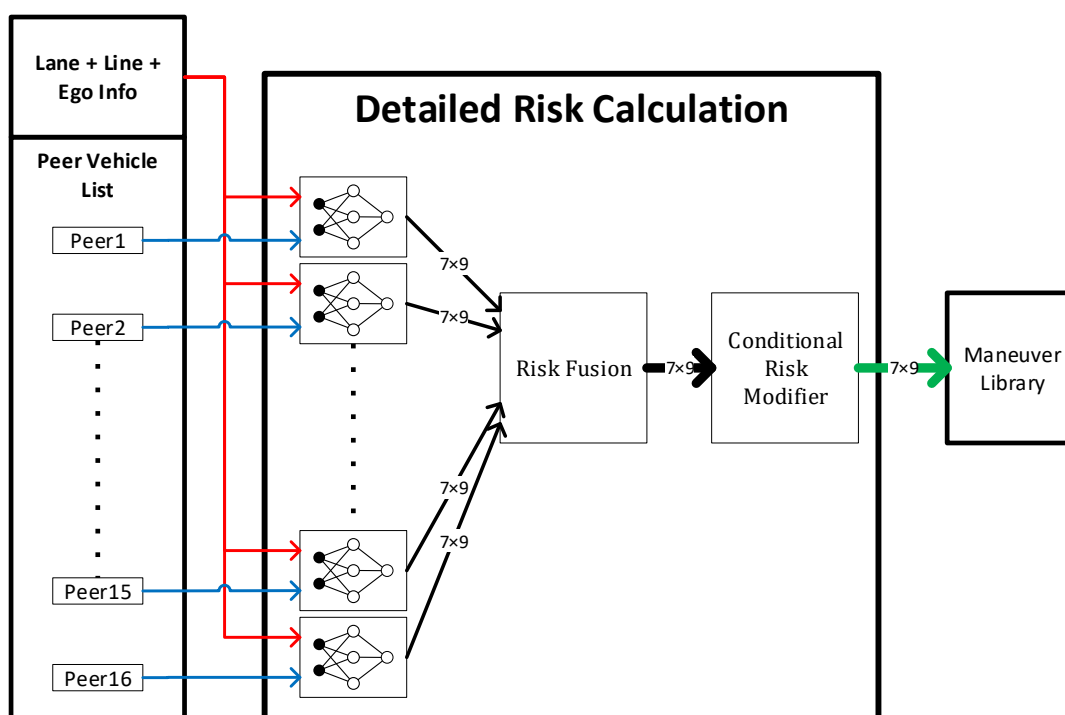
Más publikált munkában ezzel az ötlettel nem találkozni, igaz általában sem igazán találni valóban részletes kockázatszámító megoldásokat az autonóm járművekhez kapcsolódóan. Az elképzeléshez a motivációt és a benne lévő valós potenciált a neurális hálók utóbbi időben elért eredményei és egyre szélesebb felhasználási területei biztatták. Növekvő sikerrel alkalmazzák a neurális hálókat különböző magas dimenziójú parciális differenciál egyenletek gyors megoldására[33], kiemelt eredmények születtek például a folyadékok dinamikájának terén[34]. Szintén eredményesen alkalmazzák a technikát egyes mátrixműveletek gyorsítására[35]. Komoly eredményeket értek el kutatók az elmúlt időben továbbá a kaotikus dinamikai rendszerek viselkedésének megjóslásában is, ezekben a projektekben is a neurális hálók segítségével született az áttörés[36]. Ezek természetesen nem ugyanazok a feladatok mint amire itt kerül alkalmazásra a technika, de hasonlóságokat könnyedén felfedezhetünk köztük.

Fontosnak tartjuk még egyszer kihangsúlyozni az elképzeléssel kapcsolatban annak legfőbb potenciálját, miszerint amennyiben sikerül megfelelő pontosságot elérni a gépi tanulással, akkor egy olyan rendszer lesz a kezünkben, amelynek mintáján keresztül jó eséllyel reméljük, hogy a háttérben húzódó analitikus számítás tetszőleges részletezése mellett is képesek leszünk annak előnyeit valós időben kihasználni, hiszen egy betanított háló futtatása még nagyon mély architektúra esetén sem jelent jelentős futásidőt. Emellett a neurális hálók általánosító képességéből származó előnyben is komoly remények vannak. Ez alatt a rendszer azon képessége értendő, hogy olyan szituációkban is esetleg megfelelő döntést tudjon hozni, amire a szimulációs kalkuláció nem lett felkészítve és emiatt az el is rontaná, de mivel a tanítóadatokban sem szerepel ilyen minta, így a háló által megtanult szabályok



esetlegesen jobban is működhetnek mint az emberi gondolkodással megalkotottak, s ezzel ilyen esetekben is helyes döntést hozhatna. Ezt a tulajdonságát természetesen nehéz tetten érni a tesztek során, hiszen a címkéző algoritmus tervezésénél az összes tanító adat helyessége volt a cél, ezáltal nem tudni, milyen esetekben tévedhet mégis.

A rendszer tervezését és implementálását a plan-do-check-act (PDCA) ciklikussága jellemezte, így menet közben több módosítás is került bele. Ezekre a változtatásokra a 3.3. és 3.4. fejezetekben rövid kitérőt találhatunk, itt azonban csak az elkészült rendszer architektúrája kerül bemutatásra. A végső felépítés szerint objektumonként végzi el a szituáció kiértékelését a neurális háló és ezt ezután összegzi egy másik blokk. Ez az architektúra a tanítóadatok megfelelő előkészítésénél és a hálóstruktúra megválasztásánál is könnyebbséget jelent, valamint bizonyos szintű skálázhatóságot is biztosít. A rendszer blokk-diagramját a 3.2. ábrán vizsgálhatjuk meg. Látható, hogy jelenleg a különböző peerekhez párhuzamosan futtatjuk ugyanazt a neurális hálót, csak más bemenetekkel, ebből fakad a skálázhatóság is, hiszen ugyanezt ciklikusan illetve a kettőt kombinálva is elvégezhetnénk, ha esetleg erőforrás szűkében vagyunk.



3.2. ábra. Részletes kockázatszámítás blokk-diagramja

A rajzon látható *Risk Fusion* és *Conditional Risk Modifier*, mint azt a nevük sejteti, először a neurális hálóból érkező  $7 \times 9$ -es objektumonként adott kockázati mátrixokat kombinálja egyetlen  $7 \times 9$ -es a teljes szituációra jellemző mátrix formájába, majd további feltételek alapján még módosulhatnak egyes kockázat komponensek. Ezen blokkok belső működését a 3.3. fejezetben találhatjuk leírva, ennek oka, hogy kezdetben ezek a számítások is a címkéző algoritmus részét képezték.

### 3.2.3. Részfeladatok áttekintése

Az itt összefoglalt rendszer megalkotásához az alábbi feladatok elvégzésére van szükség, ezeket mutatjuk be a következő fejezetrészekben:

1. Kockázat analitikus számításának megtervezése.
2. Adatgyűjtés.
3. Az adatokat címkéző algoritmus implementálása az első pont specifikációja szerint.
4. Címkézett adatok előkészítése a tanításhoz.
5. Neurális háló tervezése, tanítása és a hiperparaméterter hangolása.
6. A megalkotott rendszer tesztelése Software in the Loop (SiL) környezetben.
7. A megalkotott rendszer tesztelése valós járművön.

### 3.3. Tanító adatok létrehozása

Ebben a fejezetben a neurális háló tanításához szükséges tanítóadatok generálásáról olvashatunk. Elsőként a 3.3.1. szakasz a kockázatszámításhoz létrehozott matematikai számításról nyújt információkat, ezt követően a 3.3.2. részben az ez alapján készült programkód kerül bemutatásra, végül pedig a 3.3.4. fejezetrész a címkézendő adatok összegyűjtési módszereiről szól.

#### 3.3.1. Algoritmus

Fontos azzal kezdenünk a leírást, hogy az LSA interfészen érkező objektumok pozícióját, sebességét és gyorsulását első lépésként egy kiegyenesített sávokkal rendelkező világba képi le a rendszer, ahol számos döntést sokkal könnyebben hozhatunk meg, ugyanis ezzel elimináltuk a sávok kanyarodásából adódó látszólagos ütközési pontokat, amelyek hamis hozzáadott kockázatot jelentettek volna. Ebben az esetben, mint azt később látni fogjuk, különös figyelemmel kell lennünk a jármű és a manőverek dinamikus tulajdonságainak eltolódására.

Az algoritmus is ebben a kiegyenesített sávok által megadott térben futtatott szimuláción alapszik, így az objektumok mozgása és a szimuláció egy nagyon egyszerű, gyorsan számítható folyamat. A teljes számítás mégis több percet vesz igénybe, mivel nagy számú trajektória mentén történik a számítás, és minden lépésben egy komplex kockázatelemzés és összegzés történik.

## Járműmodell

Az objektumok szimulációjához egy kezdetleges járműmodellt alkalmaz a számítás, mivel így könnyebben kidolgozható az algoritmus többi, a kockázatszámításra összpontosító része, ugyanakkor a projekt folytatása esetén már kevesebb erőfeszítéssel áttérhetünk egy komplexebb megoldásra a későbbiek során. Emellett fontos megemlíteni, hogy a döntéshozás logikájának teszteléséhez egy ilyen, egyszerűbb környezet alkalmazása még praktikusabb is az első tesztekhez, mint egy pontosabb, de egyben összetettebb környezet használata.

Az ego és a peer járművek is egy-egy téglalapként kerülnek reprezentálásra, melyek kizárólag translációs mozgást végezhetnek a koordináta tengelyek mentén, elfordulni nem képesek. Kitüntetett pontjuk, amely bármelyik oldal középpontja lehet, a 3.1. táblázatban feltüntetett tulajdonságokkal rendelkezik minden időpillanatban, a számítás során ezek frissülnek lépésről lépésre. Ezen felül minden objektumhoz rögzített szélesség és hosszúság adat, valamint objektum típus információ tartozik, ezek megadják a befoglaló téglalap oldalhosszait, illetve a megjelenítésnél és később az ütközés súlyosságának számításánál is fontos szerepet kapnak.

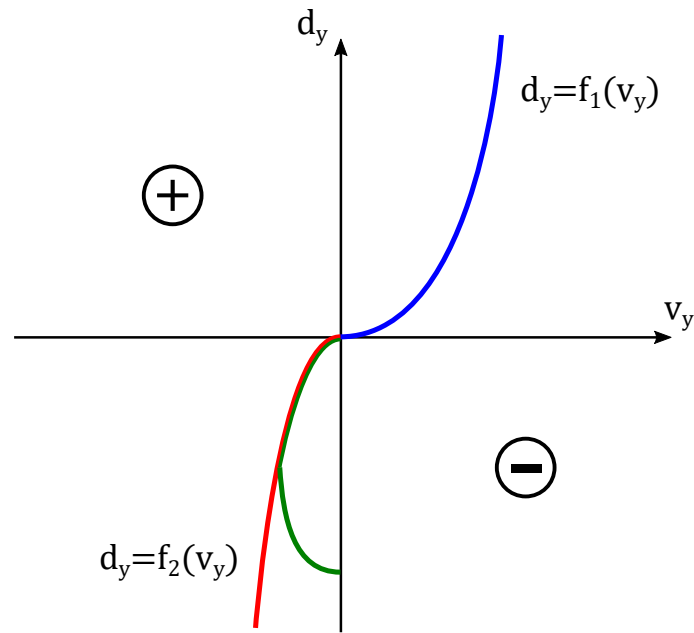
**3.1. táblázat.** *Peer inicializációs információk*

Paraméter	Leírás
$x$	Az objektum detektált pontjának $x$ koordinátája (abszolút).
$y$	Az objektum detektált pontjának $y$ koordinátája (abszolút).
$vx$	Az objektum detektált pontjának longitudinális sebessége (abszolút).
$vy$	Az objektum detektált pontjának laterális sebessége (abszolút).
$ax$	Az objektum detektált pontjának longitudinális gyorsulása (abszolút).
$ay$	Az objektum detektált pontjának laterális gyorsulása (abszolút).

A szimuláció kezdetén rögzítjük az ego jármű koordináta-rendszerét a kiindulási helyzetben, ez lesz a globális vagy más néven abszolút koordináta-rendszerünk. A számítások során az  $x$  és az  $y$  tengelyt gyakorlatilag külön kezeli a rendszer, minden szimulációs ugrás során három lépést kell végrehajtani, melyek a következők:

- *Új gyorsulásértékek meghatározása mindkét tengely mentén*  
Minden manőver végrehajtása során a jármű gyorsulását tekintve kétféle gyorsulás-állapotban létezhet, hiszen például egy sávváltás kétféle szakaszból tevődik össze, először a sáv felé gyorsulunk oldalirányban, majd pont ellenkezőleg és ezzel "ráállunk"

a sáv középvonalára. Ezt a lehetséges kétféle gyorsulásértéket már a manőver kezdetén hozzárendeljük a járműhöz, a továbbiakban  $a_+$  és  $a_-$  jelöli ezeket a konkrét értékeket. Annak eldöntése, hogy a kettő közül éppen melyik szakaszban vagyunk és így milyen laterális és longitudinális gyorsulás szükséges, két másodfokú görbe segítségével történik a  $v_y - d_y$  síkon (3.3. ábra), ahol  $v_y$  a jármű aktuális  $y$  irányú sebessége,  $d_y$  pedig a célsáv középvonalától mért  $y$  irányú távolsága. A jármű állapota a sík bármely pontjából indulhat a manőver kezdetén, a cél, hogy a meghozott szabályok hatására a tengelyek origójába jussunk el, és maradjunk is ott.



3.3. ábra. Laterális gyorsulást szeparáló görbe

$$f_1(v_y) = -\frac{1}{2a_-}v_y^2 \quad (3.1)$$

$$f_2(v_y) = -\frac{1}{2a_+}v_y^2 \quad (3.2)$$

Amennyiben a jármű állapota a görbektől balra fekvő területen vagy a piros  $f_2$  görbén fekszik, akkor a pozitív  $a_+$  oldalgyorsulás és a hozzá tartozó longitudinális gyorsulás szükséges, ellenben a negatív  $a_-$  kerül alkalmazásra. A síkon látható zöld görbe egy lehetséges sáv váltás helygörbéjét mutatja. Itt azt láthatjuk, hogy az első szakaszban  $a_-$  gyorsulást alkalmazunk, majd amikor eljutottunk a piros görbére, onnantól  $a_+$  oldalgyorsulásra váltva haladunk a piros görbe mentén egészen az origóig, innentől kezdve 0 oldalgyorsulást alkalmazunk. Fontos észrevenni, hogy minden lehetséges helygörbe az  $f_1$  és az  $f_2$  függvények megfelelően eltolt szakaszaiból állhat csak össze.

- *Új pozíció kiszámítása*

Lépésköznny időt követően a kezdeti sebességet, pozíciót és meghatározott gyorsulást

használva, az elért pozíciót az alábbi képletekkel számolhatjuk.

$$x_{n+1} = x_n + v_{x,n}t_{step} + \frac{1}{2}a_{x,n}t_{step}^2 \quad (3.3)$$

$$y_{n+1} = y_n + v_{y,n}t_{step} + \frac{1}{2}a_{y,n}t_{step}^2 \quad (3.4)$$

- *Új sebesség kiszámítása*

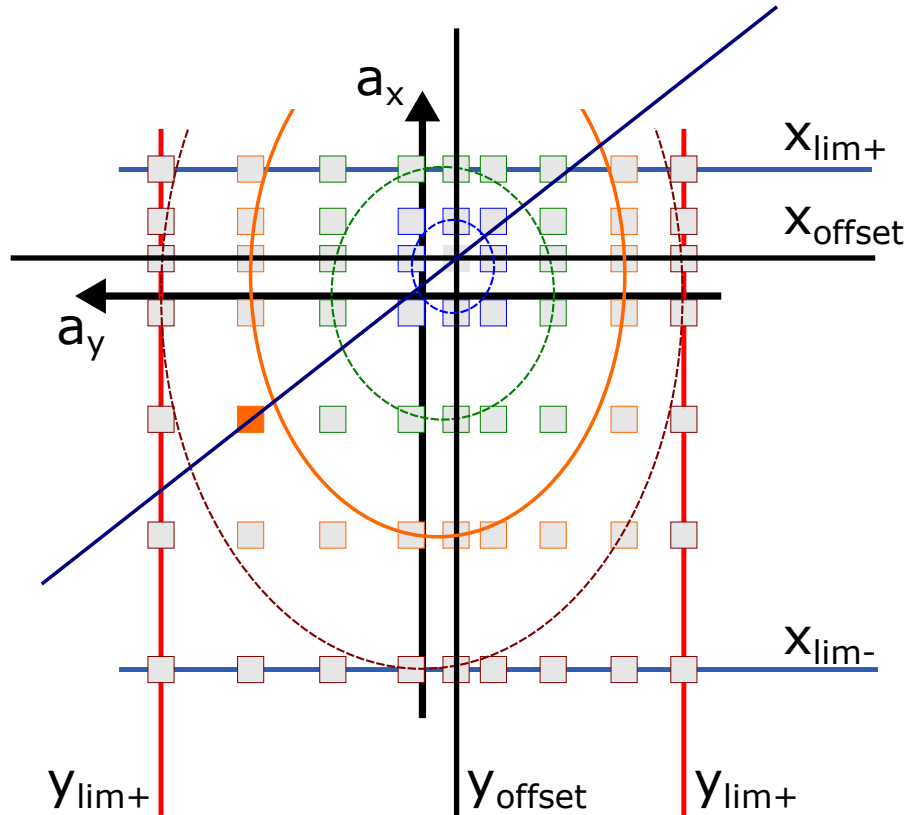
Lépésköznyi időt követően a kezdeti sebességet és meghatározott gyorsulást használva, az elért sebességkomponenseket az alábbi képletekkel számolhatjuk.

$$v_{x,n+1} = v_{x,n} + a_{x,n}t_{step} \quad (3.5)$$

$$v_{y,n+1} = v_{y,n} + a_{y,n}t_{step} \quad (3.6)$$

### Súrlódási ellipszis számítása

A manővertérben definiált manőverpontok mindegyike, mint az már említettük, szükség-szerűen egy ellipsziszre torzul, ennek a transzformációnak a módját és matematikáját az alábbiakban tekintjük át.



3.4. ábra. Súrlódási ellipszis

A 3.4. ábrán egy konkrét pont példáján kerül bemutatásra a manőver pontok ellipsziszekre

torzítása. Itt a narancssárgával jelölt pontot kívánjuk az ugyanilyen színnel jelölt ellipszisre mozgatni a kék egyenes mentén. Látható, hogy az ellipszisek középpontja vándorol, a legnagyobb a valós súrlódási ellipszis (ez jelöli a jármű tényleges dinamikai határait), így annak tengelyei egybe esnek az abszolút gyorsulások tengelyével, a belső ellipszisek viszont az eltolt tengelymetszetekhez esnek egyre közelebb, ezzel követik az eltolt manőverpontokat. A tengelyek offsetelése a valós sáv kanyarulatából és meredekségéből adódik (a későbbiekben több tényező hatását, mint például a légellenállását is tervezzük majd figyelembe venni, ezzel bővítve a modellt), mértékét a

$$x_{offset} = g \sin \left[ \tan^{-1} \left( \frac{slope}{100} \right) \right] \operatorname{sgn}(-slope) \quad (3.7)$$

$$y_{offset} = v_x^2 a_2 \quad (3.8)$$

képletek segítségével számolhatjuk ki, ahol  $g$  a gravitációs gyorsulás, a  $slope$  a saját sávunk meredeksége százalékban,  $v_x$  az aktuális longitudinális sebességünk,  $a_2$  pedig a két legközelebbi sáv vonal  $a_2$  együtthatójából számolt átlagos érték.

Ezt követően az egyenes és az ellipszis egyenletét felírjuk majd megoldjuk azokat  $x$ -re és  $y$ -ra, ezzel megkapjuk az eltolt pont új helyét.

$$y - y_{offset} = \frac{y_p - y_{offset}}{x_p - x_{offset}} (x - x_{offset}) \quad (3.9)$$

$$1 = \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} \quad (3.10)$$

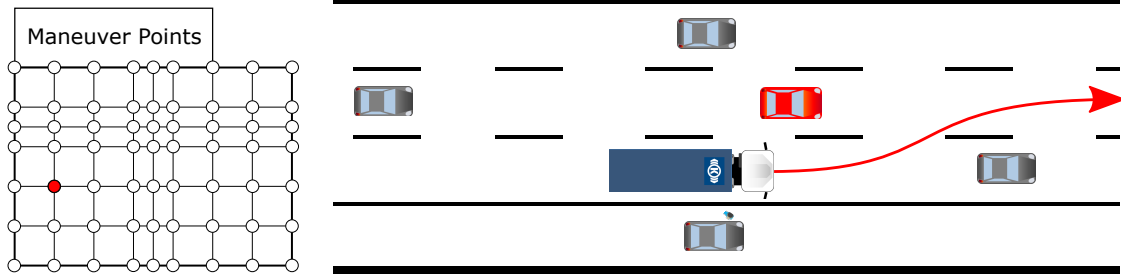
ahol a  $p$  indexű koordináták az elmozgatandó pont koordinátái, a 0 indexűek a hozzá tartozó ellipszis középpontjának koordinátái,  $a$  és  $b$  pedig ezen ellipszisek  $x$  és  $y$  tengely menti féltengelyei.

Az egyenlet megoldásának képlete igen terjedelmes, emiatt nem került ide, szimbolikus megoldó programok segítségével könnyen megkaphatjuk a felírt egyenletekből.

### Kockázati összetevők számítása

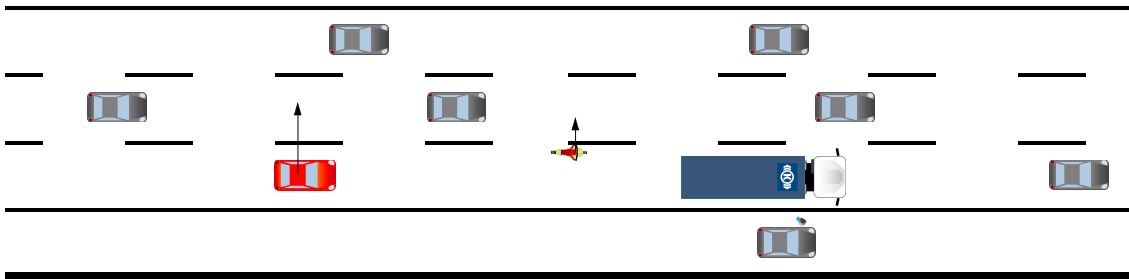
A címkéző algoritmus működését inentől egy konkrét kiválasztott ego-manőver és egy konkrét kiválasztott peer esetén vizsgáljuk meg. A 3.5. ábra segít abban, hogy konkrétan legyen mire gondolnunk egy kiválasztott manőver és peer jármű alatt. Gyakorlatban mind a  $7 \times 9 = 63$  manőverpontra és ezen belül is mind a 16 peer járműre nézve elvégzésre fognak kerülni az itt leírt számítások.

Első lépésként, ha hátulról, a saját sávunkból érkezik az objektum, akkor a laterális sebességének függvényében elhúzásra kerül vagy egy sávval jobbra vagy balra, ezt figyelhetjük meg a 3.6. ábrán szemléltetve. Ennek oka, hogy a mögöttünk lévő veszélyes közelségben lévő vagy sebességgel érkező peerek ne emeljék föl a saját sávunkban egyenletesen haladás kockázatát, amennyiben ők hátulról nekünk jönnek, az ellen nem kívánunk védekezni. További indok az elhúzás mellett, hogy hirtelen sávot válthat az ilyen típusú objektum



3.5. ábra. Kiválasztott konkrét manőver-pont és peer jármű

és nagy sebességgel előzésbe kezdhet, ilyenkor már a mi hibánk lenne, ha elé váltanánk, viszont követésnél a kötelező követési távolságot a hátul lévőknek kell betartania a KRESZ szabályai szerint. Az elhúzás irányának meghatározásához jelenleg egy egyszerű logikát használunk: a peer pillanatnyi laterális sebességének függvényében döntünk. Feltételezzük, hogy a forgalomban résztvevő társak jellemzően balról kívánnak előzéseket végrehajtani, így a határértéket  $v_y = 0$  helyett kissé érdemes eltolni a negatív irányba. Ezzel azt érzük el, hogy amennyiben a peer jármű kis mértékben jobbra tart, még akkor is a bal sávba fogjuk eltolni, és csak jelentősebb jobbra irányuló mozgásnál toljuk jobbra. Természetesen ez a megoldás nem tökéletes, viszont a peer objektumok mozgásának előre jóslása egy nagy téma, amelyről külön kutatások folynak a cégen belül is, így a későbbiekben, ha ott megfelelő eredmények születnek, akkor azt felhasználva egyszerűen bővíthetjük a modellünket.



3.6. ábra. Hátulról érkező objektumok eltolása

Második lépésként még mindig a peer jármű várható mozgásának meghatározásával foglalkozik az algoritmus, itt viszont már kicsit részletesebb modellt és kimeneti lehetőségeket veszünk számba, ugyanakkor ezt a modult is kiválthatná egy komolyabb peer mozgást becsülő modul a jövőben. Öt különböző típusú mozgásformát tartunk számon, ezek közül pontosan egyet rendelünk hozzá az objektumhoz. Ezek a mozgástípusok csak a laterális viselkedésre nézve lettek létrehozva, az objektum longitudinális mozgása minden esetben a kiindulási gyorsulásának megtartását jelenti, ugyanakkor fékezés esetén, ha teljesen megáll a jármű, akkor már nyilvánvalóan nem indul el hátrafelé ugyanazt a gyorsulást tovább követve. Az egyes mozgástípusokba tartozás feltételének meghatározásához egy általunk definiált függvényt használunk a következő formában

$$lane_{pref} = w_{pos}d_y + w_{speed}v_y + w_{acc}a_y \quad (3.11)$$

ahol:  $d_y$  – a peer jármű laterális távolsága a saját sávjának középvezetől,  
 $v_y$  – a peer jármű laterális sebessége,  
 $a_y$  – a peer jármű laterális gyorsulása,  
 $w_{pos}$  – a jármű középvezetől való eltérését súlyozó paraméter,  
 $w_{speed}$  – a jármű oldalirányú sebességét súlyozó tényező,  
 $w_{acc}$  – a jármű oldalgyorsulását súlyozó tényező,  
 $lane_{pref}$  – a jármű által preferált sávra utaló skalár érték, melynek segítségével döntést tudunk hozni a jármű viselkedéséről.

A létrehozott különböző mozgásformák megnevezését, részletes leírását és a besorolás feltételeit a 3.2. táblázatban láthatjuk, a feltételekben szereplő 1 -es és 2 -es indexű  $v$  és  $a$  értékek a számítás további beállítható paramétereit.

**3.2. táblázat.** *Peer manőver típusok*

Típus	Leírás	Tagsági feltételek
Saját sávba tartás	Az objektum célja, hogy a saját kezdeti sávjának középvezetét elérje és azt kövesse.	$( v_y  \leq v_2 \ \& \  a_y  \leq a_2)$ $\& \ (v_1 \leq  v_y  \   \ a_1 \leq  a_y )$ $\& \ ( lane_{pref}  \leq 1)$
Jobb sávba tartás	Az objektum célja, hogy a saját kezdeti sávjától jobbra lévő sáv középvezetét elérje és azt kövesse.	$( v_y  \leq v_2 \ \& \  a_y  \leq a_2)$ $\& \ (v_1 \leq  v_y  \   \ a_1 \leq  a_y )$ $\& \ (lane_{pref} < -1)$
Bal sávba tartás	Az objektum célja, hogy a saját kezdeti sávjától balra lévő sáv középvezetét elérje és azt kövesse.	$( v_y  \leq v_2 \ \& \  a_y  \leq a_2)$ $\& \ (v_1 \leq  v_y  \   \ a_1 \leq  a_y )$ $\& \ (1 < lane_{pref})$
Kezdeti dinamika tartása	Az objektum szokatlan dinamikával rendelkezik, potenciális veszélyforrás, kezdeti gyorsulás alkalmazása a teljes szimuláció során.	$v_2 <  v_y  \   \ a_2 <  a_y $
Egyenes haladás	Az objektum a sávvezetőkkel párhuzamosan halad, de nem törekszik középvezet tartására.	<b>else</b>

Az algoritmus a hagyományosnak tekinthető  $Risk = Probability \cdot Severity$  alaktól eltérően

$$R = P \cdot (1 + S) \tag{3.12}$$

formában keresi a kockázatot. Ennek oka a már fent említett megfontolás, miszerint a súlyosság csak növelni tudja a kockázat értékét.

Az ego és a peer jármű mozgását tíz másodpercig szimuláljuk előre, mivel ettől kezdve már az alkalmazott kockázati függvények mindegyike konstans zérus értéket vesz fel, ezt láthatjuk



a 3.7. és a 3.8. ábrák grafikonjain is. A trajektóriákat nem előre definiáljuk, hanem azok kiadódnak a szimuláció során a járművek meghatározott dinamikus viselkedésből.

A peer járműhöz tehát a meghatározott mozgástípus alapján célsávot (ha olyan a mozgástípus amihez tartozik), laterális valamint longitudinális gyorsulásértékeket rendelünk, és a 3.3. ábrán bemutatott módszer alapján szimuláljuk a mozgását. Az ego jármű viselkedését kilenc egymáshoz közeli dinamikával rendelkező gyorsulásérték mentén szimuláljuk ugyanezzel a járműmodellel, és az így kiadódó mindegyik trajektória esetén rögzítjük a fontos metrikákat, így sokkal pontosabb képet kapunk a valós kockázatról, ugyanis ezzel figyelembe vesszük a tervezett mozgástól való kisebb eltérés esélyét is.

A szimuláció során több számított értéket kísérünk folyamatos figyelemmel, melyeknek szélsőértékei vagy kumulált metrikái fontos leíró szerepet töltenek be az adott manőver kockázatára nézve. Ezeket a jellemzőket az alábbiakban vesszük sorra, majd a végén bemutatjuk, hogy hogyan vonjuk ezeket össze egyetlen a manővert jellemző kockázati értéké.

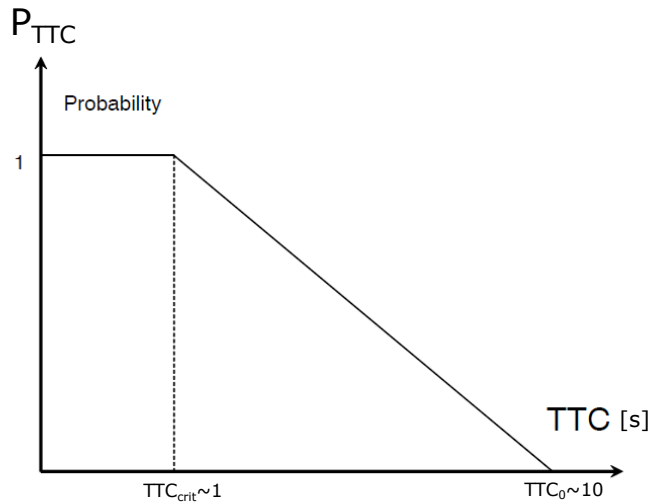
A kockázat képletében (3.12) szereplő  $P$  összetevő megfogásához a *HAVEit - Highly Automated Vehicles for Intelligent Transport* projektjének egyik publikációja[37] adta a legfőbb lökést. Ez a dokumentum két alapvetően különböző jellegű kockázati faktort vizsgál, az egyik a *Time to collision - TTC*, a másik pedig az *Inter vehicular time - TIV*. A TTC és a TIV több kutatásban és publikációban is feltűnik[38][39], gyakorlatilag minden a témával foglalkozó munka foglalkozik vele.

A publikációk alapján a TTC az az időhossz, amelyet követően az ütközés bekövetkezik, ha az objektumok jelenlegi sebességükkel haladnak tovább trajektóriájukon. A TTC tehát azt a típusú kockázatos helyzetet tudja észrevenni, amikor a két jármű relatív sebessége viszonylag nagy és ha nem avatkozik be senki, akkor ütközés következik be. A TTC eredeti formuláját a

$$TTC = \frac{D}{V - V_p} \quad (3.13)$$

képlet adja meg, ahol  $D$  a relatív távolság a két jármű között,  $V$  a saját,  $V_p$  pedig a peer abszolút sebessége. A kiszámított TTC időtartam alapján a 3.7. ábra grafikonja segítségével szokás meghatározni egy ütközés bekövetkezésének valószínűségét. Az görbe alakját és a töréspontok helyét több projekt, mint az ARCOS ([www.arcos2004.fr](http://www.arcos2004.fr)) vagy a PREVENT ([www.prevent-ip.org](http://www.prevent-ip.org)) is vizsgálta, az általuk megállapított értékeket használjuk mi is az algoritmusunkban.

Létezik egy másik típusú veszélyforrás is, amelyet a TTC segítségével nem tudunk észrevenni. Az olyan szituációkra gondolunk itt, amikor a két objektum nagy, de azonos sebességgel követi egymást közelről. Ilyen esetben az elöl haladó hirtelen fékezése könnyen balesetet okozhat, a TTC mégis végtelenre adódik, hiszen ha nem történik fékezés, akkor sosem ütköznek a járművek. Az ilyen helyzetek észrevételére alkalmas a TIV, amelyet követési

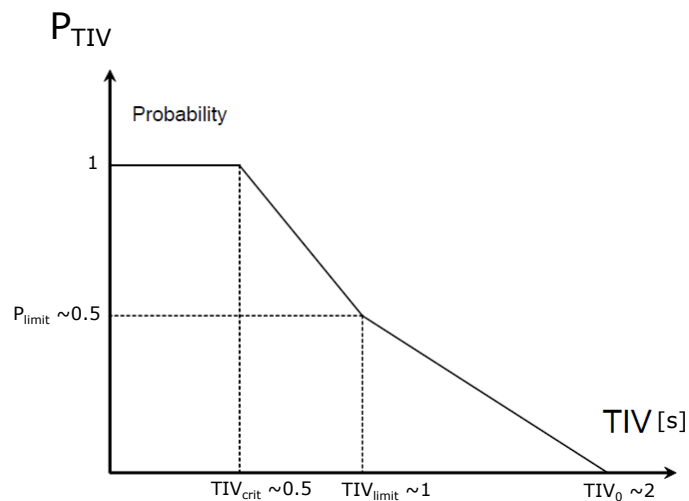


3.7. ábra. TTC görbe

időként értelmezhetünk, számítása a

$$TIV = \frac{D}{V_{rear}} \quad (3.14)$$

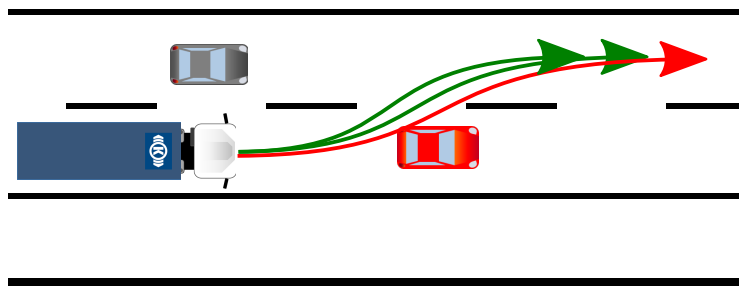
képlet segítségével történik, ahol  $V_{rear}$  a hátul lévő jármű sebessége,  $D$  pedig objektumok közti longitudinális távolság abszolút értéke. Az ebből következtetett ütközés valószínűségét a 3.8. ábrán látható függvény segítségével határozhatjuk meg, melynek alakja a TTC grafikonjához hasonló módon került megalkotásra.



3.8. ábra. TIV görbe

Fontos felismerni hogy a bemutatott TTC és TIV a kiindulási helyzetet képes csak értékelni, az általunk használt különböző manőverek végigkövetéséhez és vizsgálatához ezek kiegészítése volt szükséges számunkra.

A TTC esetében viszonylag könnyebb helyzetünk van, mivel a járműveket a szimuláció végiglépteti a kiadódó trajektóriájukon, így egyszerűen csak azt kell figyelniük, hogy mikor érnek egymásba a határoló téglalapok, elmentjük ezt az időpontot és ezt használjuk az ütközési valószínűség kiszámításához. A helyzetet kissé árnyalja, hogy az ego jármű kilenc kissé eltérő útvonalat is befut, ahogy azt korábban említettük. Ennek akkor tűnik elő a jelentősége, ha egy olyan szituációt és manővert képzelünk el, mint amilyen a 3.9. ábrán is látható, ahol az ego jármű épphogy csak elkerülné vagy épphogy csak nekiütközne a peer objektum sarkának.



**3.9. ábra.** Több közeli trajektória használatának használata

Mindkét eset szélsőséges, de észrevételük valamint megkülönböztetésük egyik oldalról kritikus, a másiktól hasznos extra információval szolgálhat. Kritikus azt észrevenni, amikor épphogy csak elkerüljük a balesetet a szimuláció szerint, hiszen ilyenkor a TTC végtelenre adódna, ezzel nem figyelmeztetve semmiféle veszélyre, holott bármiféle nem modellezett tényező eltérő trajektóriát eredményezhet a valóságban, ez pedig könnyedén balesetet okozhat ilyen szituációban. A másik irányból tekintve, amikor az egyetlen trajektória éppen találkozna a peer jármű sarkával, hasznos információ lehet egy vészhelyzet elkerülésekor, ha felismerjük, hogy az adott manővert végrehajtva van némi esély az ütközés elkerülésére. Sőt egyfajta derivált számítását is lehetővé teszi ez a módszer az adott manőverpontban, ezzel lehetőséget biztosít az optimális manőverpont keresésére, ezzel jelen dolgozatunk ugyan nem foglalkozik, de a későbbi fejlesztések szempontjából érdemesnek tartjuk megemlíteni. Ezen megfontolások alapján a kilenc trajektória mindegyike mentén rögzítésre kerül a TTC értéke, melyek közül végül a legkisebb kerül kiválasztásra. Ezen túl a szimuláció során alakuló legkisebb euklideszi távolságot is elmentjük minden trajektória mentén, majd ezeket a szimuláció végén kiátlagolva kapjuk a  $\overline{d_{min}}$  manővert jellemző értéket. Ezt a jellemzőt a

$$f(\overline{d_{min}}) = e^{-\overline{d_{min}}} \quad (3.15)$$

képlettel megadott függvény segítségével alakítjuk olyan formára, amivel a megkapott TTC-ből a 3.7. ábra grafikonja alapján számított  $P_{TTC}$  értékét módosíthatjuk a kívánt irányba. Az alkalmazott  $e^{-x}$  lecsengő függvénnyel tehát pont azt érzük el ami a célunk volt, hogy a nem nulla minimális átlagtávolságú trajektóriákkal rendelkező manőverpontokhoz (tehát ahol van olyan trajektória ami elkerüli az ütközést) kiszámolt TTC alapú kockázat csökkenni fog némileg, míg az eredetileg kockázatmentesnek adódó, de csak éppen ütközést

elkerülő manőverek kockázata drasztikusan megnő.

A TTC-ből származó kockázati komponens kiszámításához egyetlen további lépés szükséges még, fel kell skáláznunk azt a 3.2.1 szakaszban definiált kockázati értékek teljesítéséhez. Ezek alapján a kockázat TTC alapú komponensét a

$$R_{TTC} = r_{TTC,max} \cdot P_{TTC} \cdot f(\overline{d_{min}}) \quad (3.16)$$

képlet segítségével számoljuk ki, ahol  $r_{TTC,max}$  skalár paramétere a kockázatszámításnak.

A következőkben tekintsük át a TIV-ből származó kockázati komponens számításához használt megoldásainkat is. Célunk a manőver teljes hosszának feltérképezése TIV szempontjából, ez azt jelenti, hogy a TTC görbe lecsengéséig tartó időn belüli minden pillanat TIV értéke legyen hatással a számított kockázatra. Ezen gondolat számos forgalmi szituáció kockázat szinten való megkülönböztetését és megfelelő kezelését teheti lehetővé. Olyan esetekre gondolunk, mint például amit egy minket megelőző majd elénk bevágó peer jármű jelentene, ahol a hirtelen megemelkedő kockázat várhatóan magától csökkenni fog az idővel, ahogy az objektum távolodik. Ilyen esetben nem szükséges komoly beavatkozás, amit egyébként a bevágás pillanatában számolt egyedülálló TIV indokolna, elegendő ha elveszünk a gázt és kivárunk. Képzeljünk el továbbá egy az úton átkelő vadat, amely a szimuláció során épphogy átjut az ego jármű előtt, ebben az esetben TTC alapú kockázat egyáltalán nem jelenik meg, TIV alapú pedig egészen addig nincs amíg az állat be nem érkezik elénk, ez azonban nyilvánvalóan nem megfelelő kezelése a problémának.

Megoldásunk alapgondolata tehát, hogy a szimuláció minden pillanatában újraszámoljuk a TIV értéket, majd ezek súlyozott átlagát tekintjük az adott manővert és peert jellemző TIV értéként és erre alkalmazzuk a 3.8. ábra grafikonját. A súly két tagból áll össze, melyek segítségével két különböző fontos ötletet szeretnénk a kockázatok számértékeiben is megjeleníteni:

- Az időben a jelentől minél előrébb jár a szimuláció, annál kisebb súlya legyen az ott számolt TIV értéknek. Ennek oka, hogy minél távolabb vagyunk az ismertnek feltételezett jelentől, valószínűleg annál távolabb vagyunk a szimulációban a majd valóban bekövetkező valóságtól is, valamint annál több időnk van még beavatkozni egy veszélyes helyzet elkerülésére. Ezt a megfontolást a súly első tagjával vesszük figyelembe, amelyet a

$$w_{TIV,1}(t) = \frac{1}{1 + e^{t-3}} \quad (3.17)$$

szigmoid jellegű függvénnyel vesszük figyelembe, melynek alakját a 3.10. ábra a) grafikonján figyelhetjük meg.

- A másik fontos szempont, hogy a súlyozás emelje ki a veszélyesebb szituációt akkor is, ha az csak nagyon rövid ideig áll fenn, tehát az ne vesszen el teljesen egy azt követő vagy megelőző hosszan tartó TIV szempontból kockázatmentes időszak hatására.

Ennek okán a súlyfüggvénynek van egy másik, magától a számított TIV-től függő része is, ami a kisebb TIV értékeket nagyobb súllyal veszi számításba, ezt a tagot a

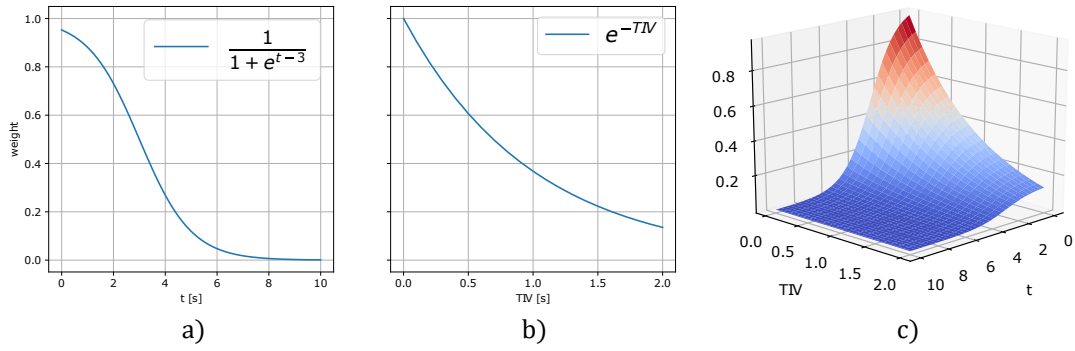
$$w_{TIV,2}(TIV) = e^{-TIV} \quad (3.18)$$

formában használjuk, a függvény formáját a 3.10. ábra b) grafikonja szemlélteti.

A teljes súlytényező a két tag szorzatából adódik a

$$w_{TIV}(t, TIV) = w_{TIV,1}(t) \cdot w_{TIV,2}(TIV) \quad (3.19)$$

képlet szerint, a függvény egy súlyfelületet definiál, amelyet a 3.10. ábra c) grafikonja segít megérteni.



**3.10. ábra.** *TIV súlyozásánál használt függvények, a) időfüggő tag, b) TIV függő tag, c) tagok együttes hatása*

A TIV lépésenkénti újraszámításánál még a súlyozás előtt két további módosító lépést hajtunk végre. Egyrészt amennyiben a TIV számítására használt (3.14) képlet eredménye nagyobb a 3.8. ábra grafikonján látható  $TIV_0$  értéknél, akkor azt  $TIV_0$ -val helyettesítjük, mivel kockázat szempontjából ezek azonos jelentéssel bírnak, a nagyon nagy TIV értékek viszont az alkalmazott súlyozás ellenére is képesek lennének túlzottan elhúzni a súlyozott átlagot. A másik fontos lépésben el kell dönteni, hogy egyáltalán releváns-e az adott peer jármű TIV szempontjából az vizsgált manőverre nézve az adott pillanatban, hiszen például egy tőlünk balra lévő sávban közlekedő jármű nem releváns a jobbra sávváltó manővereink számára, de releváns a balra sávváltókat tekintve. Előfordulhat továbbá, hogy a peer jármű a szimulált manővernek nem minden pillanatában releváns számunkra, ez jellemzően akkor fordulhat elő, ha a peer sávot vált a szimulált időtartam alatt. Amennyiben a peer jármű nem releváns számunkra, akkor a hozzá számolt TIV értékét szintén  $TIV_0$ -val helyettesítjük. Egy peer járművet relevánsnak tekintünk, amennyiben az laterális átfedésben van az ego járművel, továbbá akkor is ha átfedésben van a saját vagy a manőverünk célsávjával, valamint ha a peer jármű becsült manővere a saját sávkba vagy a manőverünk célsávjába vezet. Az így módosított TIV értékek súlyozott átlagát ekkor a

$$\overline{TIV} = \frac{\sum_{t=0}^{10} w_{TIV}(t, TIV) \cdot TIV(t)}{\sum_{t=0}^{10} w_{TIV}} \quad (3.20)$$

képlet segítségével számítjuk ki, ahol  $t$  értéke nem egységnyivel növekszik a szummázás során, hanem a szimuláció lépésközével. Az így megkapott manővert jellemző  $\overline{TIV}$  értékből a 3.8. ábra grafikonjának segítségével kapjuk meg a  $P_{TIV}$  értékét, melyet a TTC-nél látottakhoz hasonlóan szintén szükséges felskáláznunk, ezt a lépést a

$$R_{TIV} = r_{TIV,max} \cdot P_{TIV} \quad (3.21)$$

képlet szerint végezzük el. Az  $r_{TTC,max}$  és az  $r_{TIV,max}$  értékének meghatározásához érdemes figyelembe venni, hogy alacsony TTC-ből mindenképpen baleset következik, míg az alacsony TIV szintén veszélyes helyzet, de önmagában nem okoz balesetet, így az arányokat tekintve a TTC alapú kockázat súlyosabbnak tekinthető, mint a TIV alapú. A számításoknál használt konkrét paraméterek értékét a 3.3.2 rész 3.3. táblázatában tekinthetjük majd végig.

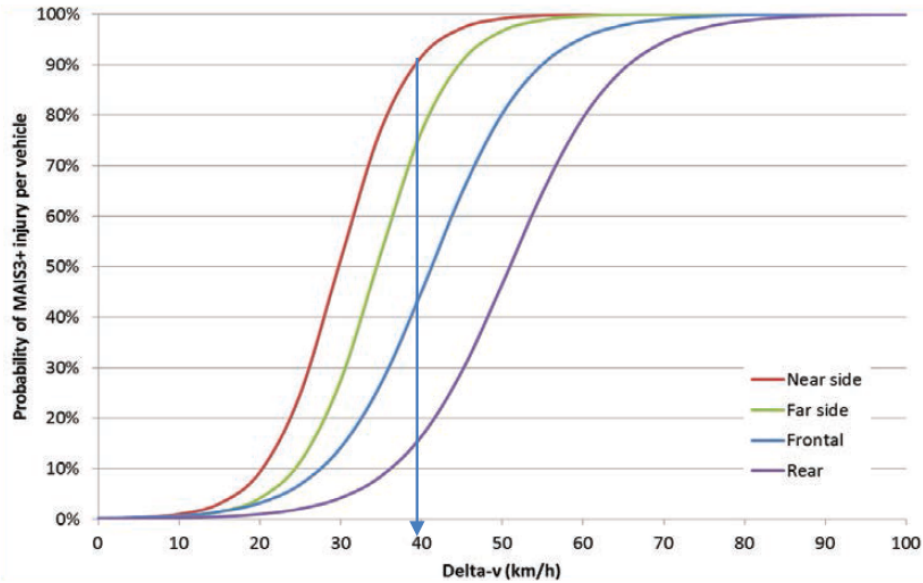
Minden adott a teljes kockázat  $P$  komponensének meghatározásához, melyet a

$$P = \sqrt{R_{TIV}^2 + R_{TTC}^2} \quad (3.22)$$

képlet segítségével teszünk meg, amely mintegy külön dimenziók mentén elhelyezkedő kockázatoknak tekinti a TIV és a TTC alapú komponenseket.

Ezt követően vizsgáljuk meg a kockázat  $S$  komponensét is amely a potenciális ütközés súlyosságát hivatott jellemezni. Gyakori megoldás erre a feladatra az *Equivalent Energetic Speed (EES)* alkalmazása[40], amely a jármű ütközés előtti és az azt követő sebességéből határozható meg, ezzel az ütközés súlyosságát a deformációs energia mértékével teszi arányossá. Ennek kiszámításához azonban már a legalapvetőbb adataink sem állnak rendelkezésre, mint például a peer jármű tömege, így egy másik módszer, a delta-v alapján hoztuk létre az algoritmusunkat, melyet Hobbs és Mills alkalmasnak talált[41] az emberi sérülés súlyosságának becslésére, azóta ezt a módszert számos további kutatás alkalmazta[42]. Ennek használatához szükséges meghatároznunk az ütközés során bekövetkező sebességváltozás mértékét, majd ezt felhasználva a széles körben használt[43]  $P_{MAIS3+}$  görbék segítségével határozhatjuk meg a súlyosság értékét. Az *Abbreviated Injury Scale (AIS)* egy 1-től 6-ig tartó skála[44], amelyen a sérülések súlyosságát szokás jelölni, a  $P_{MAIS3+}$  egy az AIS skálán legalább 3-as szintű sérülés valószínűségét adja meg, ami egy klinikailag súlyos sérülés esélyét jelöli.

Megfigyeléseink hiányossága miatt csupán egy egyszerű modellt tudunk alkalmazni a sebesség megváltozására. Jelenleg egy viszonylag szerencsétlen esetet veszünk számításba, azt amikor az ütközésben résztvevő egyik objektum tömege elhanyagolható a másikéhoz képest. Ekkor a nehezebb objektum sebessége nem változik meg az ütközés folytán, a könnyebb objektum viszont a teljes ütközés pillanatában adott objektumok közti sebességkülönbséget elszenvedti változásként. Ennek okán a szimuláció folyamán bekövetkező ütközéskor rögzítésre kerül az objektumok sebességkülönbségének abszolút értéke mind a kilenc trajektória



**3.11. ábra.** Súlyos sérülés valószínűsége az ütközésben elszenvedett sebességváltozás függvényében, különböző típusú ütközések esetén, Bahouth (2014) mérései alapján[45]

mentén, majd ezek közül a maximális kerül kiválasztásra, amelyet felhasználva a

$$S = \frac{1}{1 + e^{-0.5(\Delta v - 14)}} \quad (3.23)$$

képlet segítségével meghatározzuk az ütközés súlyosságát, melynek alakjával a 3.11. ábra kék görbét, ezzel egy frontális ütközés görbét közelítjük.

A későbbiekben olyan irányban szeretnénk bővíteni ezt a megoldást, hogy a különböző objektum típusokhoz más-más  $P_{MAIS3+}$  görbét és szintén eltérő sebességváltozás analízist alkalmazzunk, hiszen teljesen más sérülési esélyek jelentkeznek egy motorossal vagy egy másik kamionnal való ütközés esetén.

Ezzel a kockázat számításához minden komponensünk számítása ismert a (3.12) képletből.

## Risk Fusion

A számításokat tehát minden manőverre és azon belül is minden peerre el kell végezni, ezt követően a kockázat fúzió feladata a peerenkhez kiszámított risk értékek egybe olvasztása manőverenként. Több megoldás is elképzelhető, mint például a maximum kockázatú peer jármű kiválasztása vagy az egyszerű összegzés. Nehéz megmondani hogy milyen valós kapcsolat áll fenn a peerenként számolt kockázatértékek közt, azaz, hogy az egyik objektummal bekövetkező baleset hogyan befolyásolja a többivel kiszámított mindenki mástól független értékeket. Ez az algoritmus az alábbi képlet szerint összegzi a kockázatokat:

$$R_{fused} = \sqrt[16]{\sum_{n=1}^{16} R_{peer,n}^{16}} \quad (3.24)$$

A megadott képlettel kapcsolatban az alábbi fontos megállapításokat tehetjük, ezek miatt gondoljuk úgy, hogy lehetőségeinkhez mérten megfelelően tudja jellemezni a valós kockázatot ez a megoldás:

- a tizenhat peerhez tartozó tizenhat kockázati komponens összegzése a  $p$ -norma képlete alapján történik ahol  $p = 16$
- ha az egyik peerhez tartozó kockázat markánsan nagyobb a többinél, akkor gyakorlatilag ez fogja meghatározni a fuzionált eredményt is, a többi kockázati komponens csak kis mértékben fogja ezt tovább emelni,
- ha nincs kiugró, de van több hasonló kockázati értékkel bíró peer jármű, akkor ezek fúziójával egy bármelyiknél számottevően nagyobb kockázat adódhat végül.

### Conditional Risk Modifier

A kockázatok fuzionálását követően pár egyszerűbb vizsgálatot eszközöl az algoritmus és ezek alapján módosíthat egyes kockázati értékeken.

- A rendszer figyeli a saját és a két szomszédos sáv típusának megváltozását és annak távolságát is, ezeket felhasználva megnövekedett kockázattal jellemzi a megszűnő sávban való tartózkodást valamint az ilyen típusú sávba tartó manővereket is. A kockázat ezen összetevője a sáv végéhez közeledve a sebesség arányával nő, gyakorlatilag úgy kezelendő a sáv vége, mintha egy álló objektum lenne ott.
- Minden sáv váltó manőver kockázata legalább  $R_{lane\_change}$  értéket kell kapjon (ez is az címkéző egyik beállítható paramétere), ezzel preferálva a sáv tartó viselkedést. Ez azt jelenti, hogy a későbbi manővert elemző kalkulációk során erős "érvet" kell találjon a rendszer a sáv váltásra, tehát csak akkor fog így tenni, ha valamilyen okból ez komoly előnyt jelent számunkra.

### 3.3.2. Implementált számítás

A feladat megvalósításához Python3.6 környezet került alkalmazásra, a választása mellett a nyelv egyeduralmódóvá válása a legerősebb érv a data science területén. A címkéző implementálásához az alábbi kiegészítő csomagok segítségével lett kihasználva, a script futtatásához is ezekre és ezek függőségeire van szükségünk:

**scipy 1.1.0+** A szükséges fizikai konstansokat innen használja a címkéző algoritmus.

**numpy 1.14.5+** C alapokra épített tudományos számításokhoz elengedhetetlen csomag, az N-dimenziós mátrix implementációja kerül kihasználásra a többi package-n keresztül.



**xArray 0.10.8+** A numpy által definiált adatstruktúrát terjeszti ki úgy, hogy a dimenzióknak és ezek koordinátáinak saját magunk adhatunk nevet, így az indexelések sokkal átláthatóbbá válnak, ezzel saját magunk és mások számára is könnyebben olvasható kódot készíthetünk.

**h5netcdf** A HDF5 és a netCDF4 ki és bemeneti fájlok kezeléséhez. A felcímkézendő, a címkézett valamint a debugoláshoz szükséges adatokat ebben a formátumban volt célszerű használni, az xArray modul is ezt támogatja alapértelmezés szerint.

A számításban több beállítható paraméter is található, melyek kisebb-nagyobb mértékben befolyásolják az egyes manőverekhez számított konkrét kockázati értékeket, ezek szerepét már a korábbiakban részleteztük, itt most csak a felsorolásukat és az általunk használt alapértelmezett SI-ben megadott értéküket tekinthetjük át a 3.3. táblázatban. Ezeken felül természetesen az alkalmazott függvényalakok és azok együttthatói is nagyban hozzájárulnak a számítás végeredményéhez, így lényegében ezek is a címkéző algoritmus paramétereinek tekinthetőek.

**3.3. táblázat.** *Címkéző algoritmus paramétereit.*

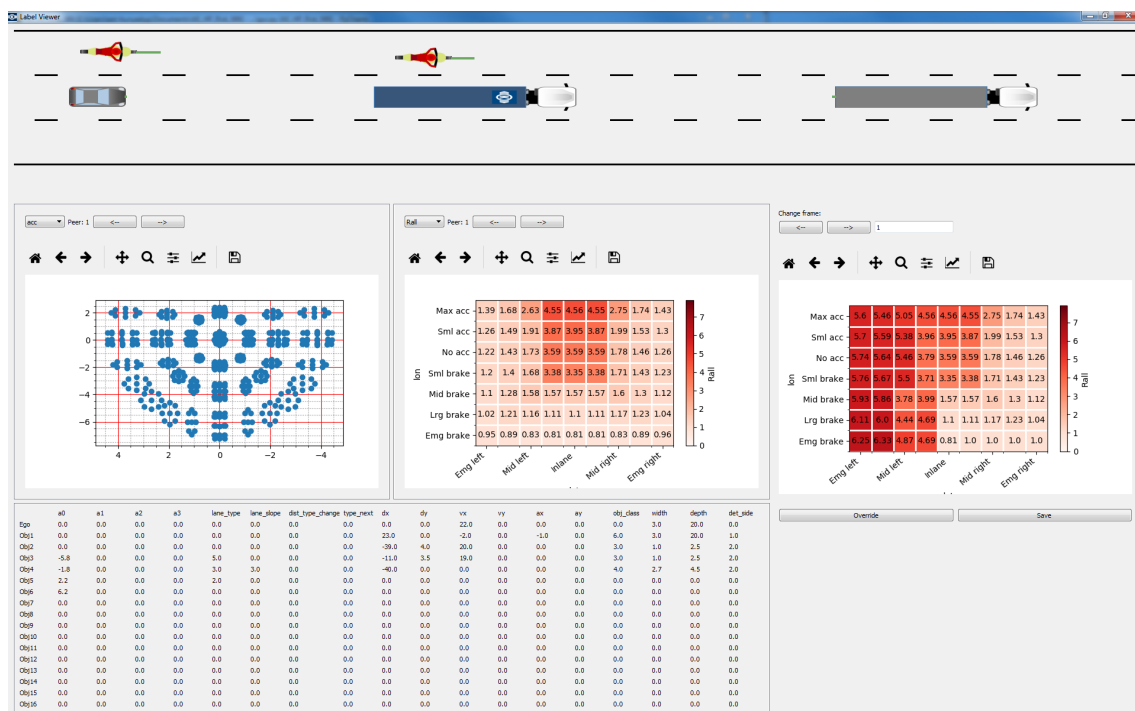
Megnevezés	Leírás	Alapérték
SIM_STEP_TIME	Szimulációs lépésköz.	0,01
EGO_WIDTH	Ego jármű szélessége.	3
EGO_DEPTH	Ego jármű hossza.	18
LEFT_RIGHT_LIM	A hátulról érkező objektumok balra vagy jobbra elhúzását szeparáló laterális sebesség.	-0,3
W_POS	Peer járművek mozgástípusát meghatározó függvény együttthatója.	0,5
W_SPEED	Peer járművek mozgástípusát meghatározó függvény együttthatója.	1,3
W_ACC	Peer járművek mozgástípusát meghatározó függvény együttthatója.	0,2
AY_PEER	Sávváltó peer járművek oldalgyorsulása a manőver végrehajtásához.	1
R_TIV_MAX	$P_{TIV}$ -ből számolt maximális kockázat értéke.	3,5
R_TTC_MAX	$P_{TTC}$ -ből számolt maximális kockázat értéke.	4
ALPHA	A kockázat súlyosságának súlyozásához használt paraméter.	0,25
R_LANE_CHANGE	Sávváltás minimális kockázata.	1

Mivel a számítások szituációnként, azon belül is peerenként és manőverenként (ráadásul a kilenc trajektóriánként is) egymástól függetlenül végrehajthatóak, így célszerűnek tűnt ezek elvégzését párhuzamosítani és így gyorsítani a tanító adatok címkézését. Ennek megvalósítására gyakorlatilag minden számítást öt dimenziós tenzorokon végeztünk, melyek az alábbi dimenziókkal rendelkeznek:

- **frame** Ezen tengely mentén sorakoznak fel a különböző címkézendő forgalmi szituációk, amelyeket egyszerre szeretnénk címkézni.
- **lateral** A manővertér laterális tengelye, e mentén sorakoznak fel a különböző oldalirányú dinamikával rendelkező manőverpontok.
- **longitudinal** A manővertér longitudinális tengelye, e mentén sorakoznak fel a különböző hosszirányú dinamikával rendelkező manőverpontok.
- **trajectory** Az egyes manőverpontokhoz tartozó kilenc közeli dinamikájú pontok ezen tengely mentén helyezkednek el a tenzorban.
- **peer** Minden szituációhoz tizenhat peer tartozik, ezeket láthatjuk ezen a dimenzión.

### 3.3.3. Grafikus megjelenítő

A fejlesztés során igen hasznosnak bizonyult egy grafikus kockázat és szituációmegjelenítő használata, mivel mindig felmerültek újabb hiányosságok, ezeknek a felderítésére volt nagyon alkalmas az alábbi GUI.



3.12. ábra. Fejlesztéshez használt GUI felület

A képen látható ablakban a már lefutott esetek kiértékelését vizsgálhatjuk meg, egyrészt a képernyő felső részén kirajzolódik a forgalmi helyzet felülnézete, amelyet a legalul lévő számokból jelenítünk meg. A középső sávban pedig a különböző peerekhez és manőverekhez tartozó kockázatok és egyéb manőver jellemzők kaptak szemléletes alakot, itt legördülő menüből választhatjuk ki a vizsgálni kívánt kockázati komponenst objektumonként, valamint a súrlódási ellipszisre torzított pontokat és az őket körülvevő 9 trajektória gyorsulásait is megvizsgálhatjuk.

### 3.3.4. Adatgyűjtés

Egy neurális háló tanításához nagy mennyiségű és "jó minőségű" tanítóadatra van szükségünk, amit ehhez a problémához nem triviális feladat beszerezni, így a munka egy komoly részét ez a feladat töltötte ki.

A beszerzett adatok több forrásból érkeztek, így ezek megfelelő feldolgozása időigényes munka volt, bizonyos jelek nem álltak rendelkezésre minden mérésben, így ezeket ilyenkor megfelelő algoritmussal pótolni volt szükséges.

- Korábbi valós rögzített LSA mérések.
- Adatok kimentése a dSPACE alapú szimulációból.
- Adatok kimentése a saját fejlesztésű szimulációból.
- Adatok generálása valószínűségi eloszlások és szakértői rendszer alapján.

A felsorolt opciók közül jelenleg az első két pontból származó adatokkal dolgoztunk, a saját szimulátor már elkészült és vannak ebből származó adataink is, ezek feldolgozására azonban még nem került sor. Az utolsó pont egy távolabbi tervet jelent, ugyanakkor hosszútávon ez a módszer fogja tudni legjobban megközelíteni a valós mérések jellegét, valamint a véletlenszerűségen is alapuló módszerrel a teljes állapotteret le fogjuk tudni fedni mintákkal.

### 3.4. Hálótervezés és tanítás

A neurális háló tanításához a Google által fejlesztett `Tensorflow` backenden futó `Keras` interfészt használtuk szintén `python` környezetben. Dolgozatunk elkészültekor már sikerült lefuttatni az első tanításokat, ezek eredményét és az ezekből a továbbiakra levont tanulságokat kívánjuk itt bemutatni. A neurális hálókról egy rövid összefoglalót a 4.2.2 fejezet részben olvashatunk.

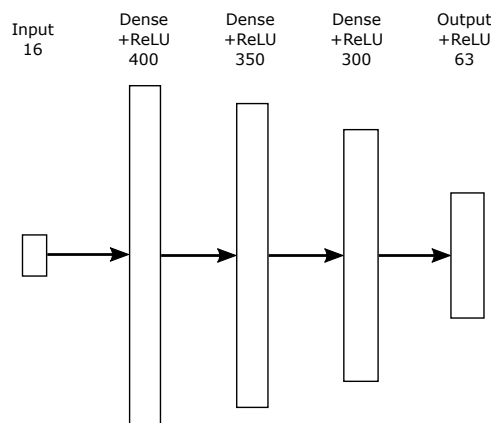
Az adatok előkészítése fontos lépés a sikeres tanítás eléréséhez, így komoly figyelmet fordítottunk a megfelelő bemenetek kiválasztására, azok standardizálására, valamint a kategorikus változók helyes kezelésére. Ezek alapján tizenhat bemenetet definiáltunk a neurális hálónak, ezeket a 3.4. táblázat tartalmazza, emellett a  $7 \times 9 = 63$  kimenet adottnak tekinthető, hiszen ennyi manőverpontunk van.

A háló struktúráját tekintve a hagyományos *Multilayer Perceptron* (*MLP*) választottuk a kiindulásként, és mint azt az eredményeken látni fogjuk, megoldható vele a feladat, hiszen a kimenet elkezdett konvergálni. A kezdeti néhány kipróbált háló közül a három rejtett réteggel és 400-350-300 neuronnal rendelkező rendszer bizonyult a legjobbnak, ennek vizualizációját a 3.13. ábrán láthatjuk. Minden réteget követően a Rectified Linear Unit (ReLU)

### 3.4. táblázat. Neurális háló bemenetei

Megnevezés	Leírás
a0	Az ego járművet közrezáró két sáv vonal $a_0$ együttthatói, ez 2 bemenetet jelent.
a2	Az ego járművet közrezáró két sáv vonal $a_2$ együttthatói közül a nagyobb abszolút értékű.
ego_vx	Az ego jármű longitudinális abszolút sebessége.
peer_x	A peer jármű longitudinális pozíciója az ego járműhöz képest.
peer_y	A peer jármű laterális pozíciója az ego járműhöz képest.
peer_vx	A peer jármű longitudinális abszolút sebessége.
peer_vy	A peer jármű laterális abszolút sebessége.
peer_ax	A peer jármű longitudinális abszolút gyorsulása.
peer_ay	A peer jármű laterális abszolút gyorsulása.
peer_width	A peer jármű szélessége.
peer_depth	A peer jármű hosszúsága.
peer_side	A peer jármű észlelt oldala, mivel ez egy négy különböző értéket felvehető kategorikus jellemző, így négy bemenetté alakul át.

aktivációt használtuk, amely komoly áttörést jelent a neurális hálók tanításában, hiszen megoldást nyújt a régóta jelen lévő elenyésző gradiens (vanishing gradient) problémára[46], ennek köszönhetően előszeretettel alkalmazzák a rejtett rétegekben. Esetünkben a kimeneten is indokolt volt az alkalmazása, hiszen jól követi a lehetséges kimeneti kockázatok értékét, a negatív értékeket levágja, pozitív irányban viszont tetszőleges mértékben engedi növekedni.



3.13. ábra. Bemutatott neurális háló felépítése

A metrikák alapján elért pontosság kiemelkedőnek mondható, a teszt adatokon mért átlagos négyzetes hiba 0.019 -re adódott, ez azt jelenti, hogy átlagosan  $\sqrt{0.019} = 0.14$  -es eltéréssel

eltalálta a modell a címkén szereplő kockázatot. A helyzetet valamelyest rontja, hogy a címkék közt sok 0 értékű található, s ezeket rendre el is találta a neurális háló, így a többi, nullától eltérő érték esetén valamivel nagyobb átlagos eltérésre lehet számítani.

A hiperparaméterek hangolásával még nem tudtunk hosszabb időt tölteni, így ezen a téren még mindenképp van lehetőség a javulásra, ugyanakkor a kezdeti eredményeket nagyon biztatónak értékeljük.

### 3.5. Fejlesztési lehetőségek

A projektünk jelenleg is folyamatos aktív fejlesztés alatt áll, így számos lépés elvégzése szükséges még. A döntéshozó és az ezt támogató algoritmusok, mint például ennek a kockázatszámításnak a tesztelésére létrehoztunk egy saját szimulációs környezetet, amelyről a 2.5 szakaszban már olvashattunk. Ennek nagy előnye, hogy a leegyszerűsített valóságból könnyedén elérjük a környezetfelismerés szimulált interfészét, valamint a szintén leegyszerűsített járműdinamika lévén végre is tudja hajtani a köztes rétegbe helyezett döntési algoritmusaink eredményeit. Fontos következő lépés tehát, hogy a kockázatszámító megoldásunkat integráljuk a szimulátorunkkal és ezen keresztül keressük meg a potenciális gyenge pontokat, melyeket megismerve célzottan tudunk majd megfelelőbb tanító adatokat generálni és a címkéző algoritmust valamint a hálóstruktúrát a tanítás paramétereivel együtt tovább fejleszteni.

Érdekes és egyben hasznos lehetőséget nyújt továbbá a címkéző algoritmus paramétereinek és függvényalakjainak vizsgálata, hogy ezek alkalmazása mennyire pontosan írja le a valóságot, illetve ezek finomhangolásával vagy drasztikus megváltoztatásával milyen eredmények érhetőek el.

Az egyszerű szimulátoron nyújtott megfelelő eredmények elérésekor megkezdődhetnek majd a valós járműn végzett tesztelések, továbbá célunk egy komplexebb szimulációs környezetben való kiértékelés is. Az összetettebb szimuláció elérésére jelenleg két módot is látunk, az egyik az egyszerűbb szimulátorunk tovább fejlesztése, a másik a már szintén említett dSPACE alapú szimulációs rendszer alkalmazása. Mindkét megoldásnak vannak előnyei és hátrányai is, mi jelenleg a saját szimulációnk részletezését látjuk hosszútávon célravezetőnek, a megfelelő döntés meghozásához azonban a lehetőségek részletes összevetésére lesz szükségünk.

## 4. fejezet

# Forgalmi szituációk generálása, elemzése

A 2. fejezetben ismertetett Situation Analysis (röviden SIT) rendszerkomponens feladata többrétű. A végrehajtható manőverek meghatározásán és súlyozásán túl további cél, hogy a viselkedéstervezést megtámogassa a környezetérzékelésből érkező adatoknál egy szinttel magasabb absztrakciós szintű információval.

Az LSA interfészen keresztül érkező adatok ugyanis csak a felismert forgalmi szereplőket tartalmazzák. Az összességükhöz kapcsolódó aggregált információk, illetve a bizonyos részhalmazuk által végrehajtott akciók, manőverek meghatározhatnak gyakori forgalmi szituációkat, amelyek detekcióját felhasználhatjuk a viselkedéstervező döntéshozási folyamatának befolyásolására.

Ez a lépés hozzájárul az önvezető funkció biztonságos működéséhez, hiszen az emberi vezetés során is alkalmazkodunk vezetési stílusunkkal a körülöttünk lévő szereplők helyzete, cselekedetei által adódó helyzetekhez, ezzel tartva magunkat a balesetmentes közlekedéshez, mint elsődleges célhoz.

### 4.1. Forgalmi szituációk osztályozása

A forgalmi szituációt definiálhatjuk egy adott pillanatban körülmények, feltételek, tények egy meghatározott kombinációjaként [47]. Másképp megfogalmazva közlekedési szabályok következményeként végrehajtott lépések, interakciók egy meghatározott halmaza [48].

A forgalmi helyzeteket többféle módon is osztályozhatjuk. Dolgozatunkban az autópályás közlekedéssel kapcsolatos szakirodalmi megközelítéseket foglaljuk össze a továbbiakban.

#### 4.1.1. Végrehajtott manőver szerinti felosztás

Desel és Hermann [48] a végrehajtható manővereket általános szemszögből gyűjtik össze. Csoportosítási módszerük alapja, hogy az önvezető funkció által végrehajtott művelet

tartalmaz-e másik közlekedési szereplőt. Eszerint különböztetnek meg akciókat és interakciókat. Osztályozásukban az akciók olyan műveletek, amelyek leírása nem igényel másik forgalmi résztvevőt. Néhány példa e manőverekre a sávváltás, leállítás, sáv követése, stb. Az interakciók közé sorolható a másik jármű megközelítése, követése, előzése, stb.

A TrafficNet [49] projekt hasonló megközelítéssel dolgozik, melynek célja, hogy az utóbbi évtizedben nagy mennyiségben készült forgalmi felvételek adatainak feldolgozását támogassa, csoportosítsa a bennük megtalálható szituációkat.

#### 4.1.2. Speciális helyzetek felismerése

Előfordulhatnak olyan speciális szituációk, amelyek egy level 3 rendszer esetén akár az irányítás sofőr általi átvételét is igényelhetik, így elengedhetetlen megbízhatóan felismerni őket a SIT modulnak. A [50] tanulmány felsorol néhány ide köthető példát:

- Forgalmi sávok funkciójának megváltoztatása: ideiglenes lezárások, korlátozások egyes közlekedési szereplők számára.
- Sávok ideiglenes átirányítása: annak az esete, amikor a sávnak a pozícióját megváltoztatják, tipikusan építési munkálatok során tapasztalhatunk ilyet.
- Megkülönböztető jelzéssel haladó járművek megjelenése.
- A jármű útvonalában megjelenő akadályok elkerülése.

A rendszerünkben található környezetérzékelés által biztosított absztrakt világkép tartalmaz támogatást a fent ismertetett helyzetek felismerésére.

#### 4.1.3. Forgalom sűrűségének osztályozása

Az autópálya egy adott szakaszára vonatkozólag az áthaladó forgalom osztályozása egy népszerű terület. A [51] tanulmányban az adott szituációt forgalmi sűrűség szerint három kategóriába sorolták: könnyű, közepes és erős forgalom. Az osztályozáshoz különböző tanulóalgoritmusok működését hasonlították össze, melyek a K legközelebbi szomszéd, naiv Bayes, SVM és az MLP (multilayer perceptron) voltak. Kutatásukban a neurális háló alkalmazásával érték el a legjobb eredményt 94,5%-os osztályozási pontossággal, de a többi algoritmus teljesítménye sem maradt el sokkal az előbbtől.

Az említett osztályozási módszer releváns abból a szempontból, hogy az emberi sofőrök is fokozatosan változtatják vezetési attitűdüket a forgalom erősödésével. Például üres autópályán sokkal gyakrabban váltunk sávot, hajtunk végre előzéseket, mint egy dugóközeli szituációban, ahol néhány autó elkerülése alig hoz stratégiai hasznot útvonaltervünk szempontjából, ráadásul kis követési távolságok esetén minden sávváltás további kockázatot jelent a résztvevőkre.

A viselkedéstervezésnek tehát hasznos bemenetet jelenthet egy, az előbbieken említett osztályozó modul kimenete. Kutatásunkban megvizsgáljuk, mely osztályozó algoritmusok biztosítják alkalmazási területünkön a legjobb eredményt.

#### 4.1.4. Osztályozás az érzékelt veszély mértéke szerint

Megközelíthetjük a forgalmi szituációk osztályozását a környezetünkben észlelt veszélyes, vagy előre nem várt manőverek száma és minősége alapján is, vagyis a forgalom hektikuságának mérésével.

A forgalmi helyzet kiszámíthatatlanságát jellemezhetik olyan mérőszámok, mint például sávonként az egyes járművek között mért követési távolságok szórása, illetve egy adott peer jármű kiugró gyorsítása/lassítása.

A [52] cikk szerzői algoritmusukkal a jármű sebessége, hosszirányú- és keresztirányú gyorsulása alapján, illetve a sofőr fiziológiai állapotának több szenzorral történő mérésével osztályozták az adott helyzetben valószínűleg kialakuló veszélyhelyzetet, amelyhez három szintet definiáltak:

- *Veszélyes közlekedési esemény*: amely vagy elkerülhetetlen ütközéssel jár, vagy a sofőrtől hirtelen, erőteljes beavatkozást igényel a baleset elkerüléséhez.
- *Kockázatos események* definiálhatóak az olyan helyzetek, amelyben a vezetőnek bár nem kell vészhelyzeti manővert végrehajtania, de kiemelt figyelmet kell tanúsítania az ütközés megelőzéséhez.
- *Biztonságos szituációk*: az előző helyzetekbe nem tartozó, olyan közlekedési események, amelyek nem befolyásolják a járművezető normál viselkedését.

A [53] cikkben hasonló megközelítést tapasztalhatunk, ahol a közlekedésben résztvevő manővereinek veszélyességét értékeli, például a hirtelen sávváltásokat és ún. zigzagging akciókhoz köthető gyorsulás- és sebességadatok meghatározó paramétereit elemzik.

#### 4.1.5. Érzékelés minőségét befolyásoló környezeti tényezők felismerése

További viselkedéstervezést befolyásoló körülmény a környezeti viszonyok és ebből fakadóan a környezetérzékelés minőségének romlása, amely méréséhez releváns adat például a látótávolság, illetve az ehhez képesti becsült fékút. Ezen információk meghatározásához pedig elengedhetetlen a szenzoraink által lefedett terület feldolgozása, továbbá minél több speciális időjárási körülmény és útviszony felismerése. A teljesség igénye nélkül néhány példa: rossz minőségű út (kátyúk), külső hőmérséklet, köd, esetleges jegesedések lehetősége. A felsorolt környezeti paraméterek méréséhez további szenzorok felszerelése és mérési módszerek kidolgozása szükséges, így dolgozatunkban eltekintünk a környezetérzékelés minősége alapján történő osztályozástól.



## 4.2. Konceptió

A szituáció osztályozó modul jelenlegi célja, hogy a jármű körüli forgalom sűrűségének lehető legpontosabb becslésével hasznos bemenettel szolgáljon a viselkedéstervező számára. Ennek elvégzéséhez több algoritmus működését, az általuk adott eredményt hasonlítjuk össze, majd a legmegfelelőbbnek ítélt működését optimalizáljuk.

A modul bemeneteként az LSA interfész adatai szolgálnak, amelyekből a következő jeleket használjuk fel:

- jármű állapotvektorából:
  - `vx` - longitudinális ego sebesség
  - `ax` - longitudinális ego gyorsulás
  - `depth` - ego járműszerelvény hossza
- Sáv információk:
  - `range` - elválasztó vonal detektálási hossza
  - `quality` - detektálás minősége
- Felhasznált bemenetek az érzékelt objektumok paraméterstruktúrájából:
  - `dx` - objektum relatív, hosszirányú távolsága
  - `vx` - objektum relatív, hosszirányú sebessége
  - `ax` - objektum relatív, hosszirányú gyorsulása
  - `width` - objektum szélessége
  - `depth` - objektum hossza
  - `det_side` - objektum detektált oldala
  - `lane_assign_1` - objektumhoz rendelt forgalmi sáv(ok)
  - `lane_assign_2`

Minden peer járműhöz az interfész két forgalmi sávot rendel (`lane_assign_1` és `lane_assign_2`), mindkettőhöz tartozik egy-egy sáv index, illetve az adott sávban tartózkodáshoz tartozó valószínűség.

A fent felsorolt adatokból minden sávra meghatározzuk a következő aggregált információkat:

- objektumok egységnyi hosszra jutó száma,
- objektumok átlagos sebessége,
- objektumok átlagos TIV paramétere,

- objektumok átlagos longitudinális gyorsulása.

A fent említett paraméterek közül az első meghatározásához összegezzük adott forgalmi sávban található objektumokat súlyozva a sávhoz tartozó valószínűségükkel, az összegnek pedig vesszük az érzékelési tartománnyal való hányadosát.

A TIV-t, vagyis a követési időt a 3.3.1. szakaszban már ismertetett módon számíthatjuk két egymást követő járműre, amelyet meghatározunk minden egymást követő járműpárosra, majd vesszük minden sávra a számtani közepüket.

E paramétereket tehát minden időpillanatban (LSA frame) kiszámítjuk, melyek algoritmusunk bemeneti paraméterterét szolgáltatják. A fluktuációk kiszűrése céljából alkalmazunk egy 10 másodperces ablakot, amelyben 100ms mintavétellel veszünk 100 LSA frame-t.

Az osztályozást minden forgalmi sávra egymástól függetlenül végezzük el, ahol a bemenetet tehát az utolsó 10 másodpercben vett 100 időpillanatban kiszámolt 4 paraméter jelenti.

Algoritmusunkkal a kimenetet 5 osztály egyikeként adjuk meg, finomítva a felbontást a [51] kutatásban találtakhoz képest. Az egyes szintekről leíró jelleggel a következőt állíthatjuk:

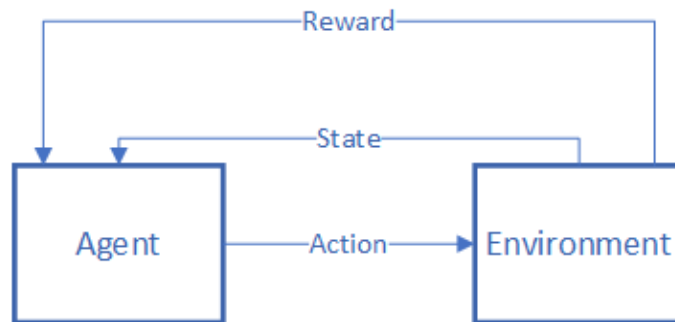
- **Empty**: forgalmi sűrűség legalacsonyabb szintje, gyakorlatilag az üres forgalmi sávot jelképezi.
- **Open flow**: szabadon haladó forgalom.
- **Mild congestion**: közepesen torlódó forgalom.
- **Heavy congestion**: erős torlódást jelző osztály.
- **"Stop and go" traffic jam**: forgalmi dugó teljes megállásokkal.

#### 4.2.1. Gépi tanulás

Az előbbiekben ismertetett problémára a gépi tanulás osztályozó algoritmusait alkalmazzuk, amelyek működését az alábbiakban röviden bemutatjuk. Az algoritmusokat a tanulás módja szerint jellemzően három kategóriába soroljuk, melyek a *felügyelt tanulás*, az *ellenőrizetlen tanulás* és a *megerősítéses tanulás*. A megerősítéses tanulás esetében a cél, hogy egy olyan intelligens rendszert fejlesszünk, ami javítja a teljesítményét a környezetével való interakciója során, hogy megtanulhasson egy (közel) optimális stratégiát. Ezt úgy éri el, hogy visszacsatolunk a környezetből egy ún. jutalom jelet, amely visszajelzéssel szolgál az algoritmusnak a korábban tett lépéseiről.

Klasszikus példa a tanulás e formájára a sakk motor működése. Az ágens akcióit lépések sorozata jelenti a sakktáblán, amely esetünkben a rendszer környezeteként szolgál. Visszacsatolásként pedig játék megnyerése, vagy elvesztése adható meg az algoritmus számára.

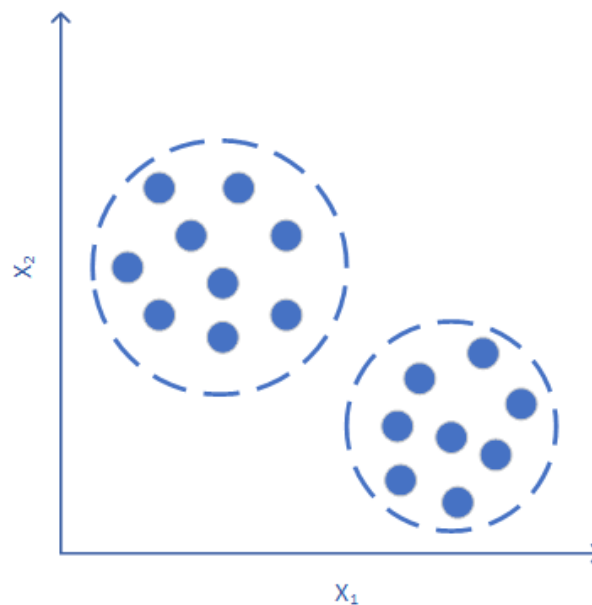
A megerősítéses tanulás folyamatát a 4.1. ábra szemlélteti.



4.1. ábra. A megerősítéses tanulás szereplői [54]

Az ellenőrizetlen tanulás során címkézetlen vagy ismeretlen struktúrájú adatokkal dolgozunk. A célunk, hogy a rendelkezésünkre álló adathalmazból értékes információt nyerjünk ki anélkül, hogy bármilyen visszacsatolást felhasználna.

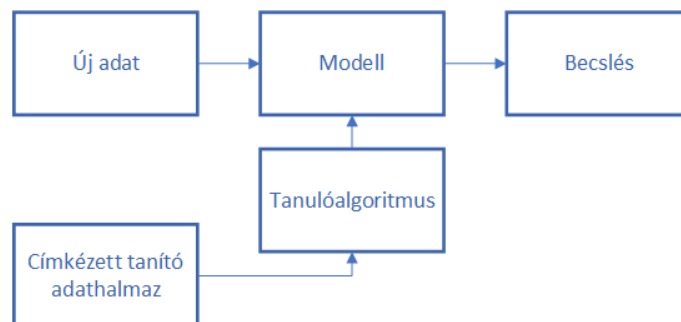
Klaszterezés során például a tanulóalgorithmus alcsoportokba rendezi az adatpontokat a paramétereik hasonlósága alapján. Ezt egy kétparaméteres adathalmaz esetében a 4.2. ábrához hasonlóan tudjuk szemléltetni, ahol címkézetlen adatokat az algoritmus két diszjunkt csoportba rendezi.



4.2. ábra. A klaszterezés kimenetének illusztrációja [54]

A felügyelt tanulás célja, hogy címkézett tréningadatok segítségével felépítsen egy olyan modellt, ami számunkra megfelelő pontossággal von le következtetéseket tanulása során még nem ismert adatokról. Az ellenőrizettség abból fakad, hogy a kimeneti jel (címkék) ismertek.

A modellt a tanulási folyamat során állítjuk fel, amellyel a predikciókat végezzük, és addig javítjuk, amíg a modellünk eléri a kívánt pontosságot a tanító adathalmazon. A folyamatot a 4.3. ábra szemlélteti.



**4.3. ábra.** A felügyelt tanulás folyamata [54]

A felügyelt tanulásnál megkülönböztetjük az osztályozást és a regresszióanalízist. Osztályozás esetén a cél, hogy múltbeli megfigyelések alapján megjósoljuk új adatokra, hogy milyen osztályba tartoznak. Ezek a címkék diszkrét, rendezetlen értékek, melyeket értelmezhetünk úgy, hogy adott példányok melyik csoport elemei. Gyakran említett példa a spam emailek szűrésének problémája.

Regresszió során egy folytonos jel értékét becsüljük meg. Rendelkezésünkre állnak magyarázó (prediktor) változók, melyek segítségével keressük a kapcsolatot kimeneti változóval. [54]

#### 4.2.2. Osztályozó algoritmusok bemutatása

Az osztályozó algoritmusok tehát adatokat kategorizálnak a priori ismeretek alapján. Az adatbányászat egyik népszerű területként számos technikát dolgoztak ki e problémák megoldására.

#### Döntési fák

A döntési fa osztályozóval történő tanulás egyike a legegyszerűbb, legkönnyebben implementálható algoritmusoknak. [55] A nemparaméteres tanulási módszerek közé tartozik, mely az osztályozáson kívül regresszió analízisre is alkalmas.

A döntési fa egy tesztsorozat elvégzésével jut el a döntéshez. A fa gyökércsomópontjából kiindulva halad, minden belső csomópont valamely paraméter értékére vizsgál, és az eredménynek megfelelően az egyik ágon halad tovább. A fa minden levélcsomópontja egy kimeneti értéket takar. [56]

A döntési fák tehát könnyen vizualizálhatók, kevés adatelőkészítést igényelnek, például nem szükséges normalizálni a bemeneti adathalmazt. Előnyök közé tartozik még, hogy kevés számítás igényel, sokféle adattípussal használható.

Azonban a fa felépítése költséges lehet időben, és magas dimenziójú adathalmazok feldolgozása is nehezebb. Az optimális döntési fa megtanulása NP-teljes probléma, így a

gyakorlatban olyan heurisztikus algoritmusokat használunk, amelyek közel optimális eredményt adnak.

Ilyen például a mohó-algoritmus, amely lokálisan optimális döntéseket választ minden csomóponthoz. Természetesen ezzel nem garantálható, hogy megtaláljuk a legjobb struktúrát, azonban több fa együttes, paramétereket véletlenszerűen mintavételező tanításával kiválaszthatjuk a legsikeresebbet közülük.

### **Legközelebbi szomszéd**

A KNN (K Nearest Neighbour), vagyis K legközelebbi szomszéd algoritmus a nemparaméteres módszerek alcsoportjába, a példányalapú tanulási eljárások közé tartozik. E csoportba tartozó algoritmusokra az jellemző, hogy nem építenek fel egy belső modellt, hanem eltárolják a tanító adatokat. Az osztály meghatározása egyszerűen a legközelebbi szomszédokhoz tartozó címkék alapján történő többségi szavazással kerül kiszámításra. Tehát a kérdéses ponthoz az az osztály fog tartozni, amelyikből a legtöbbet találunk az adatpont adott környezetében.

Felmerülhet a kérdés, mi alapján definiálhatjuk a pont szomszédságát. Egyik megközelítés, hogy egy fix távolságot határozzunk meg, ekkor előfordulhat, hogy túl kicsi érték esetén nem találunk adatpontot, vagy ellenkező esetben az összes adatpontot figyelembe kell vennünk. [56]

Másik módszer, hogy egy ésszerű K érték megválasztásával rögzítjük, hogy a legközelebbi szomszédok közül hányat veszünk figyelembe. A gyakorlatban az 5-10 közé eső értékkel érik el a legjobb eredményt.

A KNN algoritmus az ún. lazy-learning módszerek közé tartozik, amelyek esetében a tanítási fázis igényel kevesebb számítási teljesítményt. Ez az algoritmus pedig konkrétan nulla tanítási költséggel rendelkezik.

A módszer előnye, hogy az osztályozó azonnal adaptálja magát az új adatokhoz, azonban velejáró hátránya, hogy a számítási igény lineárisan nő az tanító adathalmaz méretével.

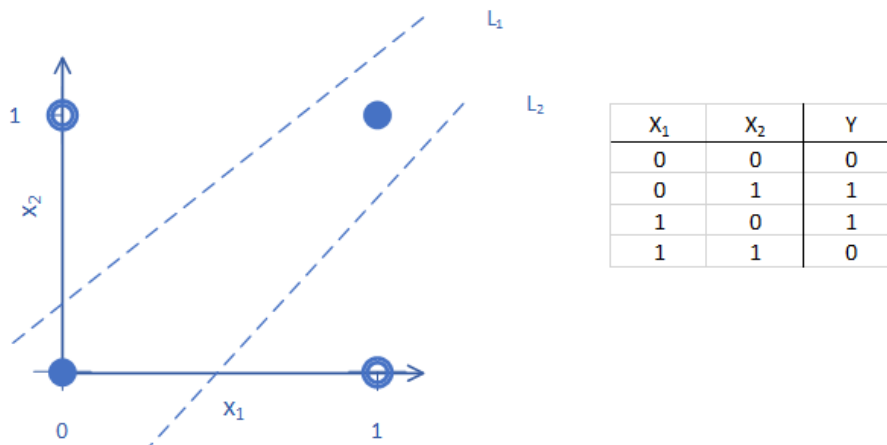
További hátránya az algoritmusnak, hogy érzékeny az esetleges zajra és az irreleváns paraméterekre.

### **Neurális háló**

A neurális háló olyan számítási feladatok megoldására képes rendszerek, amelyek a biológiai rendszerek feldolgozási módszerén és tanulási képességén alapulnak, modellezve néhány tulajdonságukat.

A Perceptron az egyik legegyszerűbb neurális hálózat, amely gyakorlatilag egy egyrétegű, egy feldolgozó elemből álló előrecsatolt hálózat. Lineárisan elválasztható adathalmazok osztályozására alkalmas.

Azonban léteznek nem lineárisan szeparálható esetek is, mint a példaként gyakran említett XOR probléma. A XOR művelet a kizáró vagy kapcsolatot jelenti, amelynek igazságtábláját és grafikus ábrázolását a 4.4. képen láthatjuk. Az osztályok egy lehetséges elválasztását szemléltetik az  $L_1$  és  $L_2$  egyenesek.



4.4. ábra. A XOR probléma

Ilyen esetekben a Multi-Layer Perceptront (MLP) a leggyakrabban alkalmazott hálózati struktúra. Az MLP egy rejtett rétegeket tartalmazó struktúra, képes komplex függvényeket modellezni, robusztus az irreleváns bemenetekkel és a zajjal szemben. [57]

### Szupport vektor gépek

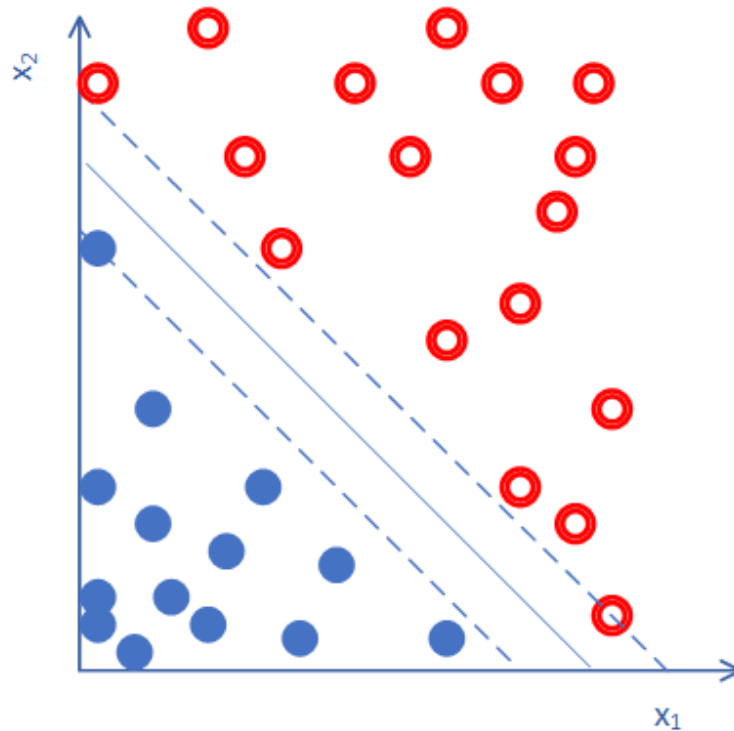
A Support Vector Machines, röviden SVM algoritmus egy széles körben használt osztályozó, amely a Perceptron kiterjesztésének is tekinthető. Míg a Perceptron célja az osztályozási hibák minimalizálása volt, az SVM optimalizációs célja, hogy maximalizálja az osztályok példányait elválasztó hipersíkhöz tartozó tartalékot. A tartalék nem más, mint az osztálypéldányokat szétválasztó sáv szélessége, amelynek a szemléltetését láthatjuk a 4.5. ábrán.

Lineárisan elválasztható adatok esetén, miután megtaláltuk a megfelelő szeparáló síkot, vegyük a tartalékon fekvő adatpontokat. Ezeket, a szeparátorhoz legközelebb eső pontokat nevezzük szupport vektoroknak. Az algoritmus kizárólag őket veszi figyelembe, ezért a modell komplexitása független a tanító halmaz méretétől, ami ideálissá teszi nagyszámú bemenettel rendelkező problémákra való alkalmazáshoz. [58]

Nem szeparálható esetekben kernelfüggvények alkalmazásával magasabb dimenziós térbe transzformálhatjuk a bemenetünket, ahol már találhatunk lineáris szeparátorokat. A megfelelő függvény kiválasztása meghatározó a tanítás sebességében.

### Bayes-hálók

A Bayes-hálók, mint valószínűségi modellek tanulási módszerei az 5.2. szakaszban kerülnek ismertetésre, itt csak mint a statisztikai tanulási módszerek egyikeként említjük meg.



4.5. ábra. Kétdimenziós, lineáris szeparálható adathalmaz szemléltetése.

### 4.2.3. Tanító adatok gyűjtése

Ahhoz, hogy a 4.1.3. szakaszban ismertetett feladatra különböző tanulóalgoritmusok működését megvizsgálhassuk, nagyszámú tanítóadat összegyűjtésére van szükség, amelyek forrásul szintén a 3.3.4. szakaszban már említett lehetőségek álltak rendelkezésünkre.

A tanító adatok feldolgozása, címkézése időigényes folyamatnak ígérkezett, hiszen ahogy az már a kockázatelemzésnél alkalmazott háló tanításánál is említésre került, a különböző forrásokból érkezett adatok leíró struktúrája nem teljesen egyezett meg, így az egységesítésük is további munkát igényelt.

Továbbá a címkézési folyamatot "kézzel" végeztük el, ugyanis nem ismert számunkra, hogy várhatóan milyen paramétereiktől függ, illetve milyen összefüggés alapján adódik egy adott szituációhoz rendelhető kimeneti osztály.

### 4.2.4. A tanítási folyamat és az eredmények értékelése

Az implementációhoz a Python programozási nyelv scikit-learn modulját választottuk, mivel az gépi tanulás terén ismeretes főbb algoritmusok mindegyike optimalizáltan megtalálható a könyvtárban, lényegesen megkönnyítve a fejlesztést, tesztelést. Támogatást nyújt az adatok vizualizációjára, és a tanítás kiértékelésére különböző metrikákkal. [55]

A bemeneti adatok összegyűjtését követően első lépésként megvizsgáltuk, hogy SVM-et használva konvergál-e az algoritmus. A kezdeti fókusz azért került a szupport vektor gépekre, mert a szimulátorral nagyszámban előállított tanító adatok felhasználásához célszerűnek

adódott először egy olyan algoritmus teljesítményét megvizsgálni, amelyiket nem befolyásolja negatívan a bemenet mérete.

Folytatásként ki kell értékelni a többi gyakorlatban alkalmazott osztályozó algoritmusok tanulási folyamatának eredményeit is, illetve még feladat, hogy a bemeneti adatok további előfeldolgozásával mennyit lehet javítani az osztályozás pontosságán.

A rendszerben a szituáció analízisre épül a viselkedéstervező komponens. Egyrészt a kockázatelemzés meghatározza számára, hogy biztonsági szempontból milyen döntések mérlegelése jöhet egyáltalán szóba. Másrészt a szituáció osztályozás a viselkedéstervező taktikai prioritásait befolyásolhatja. Például navigáció szempontjából releváns autópálya lehajtó közeledésével fontos kérdés, hogy mikor válik szükségessé a megfelelő forgalmi sávba való átváltás. Sűrűbb forgalmi helyzetek érzékelése elősegítheti a taktikai cél megvalósulását azzal, hogy sürgetésre kerül a végrehajtandó manőver végrehajtásának elkezdése.



## 5. fejezet

# Valósídejű viselkedéstervezés

Az eddigi fejezetekben láthattuk, hogy az autó hogyan képes megbecsülni az elvégezhető manőverekkel járó kockázatokat, illetve elemezni az őt körülvevő forgalmi helyzetet. A továbbiakban ejtsünk szót a beavatkozást megelőző döntést hozó viselkedéstervező komponens működési elvéről.

### 5.1. A feladat körvonalazása

A viselkedéstervezés absztraktabb fogalom, mint a jármű pusztá irányítása. Nem csupán egy forgalmi sávban akarunk maradni egy adott sebességgel, ennél lényegesen magasabb szinten gondolkodunk: sávot szeretnénk váltani, időnként végrehajtanánk egy-egy előzést annak érdekében, hogy egy bizonyos szempontból (például menetidő, fogyasztás, navigációs okok, stb.) optimális döntést hozzunk. Az egyszerűség kedvéért megfogalmaztunk alapvető feladatokat, amelyeket a rendszernek meg kell oldania:

1. Saját sávban haladás
2. Előzés
3. Sávvtáltás
4. Veszélyelkerülő manőversorozat megtervezése

A felsorolásból láthatjuk, hogy a feladat gyakorlatilag az autópályán való vezetés algoritmizálása az emberi gondolkodás szintjén. A valóságban mi sem sávközépre szabályozás és gázpedálállás szintjén közlekedünk, hanem figyeljük a környező járműveket és egyéb objektumokat, tisztában vagyunk a saját céljainkkal és ezek elérése érdekében hozunk döntéseket. Egy adott helyzetben járművezetési ismereteink és korábbi tapasztalataink alapján cselekszünk. Az algoritmus kidolgozása során ebből adódóan ismét a mesterséges intelligenciához nyúltunk és valószínűségi hálók alkalmazásával dolgoztunk ki egy elméleti megoldást.

A viselkedéstervező modul megalkotását részfeladatokra bontottuk, melyek a következők:

1. A matematikai apparátus, a valószínűségi hálók felépítésének megismerése.
2. Be- és kimenetek definiálása.
3. Egyszerű háló tervezése.
4. Tanító adatok gyűjtése, címkézése, a háló tanítása.
5. Döntéshozó célfüggvény definiálása.
6. A megalkotott rendszer tesztelése Software in the Loop (SiL) környezetben.
7. Háló bővítése további bemenetekkel.
8. A bővített háló tanítása.
9. A bővített háló tesztelése.
10. A megalkotott rendszer tesztelése valós járművön.

Az első háló tesztelését követő bővítés, ismételt tanítás és tesztelés természetesen többször elvégzendő lépések, az algoritmus finomítása nem oldható meg két iterációval.

Az algoritmus fejlesztése jelenleg egy egyszerű háló felépítésénél és annak tanításánál jár, a fejezet következő részeiben a felsorolt pontokat eddig fogjuk ismertetni. A további lépésekről, az azokhoz szükséges további feladatokról a jövőbeli terveket ismertető szekcióban fogunk szót ejteni.

## 5.2. Valószínűségi hálók elmélete

Az algoritmus ismertetése előtt a választásunkat fogjuk indokolni, majd egy rövid példán keresztül a valószínűségi hálók matematikai alapjait fogjuk ismertetni. Az elmélet összefoglalásában egy mesterséges intelligenciáról szóló angol nyelvű szakirodalom [59] magyar nyelvű fordításának [60] elektronikus változata [56] volt segítségünkre.

### 5.2.1. A választás okai

A viselkedéstervezés mint feladat algoritmizálása korán sem triviális probléma. Nem egy egyszerű szabályozásról van szó, lényegesen magasabb szinten kell a járművezetés fogalmát megfognunk. Egy ember vezetés közben a forgalmi szituáció tudat alatti elemzésével, korábbi tapasztalatai alapján hoz döntéseket, sokszor hagyatkozik a megérzéseire. A bizalom elve alapján próbálja megbecsülni a forgalom további résztvevőinek mozgását, reflexből

vált sávot, minden ilyen manővert az általa kitűzött cél érdekében tesz meg: a lehető leggyorsabban vagy leggazdaságosabban szeretne eljutni A-ból B-be, miközben saját és társai biztonságát is szem előtt tartja.

A probléma kezelésére a valószínűségi hálóval való leírást választottuk, ennek okai a következők:

- *A jármű tudása bizonytalan:* A környezetérzékelésből adódó információk bizonyos konfidenciával adják meg a környékünkön található objektumokat, így gyakorlatilag az algoritmus bemeneti változói valószínűségi változók, melyeknek van várható értékük, szórásuk.
- *A feladat összes bemenete nem definiálható:* Ha a vezetést egy programmal szeretnénk leírni, akkor biztosan nem ismerjük az összes olyan paramétert, amely a döntéseket befolyásolni képes. Ennek több oka van:
  - *Lustaság:* túl nagy munkát jelentene az okok és okozatok teljes rendszerének definiálása.
  - *Elméleti tudatlanság:* nem ismerjük a vezetés mint algoritmizálható feladat összes paramétereit.
  - *Gyakorlati tudatlanság:* ha ismernénk is a teljes szabályrendszert, akkor is számos bizonytalanság övezné a paraméterek meghatározását és mérését.
- *Közel áll az emberi gondolkodáshoz:* A valószínűségi hálóval egy magas szintű probléma kezelését a gondolkodásunkhoz hasonló modellel írhatjuk le, grafikus ábrázolása pedig szintén átláthatóvá teszi a megoldást.
- *A leírás csökkenti a paraméterek számát:* Számos akár ismert, akár ismeretlen paraméter kiváltható egyetlen valószínűségi változóval. Amennyiben a paraméterek száma kevésnek bizonyul, a háló tovább bővíthető, addig, ameddig a működést megfelelően nem találjuk. A paraméterek alacsony száma több előnnyel is jár:
  - Rejtett paramétereket is figyelembe veszünk a valószínűségeken keresztül.
  - Nem bonyolítjuk felesleges paraméterekkel az algoritmusunkat.
  - Valós idejű alkalmazásban a számítási időt jelentősen redukálhatjuk.

Képzeld el, hogy kezdetben egy sávváltásról csak az alapján döntünk, hogy üres-e a mellettünk lévő sáv, illetve előttünk lassan halad-e valaki. Ha az algoritmus ez alapján azt mondja, hogy ilyen esetekben például 60% valószínűséggel megéri váltani, ezt a szabályok és a forgalmi szituáció megengedik, akkor egy nagyon leegyszerűsített modell segítségével hoztunk döntést, mégis az esetek nagyobb részében fog ez jónak bizonyulni. Ha ezt a leírást további változókkal bővítjük, akkor egyre közelebb leszünk az optimális megoldáshoz, de még továbbra is távol vagyunk attól a gyakorlatilag lehetetlen feladattól, hogy a vezetést az összes létező paraméter figyelembevételével kezeljük.

- *Az algoritmusban megjelenik a predikció:* Egy sofőr döntéseit sokszor a környező járművek mozgásából adódó jóslatok figyelembevételével jellemzi. A valószínűségi változókon keresztül könnyen megfoghatóak az ilyen paraméterek, a szenzorok által detektált objektumok mozgását elemezve korábbi mérési adatok alapján becsléseket tudunk adni a közeli jövő tekintetében. Például egy mellettünk nagy sebességgel elhaladó autóról feltételezhetjük, hogy nagy valószínűséggel nem fog hirtelen megállni, de ha előtte lassan haladnak, akkor nagy az esélye annak, hogy ő is fékezni fog.
- *Valószínűségi változók alapján tudunk döntéseket hozni:* Ha egy valószínűségi hálót további csomóponttípusokkal egészítünk ki, akkor a döntési problémát egy úgynevezett döntési hálóval fogjuk reprezentálni. A döntési háló az ágens állapotáról tartalmaz információkat, megadja lehetséges cselekedeteit és azt az állapotot, amelyet ezek eredményeznek. Innentől az állapot hasznosságát egy hasznosságfüggvény maximalizálásának problémájaként tudjuk kezelni.

### 5.2.2. Valószínűségi alapfogalmak

A Bayes-hálók bemutatása előtt szükséges néhány matematikai fogalom definiálása. A szakirodalomban [61] ezeknek lényegesen nagyobb csoportjáról esik szó, itt csak a téma szempontjából legfontosabbakat említjük meg.

- *Valószínűségi vagy véletlen változó:* Az  $\eta(\omega) : \Omega \rightarrow \mathbb{R}$  függvényt valószínűségi változónak nevezzük, ha a  $P(a \leq \eta < b)$  valószínűség létezik  $\forall a < b$  esetén.
- *Értéktartomány:* A valószínűségi változó értékkészlete, három csoportba sorolható:
  - *Boole-típusú:* két érték lehetséges (igaz vagy hamis)
  - *diszkrét:* véges, vagy megszámlálhatóan végtelen tartományból vesz fel értékeket
  - *folytonos:* valós értéket vehet fel
- *Elemi esemény:* Egy elemi esemény a világ állapotának egy teljes leírását jelenti, a világot alkotó összes változóhoz való konkrét érték hozzárendeléseként is tekinthetjük. Az elemi események egymást kölcsönösen kizárják, legfeljebb az egyikük lehet igaz. Az összes elemi esemény halmaza kimerítő: az elemi események teljes halmazát tekintve pontosan az egyikük igaz. Az összes elemi esemény egyesítése ekvivalens az *igaz* állítással.
- *A priori valószínűség:* Azt a meggyőződési mértéket jelenti, amely információ hiányában az állításhoz kapcsolható, gyakorlatilag az előismereteinket reprezentálja.
- *A posteriori valószínűség:* Tények birtokában számított feltételes valószínűség, utólagos következtetés eredménye.

- *Valószínűségi változó eloszlásfüggvénye:* Egy valószínűségi változó eloszlásfüggvényének nevezzük azt a  $F(x)$  függvényt, amely minden valós  $x$  értékhez hozzárendeli annak valószínűségét, hogy az valószínűségi változó milyen valószínűséggel vesz fel  $x$ -nél kisebb értékeket:  $F(X) = P(\eta < x), \forall x \in \mathbb{R}$ .
- *Valószínűségi változó sűrűségfüggvénye:* Legyen  $\eta$  folytonos valószínűségi változó és az  $F(x)$  eloszlásfüggvénye legyen (esetleg véges sok pont kivételével) mindenütt definiálható, akkor azt a nemnegatív  $f$  függvényt, amelynek integrálfüggvényeként az  $F$  előállítható, a valószínűségi változó sűrűségfüggvényének nevezzük:

$$\int_{-\infty}^x f(\xi) d\xi = F(x) \quad (5.1)$$

- *Feltételes valószínűség:* Megadja az  $A$  teljesülésének valószínűségét akkor, ha  $B$  igaz. Ha  $P(B) \neq 0$ , akkor a  $P(A|B)$  feltételes valószínűséget az alábbi módon számítjuk:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (5.2)$$

Ha a (5.2) egyenletet átrendezzük, akkor megkapjuk a szorzatszabályt:

$$P(A \cap B) = P(A|B) \cdot P(B) \quad (5.3)$$

A szorzatszabályt értelmezése egyszerűbb: abból következik, hogy  $A$  és  $B$  együttes teljesüléséhez szükséges, hogy  $B$  igaz legyen, valamint hogy  $A$  is igaz legyen  $B$  feltétele mellett.

- *Teljes együttes valószínűség-eloszlás:* Az olyan együttes valószínűség-eloszlást, amely lefedi a változók teljes halmazát teljes együttes valószínűség-eloszlásnak nevezzük.
- *Bayes-tétel:* Egy feltételes valószínűség és a fordítottja között állít fel kapcsolatot. Írjuk fel a (5.3) egyenletet a  $P(A|B)$  és a  $P(B|A)$  feltételes valószínűségekkel:

$$P(A \cap B) = P(A|B) \cdot P(B) \quad (5.4)$$

$$P(B \cap A) = P(B|A) \cdot P(A) \quad (5.5)$$

Látható, hogy az egyenletek bal oldalán ugyanaz a valószínűség található:

$$P(B|A) \cdot P(A) = P(A|B) \cdot P(B) \quad (5.6)$$

Ebből átrendezéssel azt kapjuk, hogy ha ismert az  $A$  és  $B$  események valószínűsége, és a  $P(B|A)$  feltételes valószínűség, akkor a  $P(A|B)$  az alábbi formulával számítható:

$$P(A|B) = P(A) \frac{P(B|A)}{P(B)} \quad (5.7)$$

A (5.7) egyenlőség az úgynevezett *Bayes-tétel*.

Ha  $A_i (i \in I)$  egy teljes eseményrendszer, akkor:

$$P(B) = \sum_i P(B \cap A_i) = \sum_i P(B|A_i) \cdot P(A_i) \quad (5.8)$$

A (5.8) egyenlőséget felhasználva megkapjuk a tétel teljes eseményrendszerekre alkalmazható alakját:

$$P(A_i|B) = \frac{P(B|A_i) \cdot P(A_i)}{P(B)} = \frac{P(B|A_i) \cdot P(A_i)}{\sum_j P(B|A_j) \cdot P(A_j)} \quad (5.9)$$

- *Független események:* Az  $A$  és a  $B$  események egymástól függetlenek, ha:

$$P(A) = P(A|B) = P(A|\bar{B}) \quad (5.10)$$

Az események függetlensége esetén használható a szorzási szabály:

$$P(A \cap B) = P(A) \cdot P(B) \quad (5.11)$$

### 5.2.3. Bayes-hálók felépítése

A valószínűségi számításokkal kapcsolatos alapfogalmak összefoglalása után rátérhetünk a hálók felépítésének vizsgálatára. A Bayes-háló egy gráf<sup>1</sup>, amelynek minden pontjához számszerű valószínűségi információk vannak csatolva. Jellemzői a következők:

1. A háló csomópontjait diszkrét vagy folytonos valószínűségi változók egy halmaza alkotja.
2. Irányított élek (nyilak) egy halmaza összeköt bizonyos csomópontpárokat. Ha létezik nyíl az  $X$  csomópontból az  $Y$  csomópontba, akkor  $X$  szülője  $Y$ -nak.
3. Minden  $X_i$  csomóponthoz tartozik egy  $P(X_i|Parents(X_i))$  feltételes valószínűség-eloszlás, ami megadja a szülők hatását a változóra. Diszkrét változók esetén ezen eloszlásokat feltételes valószínűségi táblák (FVT) segítségével adjuk meg.
4. A Bayes-háló egy irányított<sup>2</sup>, körmentes<sup>3</sup> gráf.

A Bayes-háló a modellezett világ teljes leírását valósítja meg. A benne található információk lehetővé teszik az együttes valószínűség-eloszlás függvény bármely bejegyzésének<sup>4</sup>

<sup>1</sup> A gráf egy rendezett pár,  $G = (V, E)$ , ahol  $V$  egy nem-üres halmaz,  $E$  pedig ebből a halmazból képezhető párok egy halmaza.  $V$  elemeit pontoknak vagy csúcsoknak,  $E$  elemeit éleknek nevezzük. [62]

<sup>2</sup> Irányított gráf: olyan gráf, amelyben az éleket definiáló pontok megadásának sorrendje is számít, azaz az él megkülönböztetett kezdő- és végponttal rendelkezik.

<sup>3</sup> Körmentes gráf: nem található benne olyan élsorozat, amelynek kezdő- és végpontja megegyezne.

<sup>4</sup> Az együttes valószínűség-eloszlás függvény egy általános bejegyzése egy teljes (minden egyes változóhoz történő) hozzárendelés konjunkciójának a valószínűsége, úgymint  $P(X_1 = x_1 \cap \dots \cap X_n = x_n)$ . Erre a  $P(x_1, \dots, x_n)$  rövidítést szoktuk használni.

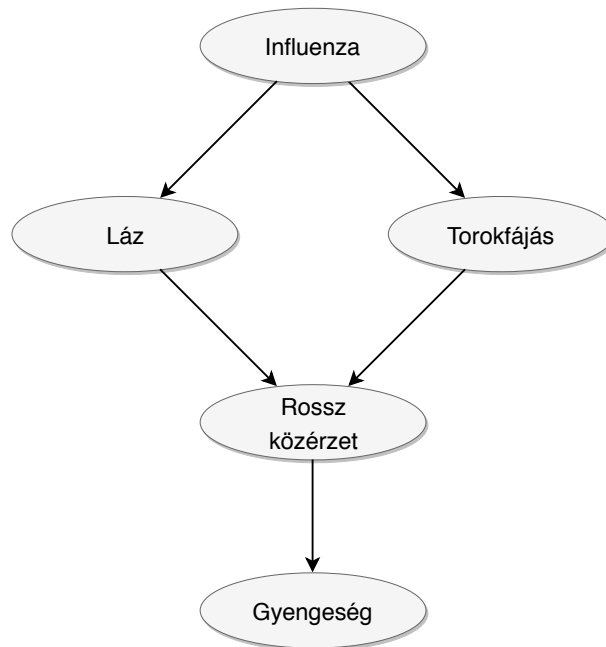
kiszámítását. Egy bejegyzés értéke a következő formulával adható meg:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(X_i | \text{parents}(X_i)) \quad (5.12)$$

ahol a  $\text{parents}(X_i)$  a  $\text{Parents}(X_i)$ -ben szereplő változók adott értékeinek együttesét jelöli, azaz az FVT egy sorának felel meg. Az együttes valószínűség-eloszlás függvényt leíró táblázat minden bejegyzése a Bayes-hálóban szereplő, feltételes valószínűségi táblák megfelelő elemeinek a szorzata. Ezzel az FVT-k valójában az együttes valószínűség-eloszlás függvény dekomponált leírását adják meg. Az állítások természetesen folytonos valószínűségi változók esetére is általánosíthatók.<sup>5</sup>

Rengeteg elméleti állítás és tétel kapcsolódik még a témához, ám dolgozatunk feladata nem ezek ismertetése és bizonyítása, sokkal inkább gyakorlati alkalmazása. A következőkben a száraz matematikai fogalmak helyett egy egyszerű tankönyvi példán [63] keresztül szeretném szemléltetni egy Bayes-háló felépítését. Az egyszerűség kedvéért bináris változókkal fogunk dolgozni, de ahogyan azt említettem, folytonos eloszlásokra is általánosítható elméletről van szó.

A példában az influenza és a hozzá kapcsolódó tünetek kapcsolatát fogjuk modellezni. A betegség gyakori tünete a láz és ritkább esetben a torokfájás. Ezen két tünet rossz közérzethez vezet, amely gyengeséget okoz. Az alábbi ábrán az ismertetett egyszerű problémához tartozó Bayes-háló látható.



5.1. ábra. Az influenza tüneteit reprezentáló Bayes-háló [63]

Ha visszagondolunk a háló jellemzőit ismertető felsorolásra, akkor láthatjuk, hogy már

<sup>5</sup> Folytonos változók esetében egy elterjedt módszer az eloszlás diszkrétizálása, majd a csomópontokhoz véges elemet tartalmazó feltételes valószínűségi táblák megadása, vagy pedig kevés paraméterrel leírható nevezetes eloszlások használata (például Gauss).

csak egy dolog hiányzik a pontos megadáshoz: a csomópontokra jellemző valószínűségi eloszlások, a feltételes valószínűségi táblák megadása. A láz valószínűségét az influenza igaz értéke mellett 80%-nak tételezzük fel, a táblázat második sorában pedig azt jelezzük, hogy a betegség megléte nélkül mekkora eséllyel jelentkezik a tünet.

**5.1. táblázat.** *A láz csomópont feltételes valószínűségi táblája*

Influenza	P(Láz   Influenza)
Igaz	0,8
Hamis	0,015

Az előzővel teljesen analóg módon adható meg a torokfájás csomópont feltételes valószínűségi táblája. Ez a tünet ritkábban jelentkezik, a példában 23% valószínűséggel szerepel influenza megléte esetén, 1,1%-ban pedig a betegség hiányakor.

**5.2. táblázat.** *A torokfájás csomópont feltételes valószínűségi táblája*

Influenza	P(Torokfájás   Influenza)
Igaz	0,23
Hamis	0,011

A következő csomópont a rossz közérzet, amely az eddigiektől eltérő módon két szülővel rendelkezik. Az FVT ekkor tartalmazza a szülők által felvehető logikai értékek összes lehetséges kombinációjára a feltételes valószínűségeket.

**5.3. táblázat.** *A rossz közérzet csomópont feltételes valószínűségi táblája*

Láz	Torokfájás	P(Rossz közérzet   Láz, Torokfájás)
Igaz	Igaz	0,89
Igaz	Hamis	0,2
Hamis	Igaz	0,73
Hamis	Hamis	0,01

A gyengeség csomópont feltételes valószínűségi táblája az első kettővel teljesen analóg, mivel ennek is csak egy szülője van.

Ha a Bayes-hálóból ezek után kíváncsiak vagyunk az együttes valószínűség-eloszlás függvény egy általános bejegyzésére, akkor a (5.12) egyenlőség segítségével elvégezhetjük a dekomponálást. Ehhez először fel kell állítanunk egy változósorrendet, amely esetünkben leolvasható a struktúrából. Az influenza legyen  $X_1$ , a láz  $X_2$ , a torokfájás  $X_3$ <sup>6</sup>, a rossz közérzet  $X_4$ , a gyengeség pedig  $X_5$ . Rövid megfontolás után a dekomponálás eredményeként az alábbi egyenlőséget kapjuk:

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1) \cdot P(x_2|X_1) \cdot P(x_3|X_1) \cdot P(x_4|X_2, X_3) \cdot P(x_5|X_4) \quad (5.13)$$

Tegyük fel, hogy egy téli járványidőszakban 10% valószínűséggel kapja el valaki az influenza vírust. Mekkora a valószínűsége annak, hogy a betegnek nem lesz láza, de minden egyéb tünetet produkálni fog? Az előbb kapott képletbe helyettesítve a megadott táblázatok

<sup>6</sup> A struktúrából egyértelműen látszik, hogy az  $X_2$  és  $X_3$  változók szabadon felcserélhetők, így nem ez az egyetlen helyes hozzárendelés.



**5.4. táblázat.** A gyengeség csomópont feltételes valószínűségi táblája

Rossz közérzet	P(Gyengeség   Rossz közérzet)
Igaz	0,8
Hamis	0,05

megfelelő sorait megkaphatjuk az eredményt:

$$\begin{aligned}
 & P(x_1 = i, x_2 = h, x_3 = i, x_4 = i, x_5 = i) = \\
 & = P(x_1 = i) \cdot P(x_2 = h|X_1) \cdot P(x_3 = i|X_1) \cdot P(x_4 = i|\overline{X_2}, X_3) \cdot P(x_5 = i|X_4) = \\
 & = 0,1 \cdot (1 - 0,8) \cdot 0,23 \cdot 0,73 \cdot 0,8 = 0,00268 = 0,268\%.
 \end{aligned}$$

Ebben az egyszerű példában láthattuk, hogy a Bayes-hálóknak nagyon hatékony eszközt jelentenek feltételes eloszlások reprezentálására, kiválóan alkalmasak ok-okozati kapcsolatokat grafikus megjelenítésére és közben bizonyítás nélkül ugyan, de láthattuk, hogy a csomópontokhoz tartozó feltételes valószínűség-eloszlás függvények segítségével leírható az együttes valószínűség-eloszlás függvény is.

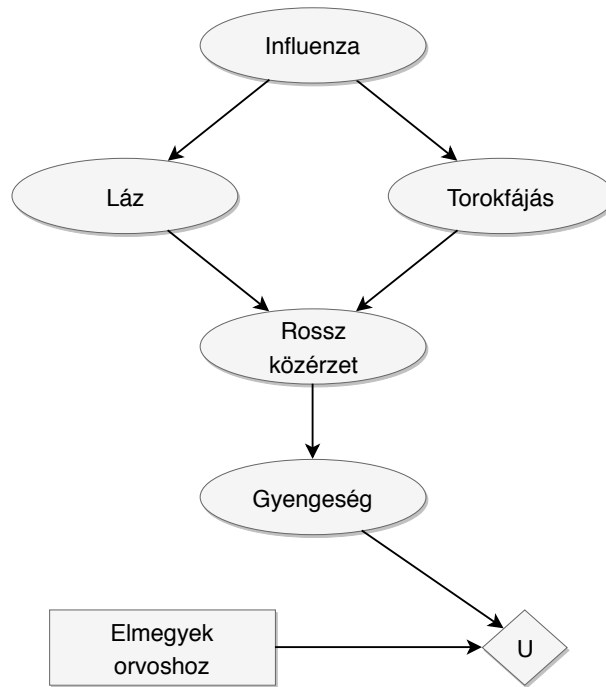
Az előző példában csak az élek előre mutató ágaival foglalkoztunk, azonban könnyen belátható, hogy a (5.7) és (5.9) egyenletekkel leírt Bayes-tétel alapján a feltételes valószínűségek fordítottja is kiszámítható. Ezen tulajdonságok alapján orvosi diagnosztikai szoftverekben előszeretettel használják ezt az algoritmust.

Megemlítendő még, hogy a háló segítségével tetszőleges feltételes valószínűség kiszámítható. Erre a legegyszerűbb algoritmus a teljes együttes eloszlás tagjainak összegzése. Ha például az előző példában arra vagyunk kíváncsiak, hogy mekkora valószínűséggel lázas valaki, feltéve hogy fáj a torka és rossz a közérzete, akkor kiszámítjuk az összes ilyen eshetőséget és az összegzésből következtetünk. Az algoritmus optimalizálására léteznek különböző eljárások, amelyekkel nagy változószám esetén a hatékonyság növelhető, illetve közelítő eljárások is alkalmazhatók, de ezt a témakört nem célok tárgyalni.

### 5.2.4. Döntési hálóknak

A hatásdiagramok, vagy más néven döntési hálóknak további csomóponttípusokat definiálnak cselekvések és hasznosságok reprezentálásához, gyakorlatilag ezekkel bővítik a valószínűségi hálót. A döntési háló a jelenlegi állapotról tartalmaz információkat, megadja a lehetséges cselekvéseket, azt az állapotot, amely a cselekvés következménye lesz, illetve ennek az állapotnak a hasznosságát. A három csomóponttípus tehát a következő:

- *Véletlen csomópont (ovális):* valószínűségi változó, a Bayes-hálóknakban ilyenekkel találkozunk.
- *Döntési csomópont (négyzet):* azok a beavatkozási pontok, ahol a döntéshozónak lehetősége van cselekvésre.
- *Hasznosságcsomópont (rombusz):* a hasznosságfüggvényt reprezentálja.



**5.2. ábra.** Az influenza tüneteit reprezentáló háló kiegészítése döntési hálóvá

A háló kiértékelésekor a cselekvéseket úgy választjuk ki, hogy a döntési csomópont minden lehetséges beállítását értékeljük. A döntési csomópont a beállítását követően úgy viselkedik, mint egy véletlen csomópont. A háló kiértékelését a következő algoritmus végzi:

1. Tényváltozók beállítása a jelenlegi állapotra.
2. A döntési csomópont minden lehetséges értékére:
  - (a) Döntési csomópont beállítása az adott értékre.
  - (b) A posteriori valószínűségek kiszámítása a hasznosságcsomópont szüleiére egy szabványos valószínűségi következtető algoritmust használva.
  - (c) Cselekvés hasznosságának kiszámítása.
3. Visszatérés a legnagyobb hasznosságértékkel.

A primitív példánknál maradva ez az jelenti, hogy két lehetőségem van: vagy elmegyek orvoshoz, vagy nem. Ha azt feltételezzük, hogy gyengeségemet valószínűségi következtetések segítségével állapítottam meg, akkor az egyetlen hasznosságcsomóponton a mindkét döntés esetére lefuttatom a számításokat. Amelyik választás hasznosabbnak bizonyul, aszerint fogok cselekedni. A cselekvés hatásának értékelésére a hasznosságcsomópontra egy többváltozós hasznosságfüggvényt kell definiálni, a probléma innentől kezdve gyakorlatilag egy egyszerű optimalizációs feladat, művészeti értéke a függvény megalkotásának van.

### 5.3. Az algoritmus megtervezése

Az olvasóban az utóbbi oldalak tanulmányozása közben felmerülhetett a kérdés, hogy még mindig egy autonóm járművek problémáit tárgyaló dolgozatot tart-e a kezében. Hogy jön az influenza az önvezetéshez? A következőkben erre a kérdésre keressük a választ. Mielőtt azonban részleteznénk a valószínűségi hálók viselkedéstervezéshez való felhasználásának módját szögezzünk le két kritériumot a modul hatáskörével kapcsolatosan:

1. A viselkedéstervező nem hozhat biztonságkritikus döntéseket. Ez azt jelenti, hogy csak azon manővertérben definiált pontok közül választhat, amelyeket a kockázatelemző algoritmus biztonságosnak ítélt meg (kockázata alacsony vagy minimális besorolású).
2. Amennyiben a manővertér összes pontjához közepes, vagy annál magasabb kockázati értéket rendelt a kockázatelemző modul, abban az esetben a döntést nem a korábban megtanult feltételes valószínűségi táblák alapján hozzuk meg.

Ez azt jelenti, hogy a valószínűségi hálóra csak olyan döntéseket bízunk, amelyek nem járhatnak a jármű és környezete veszélyeztetésével. Egy esetleges rosszul kiválasztott manőver csak egy olyan hibát jelenthet, amely a kívánt cél elérése szempontjából okoz problémát (például lassabb haladást eredményez).

#### 5.3.1. Az algoritmus bemenetei

A viselkedéstervező algoritmus a korábbi fejezetekben látott bemenetektől, illetve az eddig bemutatott kockázatelemző és szituációosztályozó modulok kimeneteiből dolgozhat. Összefoglalva a rendelkezésre álló információ tehát a következők:

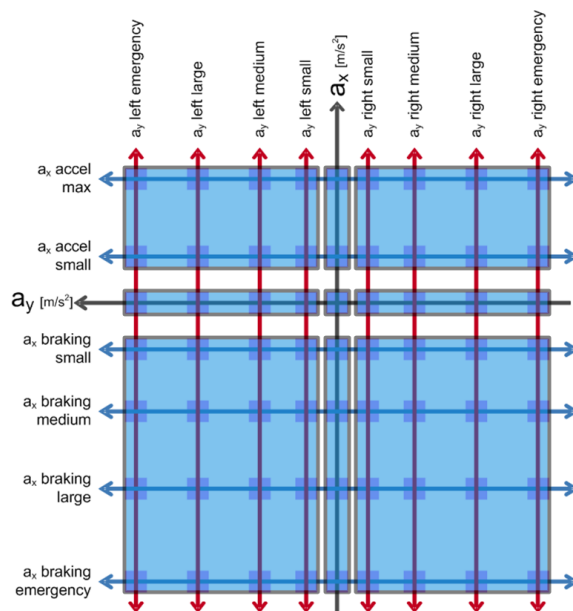
1. jármű állapotvektora
  - (a) longitudinális és laterális sebesség
  - (b) longitudinális és laterális gyorsulás
  - (c) jármű szélessége és hossza
2. környezetérzékelés adatai
  - (a) sávinformációk
  - (b) sávvonali információk
  - (c) környező járművek adatai
    - i. pozíció
    - ii. longitudinális és laterális sebesség
    - iii. longitudinális és laterális gyorsulás

3. kockázatelemző modul kimenete
4. szituációosztályozó modul kimenete

A továbbiakban leginkább a kockázatelemző modul által számított értékekre, a jármű állapotvektorára és a környező járművek sebességére fogunk alapozni.

### 5.3.2. Sáv váltási lehetőségek meghatározása

A kockázatelemző modult leíró fejezetben 63 különböző longitudinális és laterális gyorsulás limittel definiált manőverponthoz rendeltünk kockázatértékeket, melyek jelentését a 3.1. ábra szemléltette. A következőkben ezen paraméterek számát szeretnénk háromra redukálni: rendeljük valószínűségeket a saját sávban haladás, illetve a tőlünk jobbra és balra lévő forgalmi sávokba való váltás kockázatosságához. Ehhez első lépésként osszuk csoportokba (vagy zónákba) a manőverter pontjait az 5.3. ábrán látható módon.



**5.3. ábra.** Sáv váltási kockázati valószínűség zónái

A manőverteret tehát kilenc területre osztottuk két jellemző alapján:

1. longitudinális irányú gyorsulás
  - (a) gyorsítás
  - (b) sebességtartás
  - (c) lassítás
2. laterális irányú cselekvés
  - (a) sáv váltás balra
  - (b) sáv tartás

(c) sávváltás jobbra

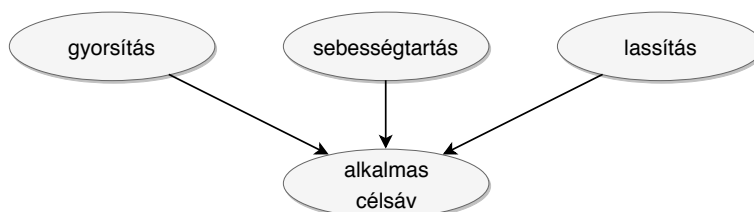
Ezt követően a zónákhoz szeretnénk egy-egy kockázati értéket rendelni, ehhez a kockázatelemzést leíró fejezetben használt összegző formulához hasonló képletet használunk:

$$R_{zone} = \sqrt[n]{\sum_{k=1}^n R_k^n} \quad (5.14)$$

ahol  $n$  a zónában található manőverpontok száma.

Ahhoz, hogy a kapott kockázatértékből valószínűséget kapjunk normálnunk kell, hogy nulla és egy közé essen, ezért ismételten a 3.1. ábrára kell támaszkodnunk. Négynél magasabb érték biztos balesetet jelent, ezért ennyivel fogunk osztani. Az egynél nagyobb értékek elkerülése érdekében az eredményt szaturáljuk. Ezzel gyakorlatilag egy valószínűségi változót kaptunk, amely a zóna kockázatosságát jellemzi.

Az eddigi számítások eredményeként célsávonként három jellemző értékünk van, melyekre definiálhatunk egy-egy egyszerű Bayes-hálót, melyet tanítva a célsávra való váltás (vagy maradás) lehetőségességére kapunk egy százalékos értéket. Ne felejtjük el, hogy veszélyes döntést nem hozhat ez alapján az algoritmus, mert a kimenet csakis biztonságos kockázati értékű manőverpont lehet.



**5.4. ábra.** Bayes-háló a célsávba való váltás lehetőségének számítására

A sávváltást a forgalmi okokon felül gátolhatják még közlekedési szabályok is, ezt a döntéshozó célfüggvény közvetlen bemeneteként érdemes figyelembe vennünk.

### 5.3.3. Sávváltási igény meghatározása

Az előző pontban már rendeltünk valószínűséget a manőver célsávjának kockázatához, mellyel meg tudjuk mondani, hogy mekkora valószínűséggel biztonságos egy adott sávban haladni, most pedig következhet a sávváltási igények meghatározása. Alapvetően két okból szoktunk sávot váltani:

1. Navigációs okból.
2. A gyorsabb haladás reményében.

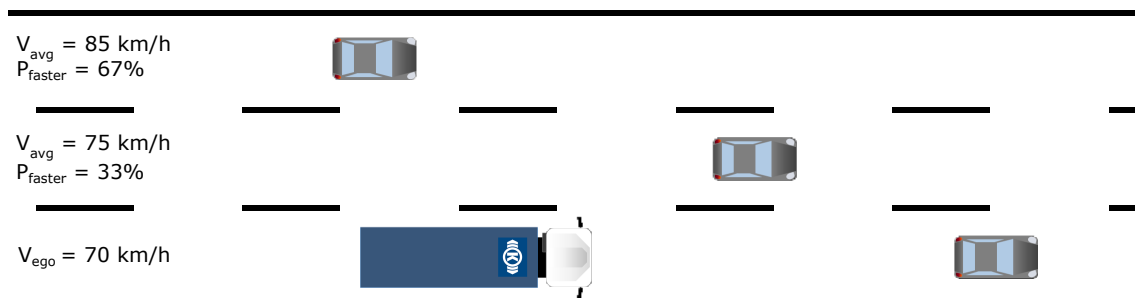
A navigációs okból adódó igény kezelése egyszerű: sávonként rendelkezésre áll az az információ, hogy a beállított célállomás elérésére alkalmas-e, vagy sem, illetve az is, hogy mikortól nem lesz már alkalmas. Például az M1-M7 autópályák közös kivezető szakaszán a Budaörs előtti lejtőnél még mindhárom sáv alkalmas arra, hogy Győrbe jussunk, de az elágazás előtt 500 méterrel már meg kell fontolnunk, hogy melyik sávban haladunk. Ha ezt a bináris értéket egy adott úthosszon egy nulla és egy között interpoláljuk (például egy egyenes mentén), akkor egy valószínűségi változónak is tekinthetjük, amely a navigációs célokat tekintve százalékosan fejezi ki, hogy mekkora eséllyel megfelelő nekünk az adott forgalmi sáv.

A gyorsabb haladás reményében történő váltás szintén nem túl bonyolult probléma. Ismerjük az adott útszakaszra érvényes maximális haladási sebességet, illetve a saját sebességünket. Látjuk a körülöttünk lévő és mellettünk elhaladó járműveket, tudjuk, hogy melyik sávot használják. Ezen adatok alapján a mellettünk lévő sávokra tudunk átlagsebességet számítani, és ha azt látjuk, hogy például a tőlünk balra lévő sávban nem korlátoznának a haladásban, akkor sávot válthatunk. Ha ehhez valószínűségi változót szeretnénk rendelni, akkor jelöljük ki két pontot, melyek között lineárisan interpolálunk:

1. Rendeljük 0% valószínűséget a miénknél nem nagyobb haladási sebességekhez.
2. Rendeljük 100% valószínűséget a miénknél egy adott értékkel (például 15 km/h) nagyobb sebességekhez.

Egy forgalmi sáv átlagsebességét egy egyszerű diszkrét szűrővel mérjük, majd határozzuk meg a lehetséges haladási sebességünket váltás esetére. Ez vagy a sáv átlagsebessége lesz, vagy a megengedett maximális sebességünk, amennyiben egyébként a sáv ennél gyorsabban halad. Ha a sáv egy ideje üres, akkor a lehetséges haladási sebesség a megengedett maximális. A kapott értékhez rendeljük valószínűséget az említett szabály alapján.

Ha például egy autó 70 km/h-val halad előttünk a külső sávban, eközben a belső sáv 85 km/h-val halad, akkor elvileg 100%, hogy megéri váltani. Vegyük azonban figyelembe, hogy kamionnal közlekedünk, amely csak 80 km/h sebességgel közlekedhet, így a forgalmi okból jelentkező váltási igényt jellemző valószínűségi változó csak 67% értéket vesz fel.



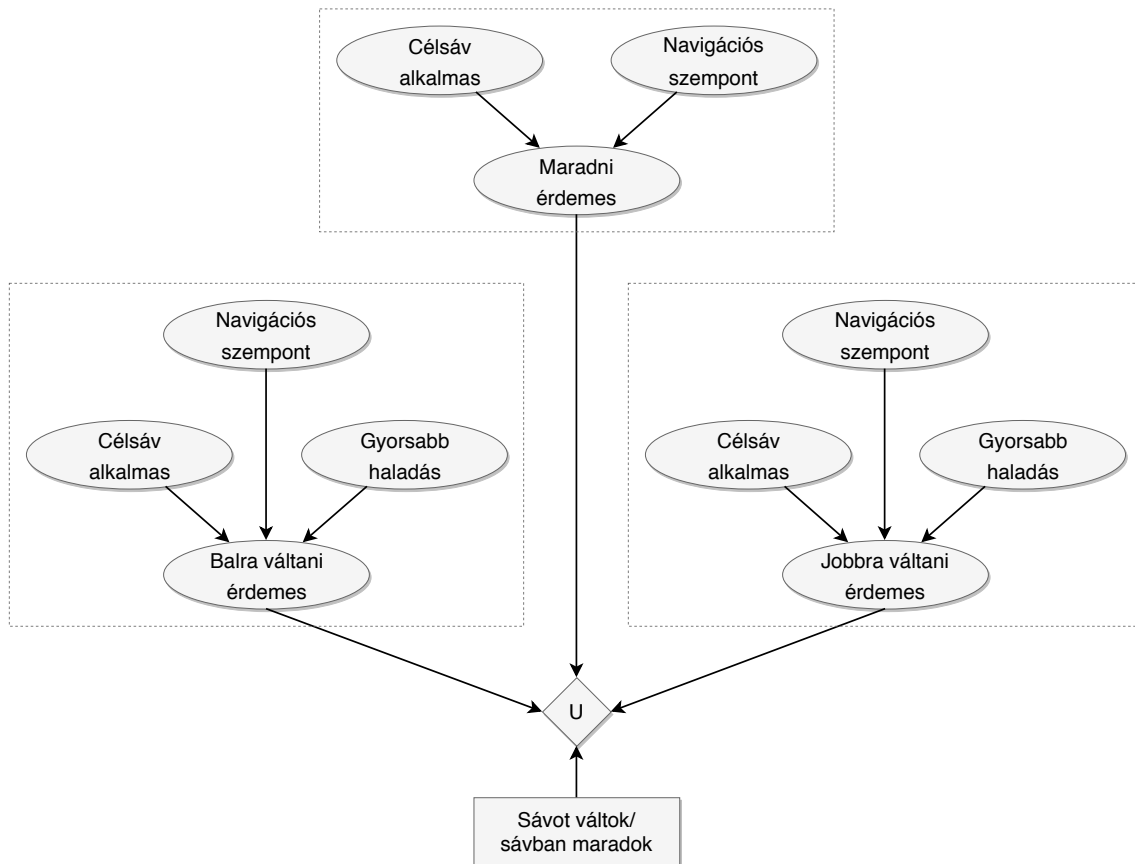
5.5. ábra. Átlagos haladási sebesség és a gyorsabb haladás valószínűsége

### 5.3.4. A valószínűségi háló felépítése

Az eddig felsorolt egyszerű számításokkal képzett valószínűségi változók lehetővé teszik a fejezet bevezetésében említett egyszerű Bayes-háló felépítését. Foglaljuk össze a háló bemeneti pontjait:

1. Célsávok alkalmasságának valószínűsége kockázat alapján.
2. Navigációs célú sávváltási igény.
3. Gyorsabb haladást lehetővé tevő sávváltás igénye.

A felsorolt bemeneteket felhasználva sávonként levonhatjuk a következtetést, hogy érdemes-e oda váltani (vagy a saját sávunk esetén ott maradni), majd az így kapott valószínűségeket felhasználva meghozhatjuk a számunkra leghasznosabb döntést. A felépített Bayes-hálót az 5.6. ábra szemlélteti.



5.6. ábra. A felépített egyszerű Bayes-háló

Látható, hogy a saját sávunk esetében nincs értelme gyorsabb haladásra valószínűségi változót definiálni az átlagsebesség alapján, mivel az konstans nulla lenne. Megfigyelhető továbbá, hogy az ábra már döntési és hasznossági csomópontot is tartalmaz. A döntések a szomszédos sávokba váltást, vagy a saját sávunkban maradás lehetnek, a hasznosságfüggvény pedig egyelőre a legjobb sávot választja ki a legnagyobb valószínűség alapján.

A viselkedéstervező modul kimenetének egy manővertérbeli pontnak kell lennie, a háló viszont csak a célsávot határozza meg. Ha a sebesség maximalizálást választjuk célnak, akkor a választás egyszerű: a célsávunkba tartó manőverek közül azon legnagyobb (még szabályos) célsebességet eredményező pontot választjuk, amely kockázati értéke már elfogadható a teljes biztonság szempontjából.

### 5.3.5. Az algoritmus működése különböző szituációkban

Az előző pontban látott valószínűségi háló meglehetősen egyszerű módon, de képes kezelni a fejezet elején definiált feladatokat. A következőkben tekintsük át, hogy mit várunk el az algoritmustól a különböző szituációkban és hogy az elvárásaink hogyan teljesülnek.

**Saját sávban haladás:** Haszongépjárművek esetén az autópályán való közlekedés legjelentősebb részét ez a szituáció jellemzi. Az előző pontban vázolt valószínűségi háló (megfelelő tanítás esetén) megfelelő haladási sebesség esetén mindig a saját sávunkban fog tartani, amennyiben az navigációs célból is megfelelő.

**Előzés:** Az előzés gyakorlatilag két, egymást követő sávváltást jelent. Ha a háló a váltás mellett dönt, majd azt tapasztalja, hogy visszatérhet az eredeti sávunkba, akkor vissza is fog térni. Természetesen a jobbra (vagy baloldali közlekedés esetén balra) tartást ehhez a hasznossági függvényben különböző súlyokkal elő kell idézni. A jobbról előzés és a sávokban való cikázás meggátolható a jobbra történő gyorsabb haladást eredményező sávváltások tiltásával.

**Sávváltás:** A sávváltás annyiban tér el az előzéstől, hogy nem forgalmi, hanem navigációs cél idézi elő, ezért nem követi visszaváltás az eredeti sávunkba. A visszatérést a navigációs célból adódó valószínűség folyamatos csökkenése gátolja meg az eredeti sávunk esetén.

**Veszélyelkerülő manőversorozat megtervezése:** Ezt a szituációt az különbözteti meg a többitől, hogy ebben az esetben a manővertérben nem található megfelelő biztonsági kockázattal jellemezhető pont. Ekkor nem figyeljük a valószínűségi háló kimenetét, hanem kiválasztjuk a legbiztonságosabb (vagy legkevésbé veszélyes) pontot.

A felsorolt esetek természetesen a leírtaknál lényegesen komplikáltabbak és összetettebbek, de mivel biztonságkritikus döntéseket nem hozunk a háló segítségével, ezért az esetlegesen hibásan kiválasztott manőverpontok pusztán lassabb haladást vagy rossz irányba való letérést eredményeznek (ahogy az emberi sofőröknél is előfordulhat). Az ilyen hibák csökkentését megfelelő tanító adatokkal és a vázolt nagyon egyszerű háló bővítésével tudjuk garantálni, ehhez rengeteg tesztelés, mérés és működési tapasztalat szükséges.

Látható tehát, hogy az autonóm jármű által hozott döntéseket ugyanazzal a matematikai eszközzel tudjuk algoritmizálni, mint amellyel az influenza miatt otthon maradt sofőrjét diagnosztizáltuk.



### 5.3.6. A valószínűségi háló tanítása

Az előző pontban már említésre került a tanulás szó, azonban ennek módjáról még nem esett szó. Az elméleti bevezetésben az influenza tüneteit szemléltető példában már adottnak tekintettük a feltételes valószínűségi táblákat, tehát egy betanított hálóról volt szó. A valószínűségeket a tanítás során relatív gyakorisággal közelítjük, ehhez szükségünk van értékekkel felcímkézett tanító adatokra.

A tanításra jelenleg három lehetőséget tartok elképzelhetőnek:

1. Szimulátorból származó mérési adatok címkézése, majd a csomópontok tanítása.
2. Valós mérések címkézése.
3. Működés közbeni folyamatos tanulás vagy emberi sofőr döntéseinek tanítása.

Az első két esetben a méréseket offline címkézzük fel, gyakorlatilag egy adott szituációra megmondjuk hogy érdemes-e sávot váltani vagy nem, megfelelően sok adat esetén pedig a háló a későbbiekben következtetni tud eddig nem látott helyzetekre is.

A harmadik lehetőség egy már működő rendszert feltételez, a saját hibáiból tanul, vagy a sofőrjének döntéseit figyeli az önvezető mód bekapcsolása előtt. Ezzel a lehetőséggel óvatosan kell bánnunk: bár a viselkedéstervező modul nem hozhat kockázatos döntéseket, de egyéb szempontból rosszakat igen. Az online tanulást ezért a paraméterek keretek között tartásával korlátozni érdemes. További probléma, hogy például a sávváltási lehetőséget emberi vezetés megfigyelése közben csak akkor ismeri fel a rendszer, ha a sávváltás már megtörtént, a sofőr fejébe látó szenzoraink ugyanis nincsenek. Ezen okokból következően első körben mindenképpen szimulátorból generált vagy valós mérések manuális címkézésével fog elkészülni a háló első tanított verziója.

## 5.4. Összegzés

A fejezet során felvázolt koncepció elméletben működőképes, gyakorlati tesztjei azonban még hátra vannak. A tesztelést követően a hálót biztosan bővítenünk kell további változókkal, mint például a vezetési komfort, a szituációosztályozó modul kimenetei, vagy a fogyasztás optimalizációja.

A manőverekkel kapcsolatosan is vannak még további lehetőségek. Ha például sávot szeretnénk váltani, akkor azt jelezhetjük indexeléssel. Ha a környező járművek megfelelően reagálnak, akkor a korábban nem végrehajtható manőver biztonságossá válik.

Biztosak vagyunk abban, hogy a rendszer folyamatosan bővíthető lesz, hogy egyre kifinomultabb viselkedést érzünk el. Abban is biztosak vagyunk, hogy a feladat bőven meg fogja haladni egy TDK dolgozat kereteit, nem hónapok, hanem évek munkája áll még előttünk.

# Összefoglalás

A dolgozat terjedelméből láthatja az olvasó, hogy három nagyon összetett, de szorosan összefüggő témával volt szerencsénk dolgozni az elmúlt időszakban. Tudjuk, hogy munkánkkal a problémák egy szűk részhalmazát sikerült még csak megismernünk, és tisztában vagyunk azzal is, hogy nem három ember fogja megválaszolni az összes jövőben felmerülő kérdést.

A kockázati számítással foglalkozó rész áttekintése során tapasztalhattuk, hogy a témakör rengeteg kidolgozandó részt tartogat még a mérnökök számára, annak ellenére, hogy a mi munkánk is számos részproblémára kínál megoldási javaslatot. Érzékelhettük, hogy az alkalmazott egyszerűsítések ellenére is egy meglehetősen komplex rendszert alkottunk meg, amely mindemellett rugalmas lehetőséget nyújt a bővítések számára is, így érdemes a projektet továbbvinni.

Láthattuk, hogy a szituációosztályozás egy sokféle oldalról megközelíthető feladat. Mint döntéselőkészítő lépés, fontos, hogy valós időben, releváns információkkal lássa el a beavatkozásért felelős intelligenciát. A forgalmi sűrűség becslésén túl a többi megemlíttett irány megvizsgálendő az elkövetkezőkben.

Végezetül bepillantást nyertünk a valós idejű viselkedéstervezés problémáiba és láthattuk, hogy egy olyan strukturált környezetben, mint az autópálya sem triviális feladat a jármű viselkedését meghatározni. A vázolt egyszerű felépítésű koncepció tesztelése csak a kezdet, láthatta az olvasó, hogy a jövőben mennyi feladat vár még ránk.

Mindhárom probléma jócskán tartogat még kihívásokat a jövőre nézve, az autonóm járművek térhódítása pedig biztosan azt fogja eredményezni, hogy egyre több és több mérnök fog a miénkhez hasonló kutatásokat végezni. Örömteli számunkra, hogy bepillantást nyerhettünk egy olyan projektbe, amelyből jellegét tekintve egyelőre még nem sok található a világban.

# Köszönetnyilvánítás

Dolgozatunk nem jöhetett volna létre a Knorr-Bremse Fékrendszerek Kft. segítségével nélkül. Külön ki szeretnénk emelni Gyurkó Zoltán és Straub Kornél konzulenseinket, akik folyamatosan figyelték munkánkat. Köszönettel tartozunk Horváth Csaba felettesünknek, aki rendszeres megbeszéléseket szervezett ezzel biztosítva a témák gördülékeny haladását.

Hálásak vagyunk továbbá egyetemi konzulenseinknek, akikkel szintén rendszeres kapcsolatban álltunk munkánk során. Köszönjük Gyurkó Zoltánnak és Kiss Domokosnak, hogy a dolgozatot többször is átolvasták, hibajavításokkal és tanácsokkal segítették annak elkészültét.

# Ábrák jegyzéke

1.1. V2X kommunikációs protokollok . . . . .	8
1.2. The Stanford Cart 1980 (Forrás: Hans Moravec [8]) . . . . .	9
1.3. Az áruszállítás várható növekedése Németországban [10] . . . . .	10
1.4. A közúti árufuvarozás költségei [11] . . . . .	11
1.5. A baleset valószínűsége (y tengely) a vezetési idő túlteljesítésének (x tengely) függvényében [13] . . . . .	11
1.6. Az eredeti Trolley-dilemma vázlata . . . . .	17
1.7. Önvezető autókkal kapcsolatos morális kérdések [22] . . . . .	18
2.1. Az előfejlesztési osztály tesztautója [13] . . . . .	20
2.2. Highway Pilot (HP) funkcionális architektúrája [26] . . . . .	21
2.3. A tesztjármű szenzorrendszere [13] . . . . .	22
2.4. Rendszám tábla koordináta rendszer . . . . .	22
2.5. Az autó vonatkoztatási rendszerében definiált manővertér [26] . . . . .	23
2.6. Dinamikai korlátok figyelembevétele a manővertérben [26] . . . . .	23
2.7. dSPACE MotionDesk [28] . . . . .	24
2.8. A szimulátor sematikus felépítése . . . . .	25
2.9. A szimulátor GUI felülete . . . . .	26
3.1. Risk értékek értelmezése . . . . .	29
3.2. Részletes kockázatszámítás blokk-diagramja . . . . .	32
3.3. Laterális gyorsulást szeparáló görbe . . . . .	35
3.4. Súrlódási ellipszis . . . . .	36
3.5. Kiválasztott konkrét manőver-pont és peer jármű . . . . .	38
3.6. Hátról érkező objektumok eltolása . . . . .	38
3.7. TTC görbe . . . . .	41
3.8. TIV görbe . . . . .	41
3.9. Több közeli trajektória használatának haszna . . . . .	42
3.10. TIV súlyozásánál használt függvények, a) időfüggő tag, b) TIV függő tag, c) tagok együttes hatása . . . . .	44
3.11. Súlyos sérülés valószínűsége az ütközésben elszenvedett sebességváltozás függvényében, különböző típusú ütközések esetén, Bahouth (2014) mérései alapján[45] . . . . .	46
3.12. Fejlesztéshez használt GUI felület . . . . .	49

3.13. Bemutatott neurális háló felépítése . . . . .	51
4.1. A megerősítéses tanulás szereplői [54] . . . . .	58
4.2. A klaszterezés kimenetének illusztrációja [54] . . . . .	58
4.3. A felügyelt tanulás folyamata [54] . . . . .	59
4.4. A XOR probléma . . . . .	61
4.5. Kétdimenziós, lineáris szeparálható adathalmaz szemléltetése. . . . .	62
5.1. Az influenza tüneteit reprezentáló Bayes-háló [63] . . . . .	70
5.2. Az influenza tüneteit reprezentáló háló kiegészítése döntési hálóvá . . . . .	73
5.3. Sávváltási kockázati valószínűség zónái . . . . .	75
5.4. Bayes-háló a célsávba való váltás lehetőségének számítására . . . . .	76
5.5. Átlagos haladási sebesség és a gyorsabb haladás valószínűsége . . . . .	77
5.6. A felépített egyszerű Bayes-háló . . . . .	78

# Táblázatok jegyzéke

1.1. Autonóm járművek fokozatai a SAE szabványa alapján. . . . .	13
3.1. Peer inicializációs információk . . . . .	34
3.2. Peer manőver típusok . . . . .	39
3.3. Címkéző algoritmus paramétereit. . . . .	48
3.4. Neurális háló bemenetei . . . . .	51
5.1. A láz csomópont feltételes valószínűségi táblája . . . . .	71
5.2. A torokfájás csomópont feltételes valószínűségi táblája . . . . .	71
5.3. A rossz közérzet csomópont feltételes valószínűségi táblája . . . . .	71
5.4. A gyengeség csomópont feltételes valószínűségi táblája . . . . .	72

# Irodalomjegyzék

- [1] 'Phantom Auto' will tour city. *The Milwaukee Sentinel*. Google News Archive, 8 December 1926. p. 4. Retrieved 23 July 2013.
- [2] Norman Bel Geddes. *Magic Motorways*. Random House, 1940.
- [3] John Reynolds. Cruising into the future. London: *telegraph.co.uk*, 26 May 2001.
- [4] Reporter rides driverless car. *The Press Courier*. Google News Archive, 7 June 1960. p. 4. Retrieved 21 July 2013.
- [5] Carl Bergenheim, Henrik Pettersson, Erik Coelingh, Cristofer Englund, Steven Shladover, and Sadayuki Tsugawa. Overview of platooning systems. In *Proc. of the 19th ITS World Congress*, Vienna, Austria, October 22–26 2012.
- [6] Kakan Chandra Dey, Anjan Rayamajhi, Mashrur Chowdhury, Parth Bhavsar, and James Martin. Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network - performance evaluation. *Transportation Research Part C: Emerging Technologies*, 68:168 – 184, 2016.
- [7] Hans P. Moravec. *The Stanford Cart and the CMU Rover*, pages 407–419. Springer New York, New York, NY, 1990.
- [8] Hans Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Stanford University, march 1980.
- [9] Dean A. Pomerleau. Alvin: An autonomous land vehicle in a neural network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 305–313. Morgan-Kaufmann, 1989.
- [10] Shell Nutzfahrzeug Studie. Diesel oder alternative antriebe - womit fahren LKW und bus morgen? (2018. május 15.). [http://elib.dlr.de/108867/1/JA\\_1732\\_NFZ\\_Studie\\_Doppelseiten.pdf](http://elib.dlr.de/108867/1/JA_1732_NFZ_Studie_Doppelseiten.pdf).
- [11] ACEA. Commercial vehicles and CO2 (2018. május 15.). [http://www.acea.be/uploads/publications/20101013\\_Commercial\\_Vehicles\\_\\_CO2.pdf](http://www.acea.be/uploads/publications/20101013_Commercial_Vehicles__CO2.pdf).
- [12] Analysis Division Federal Motor Carrier Safety Administration. Regulatory Impact Analysis RIN 2126-AB26 (2018. május 15.). [http://www.fmcsa.dot.gov/sites/fmcsa.dot.gov/files/docs/2011\\_HOS\\_Final\\_Rule\\_RIA.pdf](http://www.fmcsa.dot.gov/sites/fmcsa.dot.gov/files/docs/2011_HOS_Final_Rule_RIA.pdf).

- [13] Dr. Németh Huba. Autonomous Yard Maneuvering előadás (2018. május 5.).
- [14] Welt.de. Robotaxis werden Mobilität sicherer machen - und billiger (2018. május 15.). <http://www.welt.de/wirtschaft/article169634562/Robotaxis-werden-Mobilitaet-sicherer-machen-und-billiger.html>.
- [15] SAE International. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (2018. május 15.). [http://www.sae.org/standards/content/j3016\\_201609/](http://www.sae.org/standards/content/j3016_201609/).
- [16] National Highway Traffic Safety Administration. Preliminary statement of policy concerning automated vehicles. NHTS, Washington, 30 May 2013.
- [17] Sebastian Blanco. Uk driver loses license for 18 months for 'silly' tesla autopilot chauffeur stunt. Forbes, April 2018. [Online; posted 22-May-2018].
- [18] Forbes.com. The level 3 Audi A8 will almost be the most important car in the world (2018. május 15.). <http://www.forbes.com/sites/michaeltaylor/2017/09/10/tthe-level-3-audi-a8-will-almost-be-the-most-important-car-in-the-world>.
- [19] Nick Mockey. A self-driving car in every driveway? solid-state lidar is the key. Digital Trends, March 2018. [Online; posted 01-June-2018].
- [20] United Nations. 19 . Convention on Road Traffic. Vienna, 8 November 1968.
- [21] Tim Higgins Greg Bensinger. Video shows moments before uber robot car rammed into pedestrian. The Wall Street Journal, 21 March 2018. [Online; posted 27-June-2018].
- [22] MIT - Massachusetts Institute of Technology. Moral Machine (2018. május 15.). <http://moralmachine.mit.edu/>.
- [23] MathWorks. MATLAB (2018. május 16.). <http://www.mathworks.com/products/matlab.html>.
- [24] MathWorks. Simulink - Simulation and Model-Based Design (2018. május 16.). <http://www.mathworks.com/products/simulink.html>.
- [25] Kovács Gábor. Gyors prototípustervezés. Irányítástechnika és képfeldolgozás laboratórium 2. mérési útmutató (2018. tavasz).
- [26] Horváth Csaba. Highway Pilot System - HAD Control Overview (2018. április 17.).
- [27] Raymond Brach and Matthew Brach. The tire-force ellipse (friction ellipse) and tire characteristics. In *SAE Technical Paper*. SAE International, 2011.
- [28] dSPACE. Homepage (2018. május 16.). <http://www.dspace.com/en/inc/home.cfm>.
- [29] dSPACE. ModelDesk (2018. május 16.). [http://www.dspace.com/en/inc/home/products/sw/automotive\\_simulation\\_models/produkte\\_asm/modeldesk.cfm](http://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models/produkte_asm/modeldesk.cfm).



- [30] dSPACE. ControlDesk (2018. május 16.). <http://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/controldesk.cfm>.
- [31] dSPACE. MotionDesk (2018. május 16.). <http://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/modesk.cfm>.
- [32] dSPACE. VEOS (2018. május 16.). [http://www.dspace.com/en/inc/home/products/sw/simulation\\_software/veos.cfm](http://www.dspace.com/en/inc/home/products/sw/simulation_software/veos.cfm).
- [33] M. Raissi. Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. *ArXiv e-prints*, January 2018.
- [34] J. Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1?4, 2017.
- [35] Zhang Yi, Yan Fu, and Hua Jin Tang. Neural networks based approach for computing eigenvectors and eigenvalues of symmetric matrix. 47:1155–1164, 04 2004.
- [36] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120:024102, Jan 2018.
- [37] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):589–606, Sept 2010.
- [38] M. Bahram, A. Wolf, M. Aeberhard, and D. Wollherr. A prediction-based reactive driving strategy for highly automated driving function on freeways. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 400–406, June 2014.
- [39] J. C. Hayward. Near miss determination through use of a scale of danger. *Highway Research Record*, 384:24–34, 1972.
- [40] Glaser S, Rakotonirainy A, Gruyer Dominique, and Nouveliere Lydie. An integrated driver-vehicle-environment (i-dve) model to assess crash risks. January 2007.
- [41] P. J. Mills and C. A. Hobbs. The probability of injury to car occupants in frontal and side impacts. In *SAE Technical Paper*. SAE International, 10 1984.
- [42] R. Pride, D. Giddings, D. Richens, and D.S. McNally. The sensitivity of the calculation of  $\Delta v$  to vehicle and impact parameters. *Accident Analysis and Prevention*, 55:144 – 153, 2013.
- [43] Chris Jurewicz, Amir Sobhani, Jeremy Woolley, Jeff Dutschke, and Bruce Corben. Exploration of vehicle impact speed - injury severity relationships for application in safer road design. *Transportation Research Procedia*, 14:4247 – 4256, 2016. Transport Research Arena TRA2016.

- [44] Elaine Wodzin Thomas A. Gennarelli. The abbreviated injury scale. American Association for Automotive Medicine, 2005.
- [45] George Bahouth, Jill Graygo, Kennerly Digges, Carl Schulman, and Peter Baur. The benefits and tradeoffs for varied high-severity injury risk thresholds for advanced automatic crash notification systems. 15:S134–S140, 09 2014.
- [46] Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.
- [47] Jean Roy. From data fusion to situation analysis. In *Proceedings of the Fourth International Conference on Data Fusion*, pages 7–10, 2001.
- [48] J. Desel A. Hermann. Driving situation analysis in automotive environment. In *Proceedings of 2008 IEEE International Conference on Vehicular Electronics and Safety*, pages 216–221, 2008.
- [49] Y. Jia D. Zhao, Y. Guo. Trafficnet: An open naturalistic driving scenario library. In *IEEE Conference on Intelligent Transportation Systems Proceedings*, pages 1–8, 2018.
- [50] Transport Systems Catapult. Taxonomy of scenarios for automated driving. <https://s3-eu-west-1.amazonaws.com/media.ts.catapult/wp-content/uploads/2017/04/25114137/ATS34-Taxonomy-of-Scenarios-for-Automated-Driving.pdf>.
- [51] L. Schnitman A. Sobral, L. Oliveira. Highway traffic congestion classification using holistic properties. In *Computer Graphics and Imaging / 798: Signal Processing, Pattern Recognition and Application*, 2013.
- [52] L. Yan Y. Zhang Y. He S.Gao. Hazardous traffic event detection using markov blanket and sequential minimal optimization (mb-smo). *Sensors (Switzerland)*, 2016.
- [53] Y. Kang. Hazardous driving event detection and analysis system in vehicular networks (heaven): Methodology and field implementation. *TRB 2013 Annual Meeting Paper*, 2013.
- [54] Vahid Mirjalili Sebastian Raschka. *Python Machine Learning Second Edition*. Packt Publishing, 2017.
- [55] Documentation of scikit-learn (2018. október 24.). <http://scikit-learn.org/stable/documentation.html>.
- [56] Mesterséges Intelligencia Elektronikus Almanach (2018. május 17.). [http://project.mit.bme.hu/mi\\_almanach/](http://project.mit.bme.hu/mi_almanach/).
- [57] A Singh, N Thakur, and A Sharma. A review of supervised machine learning algorithms. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1310–1315, 2016.

- [58] P. E. Pintelas S. B. Kotsiantis, I. D. Zaharakis. Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, 26:159–190, 2006.
- [59] Russell Stuart and Norvig Peter. *ARTIFICIAL INTELLIGENCE. A MODERN APPROACH. 2nd Edition. ISBN 0137903952*. Pearson Education Inc., Prentice Hall, Upper Saddle River, New Jersey, 07458, 2003.
- [60] Édelkraut Róbert, Antal Péter, Dobrowiecki Tadeusz, Haidegger Géza, Haidegger Tamás, Héja Gergely, Mészáros Tamás, Pataki Béla, Román Gyula, Strausz György, Tilly Károly, Valyon József, Varga Péter, and Várkonyiné Kóczy Annamária. *Mesterséges intelligencia modern megközelítésben (Második, átdolgozott, bővített kiadás)*. Hungarian Translation Panem Könyvkiadó, Budapest, 2005.
- [61] Dr. Vetier András. *Valószínűségszámítás*. Műegyetemi Kiadó, 2008.
- [62] Katona Gyula Y., Recski András, and Szabó Csaba. *A számítástudomány alapjai*. Typotex, 2002.
- [63] Mesterséges Intelligencia Elektronikus Almanach. Bayes-háló mintapélda (2018. május 17.). [http://mialmanach.mit.bme.hu/feladatok/bayes-halo\\_mintapelda/](http://mialmanach.mit.bme.hu/feladatok/bayes-halo_mintapelda/).