

MISKOLCI EGYETEM
GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR
Logisztikai Intézet



SÜRGŐSSÉGI JÁRMŰVEK MOZGÁSÁNAK SZIMULÁCIÓJA PTV
VISSIM SZOFTVERREL

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

Készítette:

Makó Dávid

Járműmérnöki BSc IV. évfolyam

Neptun kód: IRNKK6

Konzulens:

Dr. Cservenák Ákos

adjunktus

Logisztikai Intézet

Miskolc
2022

TARTALOMJEGYZÉK

1.	Bevezetés.....	4
1.1.	Dolgozat háttere.....	4
1.2.	Dolgozat felépítése	5
2.	Járműkommunikációs és intelligens forgalom irányító rendszerek	6
2.1.	Intelligens mentőrendszer.....	6
2.2.	IoT (Internet of Things).....	6
2.3.	A hálózatba kapcsolt járművek-V2X kommunikáció	8
2.4.	Okos jelzőlámpa rendszer megkülönböztetett járművekhez	9
2.5.	Okos mentőautó rendszer	10
2.6.	Intelligens forgalomirányító rendszer.....	11
2.7.	A megvalósítandó rendszerhez hasonló rendszerek	11
3.	PTV Vissim szoftver bemutatása	14
3.1.	A szoftver ismertetése	14
3.2.	A szoftver a jelen dolgozat szempontjából.....	15
3.3.	A V2V/V2I megjelenítése szimulációban	15
3.4.	API (Application Programing Interface) fejlesztések	15
3.5.	Drivermodel.dll.....	16
3.6.	Drivingsimulator.dll	16
4.	A szimuláció megvalósítása	18
4.1.	A korábbi szimuláció legfontosabb lépései [19]	18
4.2.	Jelenlegi szimuláció ismertetése.....	20
4.3.	Intelligens forgalomirányító rendszer.....	27
4.4.	Továbbfejlesztési lehetőségek	32
5.	Összefoglalás.....	34
6.	Irodalomjegyzék.....	35

Ábrajegyzék:

1. ábra Példa egy IoT rendszerre [10]	8
2. ábra A V2X kommunikáció illusztrálása [11]	9
3. ábra Csomópontokban elhelyezett érzékelők illusztrációja [13]	12
4. ábra Az elkészült szimuláció 3D változata	19
5. ábra Szimuláció futtatása, balra kanyarodás	20
6. ábra Szimuláció futtatása, piroson áthajtás két időpillanata	20
7. ábra A kiválasztott lokáció	21
8. ábra Útpályák létrehozása	22
9. ábra Járműtípus meghatározása.....	22
10. ábra Jármű osztály meghatározása	23
11. ábra Jármű kompozíció beállítása	23
12. ábra Jármű bemenetek.....	24
13. ábra Járművek lehetséges útvonalai (Dél-Észak; Dél-Nyugat irányokban).....	25
14. ábra Járművek lehetséges útvonalai (Nyugat-Dél; Nyugat-Észak irányokban).....	25
15. ábra Járművek lehetséges útvonalai (Észak-Nyugat;Észak-Dél irányokban).....	26
16. ábra Jelzések programozása	27
17. ábra Jelzés vezérlő beállítása	28
18. ábra Detektor beállítása (Lokáció)	28
19. ábra Detektor beállítása (Aktiválás).....	29
20. ábra Forgalmirányító egységek beállítása (Calling)	30
21. ábra Forgalmirányító egységek beállítása (Default)	30
22. ábra Sürgősségi jármű közeledése (Piros-zöld jelzés)	31
23. ábra Sürgősségi jármű közeledése (Piros-narancssárga jelzés)	32
24. ábra Sürgősségi jármű közeledése (Zöld-piros jelzés).....	32
25. ábra VisVAP modul	33

1. BEVEZETÉS

Napjainkban a járműipar jelentős változás felé tart. Alapvető viták vannak azzal kapcsolatban, hogy milyen meghajtás típus lenne az ideális a járművek számára környezettudatosság szempontjából, ugyanis a környezetvédelem lett az egyik fő irány a járműgyártásban, különösen a mai bizonytalan helyzetben ingadozó nyersanyagforrások miatt. A másik legalább ennyire fontos kérdés a közlekedésbiztonság. Abban szinte mindenki egyetért, hogy a biztonságosabb közlekedéshez döntő részben a járműkommunikáció fejlődésén keresztül vezet az út. Ez egy olyan irány, amely hajtástól, felépítéstől, teljesítménytől függetlenül fejlődik minden járműben. Többféle járműkommunikációs rendszer létezik, azonban ezen rendszereket nem könnyű bekezelni, ugyanis nehéz őket egyértelműen különválasztani, mivel eléggé nagy átfedések vannak közöttük. Jelen TDK dolgozat egy sürgősségi járművek figyelésére alkalmas forgalom irányítás szimulációt mutat be.

1.1. Dolgozat háttere

A dolgozat témája egy általam már korábban kidolgozott témához kapcsolódik [1]. Ez a megkülönböztetett járművek közlekedésének nehézségeit mutatja be, illetve eszköz szintű megoldás fejlesztése vette kezdetét a korábbi dolgozatban. Az eszköz a vészhelyzeti járművek közeledésének jelzésére szolgál. A megvalósítás Arduino Nano fejlesztőplatformmal történt. A rendszer működését tekintve két részből állt: egy adó és egy vevő egységből. Az adó egység a megkülönböztetett járműbe lett tervezve, a vevő egység pedig a civil járműbe. Jelen TDK dolgozat rendszer szintű megoldást próbál találni erre a problémára. Az adó egység természetesen a vészhelyzeti járműben maradna, viszont a vevő egység az intelligens forgalomirányító eszközbe kerülne beépítésre. Ez a dolgozat a beépítés technikai hátterére nem tér ki, feltételezi a működő rendszert és az utána következő további logisztikai/forgalomirányítási kihívások megoldása a cél. A kidolgozás közlekedésszimulációs szoftverben valósul meg. Az ötlet háttere az intelligens forgalomirányító rendszerek, amelyen belül most a vészhelyzeti járművekre specializálódott fejlesztés bemutatása a cél. Az alap probléma a [2] cikkben elvégzett kísérlet alapján kerül bemutatásra. Ebben egyértelműen megemlíti a megkülönböztetett járművek magas baleseti arányát. Ennek alátámasztására a német vöröskereszt vészhelyzet elhárító bázisával együttműködve videó felmérést készítettek, amely során egy mentő és egy sürgősségi orvos járművet szereltek fel videó kamerával és GPS adatrögzítővel. Ezek rögzítették a többi közlekedő viselkedését valódi vészhelyzeti utak során és gyűjtötték a vezetési adatokat, például sebesség és útvonal információkat, egy 9 napos periódus alatt. Ezen periódus alatt 21 tipikus vészhelyzeti utat rögzítettek. Átlagosan minden út

hét perces volt és négy piros, illetve három zöld fényjelzésen haladtak át. Az utak során számos kereszteződésen áthaladva, azt a megfigyelést tették, hogy minden piros lámpa 15-30 másodperces késést okoz, illetve az utak 50%-ban volt legalább egy olyan piros lámpás szituáció, ahol egy folyosó létrehozása a sürgősségi jármű részére több mint 30 másodpercebe telt. Átlagosan 2,5 civil sofőr reagált nem megfelelően, amely körülbelül egy perces teljes késést okozott utanként, ráadásul két veszélyes szituációt jegyeztek fel minden út során és egy olyan helyzetet is, amiből majdnem baleset történt. Ezek alapján világosan látszik, hogy egy intelligens forgalomirányító rendszer komoly közlekedésbiztonsági problémát tudna megoldani. Ehhez azonban szükséges az optimális beállításokat megtalálni a megfelelő működéshez, amibe az is beletartozik, hogy egy vészhelyzeti forgatókönyv létrejöttékor se történjen semmilyen veszélyes szituáció vagy fennakadás. A kihívást az a rengeteg lehetőség jelenti, amely egy ilyen vészhelyzet esetén létre jöhet. A megkülönböztetett jármű számos különböző útvonalon, különböző időpontokban, különböző sebességgel haladhat. Továbbá vészhelyzet esetén (fény és hangjelzés működik) a KRESZ-ben egyedi szabályok vonatkoznak rá. Ezzel még több kivételes és „különleges” eset jöhet szóba, amelyeket mind számításba kell venni egy valósághű szimuláció során.

1.2. Dolgozat felépítése

A dolgozat háttereként az első fejezetben bemutatásra kerül egy általam korábban fejleszteni kezdett eszköz, amely jelen dolgozatban bemutatni kívánt intelligens forgalomirányító rendszerek működtetésénél érzékelő funkciót tölthet be. A második fejezetben a jelen témához két legfontosabb járműkommunikációs technológia kerül bemutatásra. Emellett már létező intelligens közlekedési lámpa rendszerekről készített tanulmányok is említésre kerülnek ebben a fejezetben. A harmadik fejezetben kerül ismertetésre a szimuláció készítésére használt PTV Vissim közlekedés szimulációs szoftver. Először egy általános ismertető olvasható, majd egy részletesebb bemutatás a jelen dolgozat témájának szemszögéből. Számos hasznos lehetőség kerül bemutatásra, amelyek felhasználásra kerültek a jelenlegi szimulációban, vagy továbbfejlesztési lehetőségként szolgálnak. A negyedik fejezetben először röviden ismertetésre kerül egy korábban általam elkészített szimuláció. Ezzel lehetőség nyílik a jelenlegi modellezési fejlesztési folyamat egészének áttekintésére. Majd ezt követően a jelenlegi modellezés alapvető beállításai kerülnek bemutatásra, utána pedig az intelligens forgalomirányító rendszer elkészítésének lépései. Ezt követően az eredmények olvashatók, illetve ábrák segítségével a sürgősségi jármű mozgása is tanulmányozható. A fejezet végén a továbbfejlesztési tervek kerülnek bemutatásra. Az ötödik fejezetben egy rövid összefoglalása olvasható a jelenlegi dolgozatnak.

2. JÁRMŰKOMMUNIKÁCIÓS ÉS INTELLIGENS FORGALOM IRÁNYÍTÓ RENDSZEREK

Ebben a fejezetben olyan járműkommunikációs rendszerek kerülnek bemutatásra alkalmazási példákkal együtt, melyek az intelligens forgalomirányításhoz hatékonyan alkalmazhatóak. Emellett több intelligens közlekedési lámpán alapuló technológia is ismertetésre kerül egy olyan cikk alapján, amely a témához kapcsolódó munkákat mutat be. Alapvetően most az intelligens közlekedési lámpák megkülönböztetett járműveket segítő technológiáira kerül a hangsúly.

2.1. Intelligens mentőrendszer

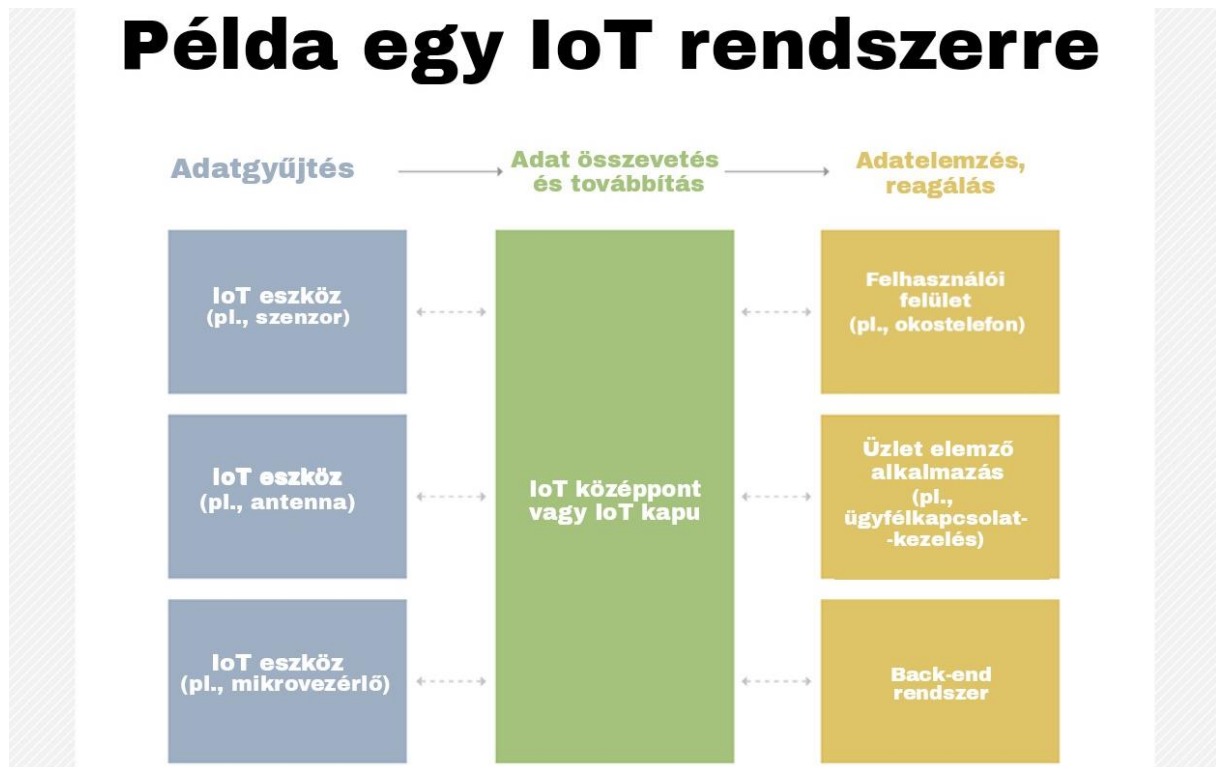
A [3] dokumentum a saját fejlesztésén kívül négy másik megvalósítási lehetőséget mutat be. Az első egy ún. Intelligens mentőrendszer, amely egy RISC mikrokontrollert alkalmazó forgalomirányítási rendszer. Ez a technológia lehetőséget ad a megkülönböztetett járműveket vezető sofőröknek, hogy vezéreljék a közlekedési lámpát. Ezt a fajta vezérlést a mentőautó vezetője egy beviteli eszközön keresztül végzi, kiválasztva azt az utat, amelyen áthalad. Amint a mentőautó vezetője megnyomja a billentyűzetet, az RF (rádiófrekvenciás) jeladó, amely a mentőautóban van, jelet (bináris jel) küld a RF rádiójel vevőnek, amely a közlekedési lámpavezérlőben van. Ez a jel információt tartalmaz a mentőautó helyéről, útvonaláról és a forgalmi torlódásról. Amikor a jelet az RF rádiójel vevő fogadja, dekódolásra kerül, ezután a közlekedési lámpasorozat megszakad, és a vészhelyzeti üzemmód lép működésbe. A vészhelyzeti üzemmódban a közlekedési lámpa elsőbbséget ad a mentőautónak az áthaladáshoz azáltal, hogy zöld fényt kapcsol a mentőautónak a mentősávban, az összes többi sávban pedig a közlekedési lámpák pirosak lesznek. A mentőautó áthaladása után a forgalom és a lámpa visszaáll az eredeti menetrendre.

2.2. IoT (Internet of Things)

A [4]-es Blogban Intelligens forgalom figyelő és –szabályozó rendszerről írnak, amely az IoT olyan verzióján alapul, amely már felhasználja a mesterséges intelligenciát is, ezt nevezik AIoT- nek. A cikkben a Smart Traffic elképzelés kerül előtérbe, amely már az előbb említett rendszeren alapszik és számos előnye van. A rendszer optimalizálja a forgalmat és javítja a biztonságot azáltal, hogy érzékelők, kamerák és útválasztók segítségével dinamikusan állítják be a vezérlő mechanizmusokat. Egy ilyen szoftver, ha képes valósidejű képfeldolgozásra, gyorsan és rugalmasan reagálhat a forgalom mennyiségének, sűrűségének és kapacitásának változásaira. Például a közlekedési lámpák, elsőbbséget adnak, priorizálják a szabad és kötöttpályás tömegközlekedést, vagy éppen a dinamikus sávmenedzsment alapján,

szabaddá teszik az autósok számára a buszsávokat, akkor, amikor a busz, azt nem használja. További előnye, -amely a jelen dolgozatban megvalósítani kívánt rendszerhez szorosan kapcsolódik-, a sürgősségi járművek priorizálása, melyet a forgalom intelligens szabályozása tesz lehetővé. A [5]-as folyóirat cikkében az „okos” közlekedési lámpákról írnak részletesen. Ezeknek az intelligens jelzőlámpáknak a ciklus beállításait háromféleképpen csoportosítják: a forgalom sűrűsége szerint, megkülönböztetett járművek várható útvonala alapján, illetve a gyalogosok érdekeit figyelembe véve. A cikk e három helyzet szemszögéből elemzi a közlekedési lámpatechnológia fejlesztésében alkalmazott rendszereket, technológiákat (melyben nagy szerepet kap az IoT-AIoT technológia), valamint a jövőbeni intelligens közlekedési lámpatechnológia fejlesztését. Szintén a [5]-as folyóiratban esik szó az okos jelzőlámpák jelentőségéről a megkülönböztetett járművek vonatkozásában. A technológia lényege, hogy minden jármű információt küld egymásnak (a sebességükről), a zöld lámpa időtartamát a jármű kereszteződéstől való távolsága és a becsült érkezési idő alapján számítják ki. A járművek számát kamera segítségével számolják és a sürgősségi járműveket érzékelők segítségével különböztetik meg. Egy másik megoldást is említene, melyben a segélyszállító járművek észleléséhez és a vészhelyzeti protokollok aktiválásához RFID címkéket használnak. Normál működési feltételek mellett megmarad a normál jelsorozat, azonban mikor a vészhelyzeti protokoll aktiválódik, az eszköz állapota megváltozik, majd a vészhelyzet elmúltával visszatér az alap állapotba. A [6]-es dokumentumban egy, olyan vezérlő rendszert mutatnak be, mely egy szoftver által meghatározott hálózaton (SDN) és az IoT-n alapul. Vészhelyzet esetén a közlekedési kamerák segítségével módosítja a közlekedési lámpák ciklusát úgy, hogy az SDN-vezérlő kapcsolatba lép a hálózattal, hogy forgalmi adatokat gyűjtsön és ezek alapján rendezze a közlekedési lámpák ciklusát. A [7]-ös folyóirat cikkében is az RFID technológiát használják a közlekedési jelzések vezérlésére a járműsűrűség mérése által. Olyan RFID alapú technológiát javasolnak, amely megszünteti a torlódásokat zsúfolt órákban és védett sávokkal látja el a sürgősségi járműveket. Ennek a rendszernek a fő célja a forgalom hatékony irányítása és a segélyszállító járművek megfelelő haladásának segítése. A sürgősségi járművek akkor bocsátanak ki jelet, ha bizonyos távolságon belül RFID jelet észlelnek. Amikor közeledik a megkülönböztetett jármű, a forgalomserver értesíti az Arduino-t a csomópontban, majd ezt az Arduino által szerzett jelet, kiegyenlíti az RFID olvasóból kapott RFID-jel. Ez megerősíti, hogy a jármű vészhelyzetben van-e vagy sem. A zöld lámpára való átkapcsolás igazolásáig, a megkülönböztetett jelzésű jármű szabad utat kap. Amennyiben a sürgősségi jármű megkapja az RFID-jelet az Arduino-tól, akkor a várható útvonalán végig zöld jelzést fog kapni a jelzőlámpáknál. A forgalmi információt elküldik a szervernek, hogy a

felhasználók a weboldalak segítségével meghatározhatják a forgalmi feltételeket. Az 1. ábrán egy IoT rendszerre látunk példát.



1. ábra Példa egy IoT rendszerre [10]

2.3. A hálózatba kapcsolt járművek-V2X kommunikáció

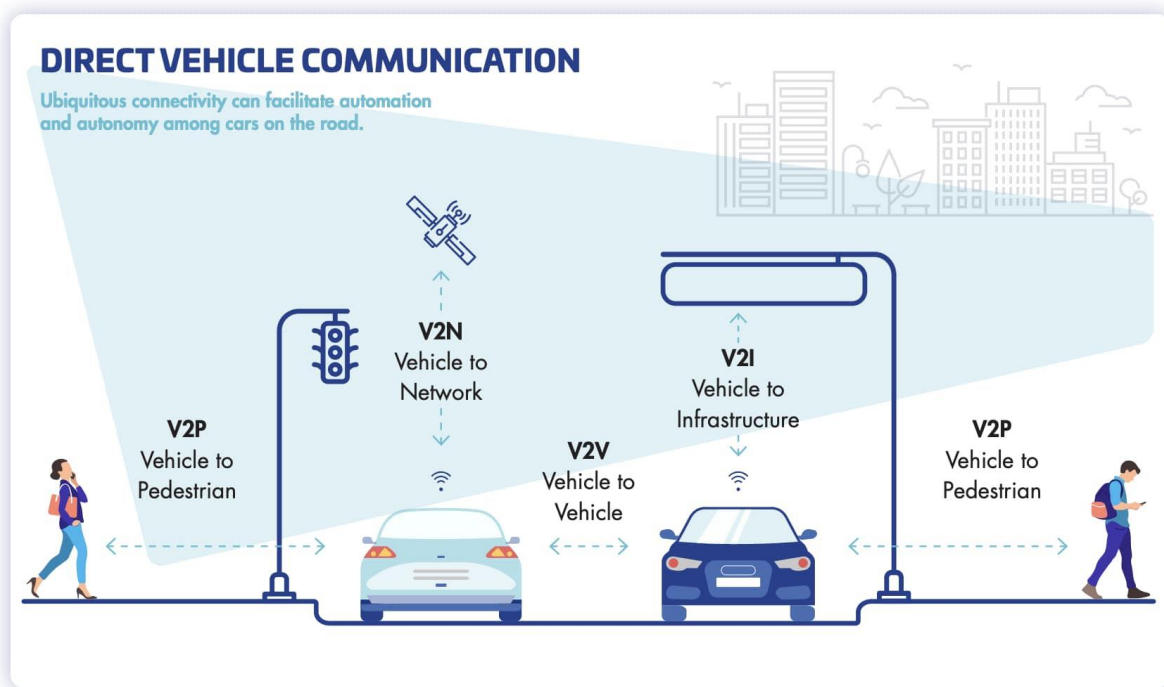
Típusai [9]:

- V2N: Vehicle-To-Network=Jármű és hálózat közötti kommunikáció
- V2V: Vehicle-To-Vehicle=Jármű és másik jármű közötti kommunikáció
- V2P: Vehicle-To-Pedestrian=Jármű és gyalogos közötti kommunikáció
- V2I: Vehicle-To-Infrastructure=Jármű és infrastruktúra közötti kommunikáció

A különböző típusokat a 2. ábrán lehet megfigyelni [9].

A [8] folyóirat cikk a V2V hálózaton belül említ meg egy olyan alkalmazást, melynek Környezet függő Biztonságos Vezetés navigátor a neve (CASD). A sofőrök számára biztonságosabb vezetést tud biztosítani, azáltal, hogy segít felismerni a veszélyes helyzeteket, illetve akadályokat.

A [8] dokumentum említi meg a V2I hálózatok használati eseteit. Szó esik egy navigációs szolgáltatásról, melynek Ön Adaptív Interaktív Navigációs Eszköz a neve (SAINT) és V2I hálózatot használ a közúti forgalom globális optimalizálása érdekében. A továbbfejlesztett változat (SAINT+) képes megadni különböző útvonalak közül a leggyorsabbat a baleseti helyszín felé, a megkülönböztetett járművek számára, mindeközben civil járműveket is képes ellátni hatékony kerülő utakkal.



2. ábra A V2X kommunikáció illusztrálása [11]

2.4. Okos jelzőlámpa rendszer megkülönböztetett járművekhez

A [12] cikkben egy fejlesztés megvalósításáról írnak, melyhez egy többszintű architektúrát használnak az okos forgalomirányító rendszerhez, ami egy fizikai rétegből, egy köztes szoftverből és egy alkalmazási rétegből áll. A köztes szoftver egy adapterrétegből és egy ideiglenes adatbázisból áll, amely az adatokat a Google Maps-től kapja. Ethernet chip szükséges a csatlakoztathatósághoz, LED-es lámpák és Arduino a fények vezérléséhez. Az adapterréteg úgy működik, mint egy ügyfélsomagolás a különböző STLS-technológiák, amelyek információt szolgáltatnak a rendszer számára, mint például a GPS és a WiFi stb. Eszközmeghajtóként szolgál, amely lehetővé teszi a különböző STLS technológiák zökkenőmentes működését. a middleware-rel. A middleware több funkcióért felelős, többek között 1) a nyers adatok lekérdezése és feldolgozása a következőktől a rendszerben rendelkezésre álló érzékelőkből, 2) az adatok előfeldolgozásáért, valamint az esetleges konfliktusok feloldása, a kiugró értékek elutasítása és a következtetések levonása, és 3) az adatok tárolása, megosztása, terjesztése és közzététele. az STLS rendszerben. Az STLS adatbázis egy ideiglenes adatbázis, amely a Google Maps és az MQTT üzenetkövetítő által szolgáltatott friss adatokat tartalmazza. A Google Maps adatait a legrövidebb útvonal és az útvonalon lévő közlekedési lámpák helyének megkeresésére használjuk. Az MQTT protokollt az útvonalon lévő egyes közlekedési lámpák érkezési idejének közzétételére és ezen információknak az összes előfizető közlekedési lámpához való eljuttatására használják. Az

alkalmazási réteg egy mobileszközön futó Android alkalmazásból áll. Az Android-alkalmazás lehetővé teszi a felhasználó számára a kiindulási hely és a végső célállomás kiválasztását. A Google térképek segítségével meg lehet találni a legrövidebb utat a forrástól a célállomásig, valamint az útvonalon lévő összes közlekedési jelzőtábla helyzetét. Az Android-alkalmazás ezután elküldi az egyes közlekedési lámpák érkezési idejét a megfelelő jelzőlámpa-irányítóknak. Végül, amikor egy segélyhívó jármű megérkezik a célállomás felé vezető útvonal közlekedési lámpáihoz, azokat a kereszteződésben lévő más jelzésekkel való ütközés nélkül nyitva találja.

2.5. Okos mentőautó rendszer

Szintén a [3] cikk említi meg az intelligens mentőautó-rendszert használnak arra, hogy engedélyt biztosítsanak a bármely sürgősségi mentőautónak az összes piros lámpa zöldre váltásával a sürgősségi jármű útvonalán. A rendszer a GPS, a globális mobilkommunikációs rendszer (GSM) használatával valósul meg, és ZigBee technológiával, a rendszer néhány percenként elküldi a beteg paramétereit a kórháznak, hogy az figyelemmel kíséri a beteg állapotát. A mikrokontroller egység a mentőautó vezérlőhelyiségéhez van csatlakoztatva, hogy elküldi a részleteket a központ és a közlekedési jelzőtábla számára. A rendszer ARM Cortex-M32-t használ, mint interfész a következőkkel közlekedési jelző és mentőautó részleggel, az ARM Cortex-M3 rendszerfejlesztéseket és magasabb szintű a támogató blokk integrációjának magasabb szintjét. Ez az intelligens rendszer alacsony költségű rendszer, a ZigBee technológiának köszönhetően, ami olcsóbb, mint más WPAN ok, a GPS mindenki számára szabadon hozzáférhető és a GSM. Az elsődleges cél a következő felismerni a vészhelyzeti járművet, és nyomon követni annak helyzetét, hogy vezeték nélküli jelet adjon a vészhelyzeti járműnek. A szokásos technológiák képfeldolgozást használnak a vészhelyzeti jármű felismeréséhez, de ez a technológia az időjárási körülmények befolyásolják, ami megakadályozza a vészhelyzeti jármű felismerését. Így ez az intelligens rendszer ZigBee transzpondert és vevőt használ, amely minden időjárási körülmények között működik. Az oldalon az intelligens rendszerben nagyfrekvenciás olvasóra van szükség a nagy hatótávolság biztosításához. ZigBee transzpondert kell a jármű műszerfalába kell beágyazni. A rendszer három részből áll: mentőautó, központ és a közlekedési jelzőrendszer. Az intelligens mentőautó elküldi a helyzetét és a beteg paramétereit a központnak. A központ elküldi a legközelebbi kórházhoz vezető utat az intelligens járműnek, a mentőautó kiválasztja az utat, és minden közlekedési jelzőtábla ebben az irányban zöld lesz.

2.6. Intelligens forgalomirányító rendszer

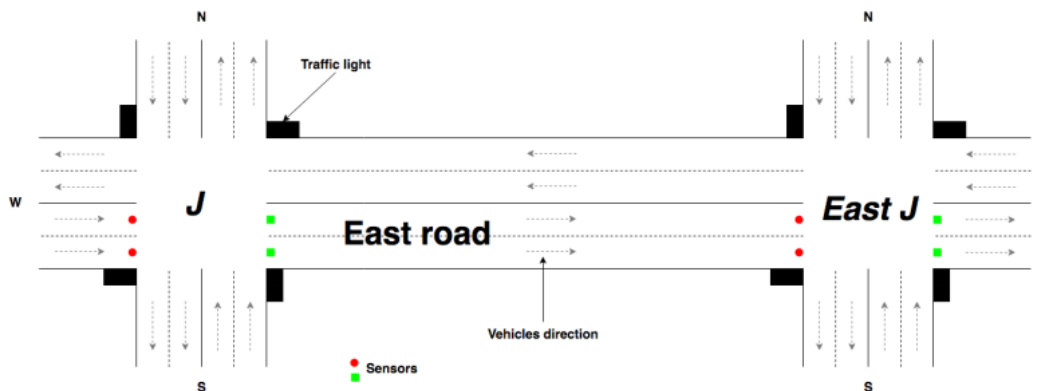
A [3] cikkben lehet olvasni egy olyan intelligens forgalom irányítórendszeréről, amely könnyű áthaladást biztosít a megkülönböztetett járműveknek. A rendszer három részből áll, és a rendszerben használt technológiák a következők: ZigBee, rádiófrekvenciás azonosítás (RFID) és globális mobilkommunikációs rendszer. (GSM). Az első rész az egy adott útvonalon áthaladó járművek számának számlálásával foglalkozik, egy adott időpontban, valamint az útvonal zöld jelzésének időzítésével. Az RFID-címkét minden egyes járműben elhelyezik, így amikor az az RFID-olvasó megszámolja az áthaladó járművek számát. A módszertan az alábbiak megismerésére a zöld jelzés időtartamának és a járművek számának ismerete a következő módon történik: ha a járművek száma több mint 10, a zöld lámpa időtartama 30 másodpercre van beállítva. Ha a szám 5 és 9 között van, akkor a zöld lámpa időtartamát a zöld lámpa időtartama 20 másodpercre, ha pedig 5-nél kevesebb, akkor 10 másodpercre. A második rész a mentőautók kiürítését kezeli, minden mentőautó rendelkezik egy ZigBee adómodullal és a ZigBee-vevő minden egyes közlekedési csomópontban van implementálva. A ZigBee vevő várja a jelzést egy ZigBee-adóról, amely minden egyes mentőautóban van. mikor a ZigBee megkapja ezt a jelet, a közlekedési lámpa zöldre vált. A közlekedési lámpa ismét pirosra vált, amint a mentőautó áthaladt. A harmadik rész egy módszert mutat be a lopott jármű észlelésére, az RFID-olvasó érzékeli a lopott járművet, a modul összehasonlítja az RFID-olvasó által leolvasott egyedi RFID-címkét a lopott RFID-kel, amelyeket az RFID-olvasóban tárol a rendszer. Ha található egyezés, a jelzőlámpa 30 másodpercre pirosra vált. Emellett SMS-t küld a rendőrségnek.

2.7. A megvalósítandó rendszerhez hasonló rendszerek

Ebben az alfejezetben egy olyan dolgozat kerül bemutatásra, amely általánosan ír a közlekedésszimulációs szoftverekről és azok működéséről. A szimulációk mögött lévő matematikai háttérrel fed fel. Emellett egy cikk is említésre kerül, amelynek szerzői ugyanazt a szoftvert használják, mint az általunk bemutatni kívánt szimulációhoz használt program.

A [13] cikkben szoftveres szimulációk kerülnek bemutatásra, intelligens forgalomirányító technológiák felhasználásával. A cikk egyértelműen kitér a megkülönböztetett járművek szempontjából való jelentőségére az intelligens forgalomirányító eszközöknek és így fogalmaz: „A sürgősségi járművek, például a mentőautók és a tűzoltóautók késése kétségtelen, a forgalmi torlódások legkritikusabb következményei közé tartozik, mivel jelentős veszteségeket okozhat vagyontárgyak és emberi életek elvesztéséhez vezethet.” A cikk bemutat egy a szerzők által javasolt Adaptív Közlekedési Lámpa Vezérlőrendszer részletes működését (ATLCS), amelynek célja a városok forgalmi torlódásainak csökkentése a délutáni

csúcsidőszakokban. A napnak ebben az időszakában a városközpontban dolgozó emberek többsége általában elhagyja a munkahelyét, és hazamegy. A rendszer a városközpontból a főutakon keresztül távozó járműveket előnyben részesíti azáltal, hogy a jelzőlámpa-irányítók szinkronizálják az adott irányba közlekedő járműveket, hogy az abban az irányban közlekedő járműveknek ne kelljen olyan gyakran megállniuk. Ez csökkenti a „stop and go” idejét, és ennek eredményeképpen több jármű hagyhatja el a várost. Azáltal, hogy ezek a járművek elsőbbséget kapnak, csökkenne a városközpontban közlekedő járművek száma, ami viszont enyhítené az ottani torlódásokat. A hatékony ATLCs sikeres kiépítésének egyik fő követelménye az utakon közlekedő járművek számának pontos becslése. Ez többféleképpen is megvalósítható, például induktív hurok, mágneses érzékelők, magnetométerek vagy akár kamerák segítségével. Miután a szerzők megvizsgálták az előnyeiket és hátrányaikat, a magnetométeres érzékelőket választották a rendszerükben alkalmazandó fő érzékelési technológiának. Ezek az érzékelők a Föld mágneses terének változását érzékelve képesek érzékelni a felettük áthaladó járművek jelenlétét. Számos kiemelkedő tulajdonságuk van, például könnyen működtethetők akkumulátorral, és teljesítményük rossz időjárási körülmények között sem romlik.



3. ábra Csomópontokban elhelyezett érzékelők illusztrációja [13]

Ebben a tanulmányban csak a két sávú utakat vesszük figyelembe, ezért az úthálózatok architektúrája csomópontonként négy érzékelőt igényel, amint azt a 3. ábra mutatja, amely a telepített érzékelőket ábrázolja.

A jármű helyzetét leíró egyenletet a sebességekifejezés időbeli integrálásával kaphatjuk meg:

$$x(t) = x_0 + v_0 \times t + \frac{1}{2} \cdot a \times t^2 \quad (t \leq t_{vL}) \quad (1)$$

$$x(t) = x_{vL} + v_L \times (t - t_{vL}) \quad (t > t_{vL}) \quad (2)$$

ahol x_0 a jármű helyzetét v_0 a jármű sebességét jelöli $t=0$ időpillanatban, x_{vL} a jármű által megtett távolság a $t=t_{vL}$ időpillanatig, t_{vL} az az időpillanat amikor a jármű eléri az adott útra vonatkozó sebességhatárt. A $t=t_{vL}$ időpillanatig megtett távolságot ki lehet fejezni a (1) és (2) egyenletekből.

$$x_{vL} = x_0 + v_0 \times t_{vL} + \frac{1}{2} \cdot a \times t_{vL}^2 \quad (3)$$

Ezek az összefüggések segítenek minél pontosabban meghatározni egy szimuláció során a valós forgalmi szituációt.

A [14] cikk is határozottan kiáll amellett a tény mellett, hogy a megkülönböztetett járműveknek városi környezetben rendkívül nehéz dolguk van az elsőbbségi jog megszerzésével kapcsolatban. A sziréna és fényjelző berendezés használata közben áthaladhatnak a piros forgalomirányító jelzésen, azonban ez nagyon veszélyes szituációkhoz vezethet, ahogyan az már a jelen dolgozat elején bemutatásra került. A [14] cikkben bemutatott projekt szimulációját a PTV Vissim szoftver segítségével valósították meg (jelen dolgozatban bemutatott szimuláció is ezzel a szoftverrel került megvalósításra). A szimuláció során a civil járművek mellett megkülönböztetett járműveket határoztak meg és folyamatos zöld hullámot biztosítottak nekik, hogy jelentős lassítás nélkül tudjanak áthaladni egy-egy kereszteződésen. A [14] irodalom alapján: „A szoftver különböző mértékben rangsorolja a forgalmi módokat. Ha egy vészhelyzeti jármű V2X kommunikáción keresztül jelzi, hogy szabad áthaladásra van szüksége, a szoftver ennek megfelelően kezeli a közlekedési lámpákat, lehetővé téve a jármű számára, hogy lassítás nélkül áthaladjon a kereszteződésen - a többi közlekedőt a lehető legkisebb mértékben érintve”.

3. PTV VISSIM SZOFTVER BEMUTATÁSA

3.1. A szoftver ismertetése

A [15] cikk alapján a Vissim mikroszkopikus szimulációs szoftver értékes szimulációs adatokat szolgáltat egyszerű csomópontoknál és kiterjedt hálózatoknál egyaránt. A VisVAP modul segítségével hurokdetektoron alapuló logikák valósíthatók meg if-else szerkezetekben. A .dll programozással tetszőleges jelzési programok állíthatók elő. A VISSIM COM-felületén keresztül a közlekedési hálózat majdnem minden objektuma irányítható (például a járművek, utak és természetesen a jelzőlámpák is), így ezzel az eszközzel előremutató elvek is vizsgálhatók a városi forgalomirányításban. A VISUM szoftver makroszkopikus közlekedési modellt használ. Teljes városok forgalomlefolysa vizsgálható vele eltérő beavatkozási esetekben. Számos közlekedési igény modellezésére szolgáló eljárás áll rendelkezésre a szoftverben. Több forgalomráterhelési eljárás használható mind az egyéni, mind a közforgalmú közlekedési szektorra. A szoftver fel van készítve GIS adatok kezelésére, így a közlekedési hálózat létrehozása egyszerűbbé és pontosabbá válik.

A [16] dokumentumban szintén a PTV Vissim és a PTV Visum szoftverről lehet részletes ismertetőt találni és az oktatásban való szerepére is nagy hangsúlyt fektet. A [16] irodalom alapján: „A közlekedésszimulációs szoftverek, mint amilyen a PTV Vissim is, lehetőséget adnak rá, hogy a lehető legnagyobb részletességgel lehessen szimulálni a forgalmi viszonyokat a közlekedési hálózat egy adott metszetében. Ez tehát a gyakorlatban azt jeleni, hogy akár métereken belüli precizitással modellezhetők a különböző kereszteződések (beleértve a körforgalmas kereszteződéseket is), forgalmi csomópontok, nagyobb parkolóterületek, a tömegközlekedés és a gyalogos közlekedés lokális szintű megoldásai, az egyes közlekedési ágazatok helyi-szintű kereszteződései stb.” Míg a közlekedésszimulációs szoftverek elsősorban a forgalom minél pontosabb mikro-szintű szimulációjára szolgálnak, addig a közlekedéstervező szoftverek az átfogó közlekedési hálózatok tervezéséhez nyújtanak segítséget. Az ilyen típusú szoftverek általában több összetevős és többszintű, komplex matematikai modellekre épülő eljárások alkalmazásával valósítják meg a nagy kiterjedésű közlekedési rendszerek forgalmi viszonyainak pontos és részletes modellezését. A cél általában az, hogy minél nagyobb pontossággal lehessen előre jelezni a forgalmi viszonyokat, illetve a veszélyes szituációk minimálisra csökkentése, ami alapvető támpontként szolgálhat a közlekedési hálózat jövőbeni fejlesztéséhez.

3.2. A szoftver a jelen dolgozat szempontjából

A PTV VISSIM forgalomszimulációs szoftverben fejlesztésként a V2V, illetve V2I kommunikáció szimulációba való implementálása lenne a cél. Olyan forgalmi helyzet kerülne szimulálásra, mely során egy vészhelyzeti jármű a megkülönböztetett jelzését használva közeledne egy forgalomirányító jelzőlámpa rendszer felé és az a ciklusát megváltoztatva minden esetben a sürgősségi járműnek adna elsőbbséget a zöld jelzés biztosításával, illetve a megkülönböztetett jármű útját keresztező utakon haladó járműveknek piros fényjelzés kapcsolásával. A [17] dokumentum alapján a VISSIM szoftver alapértelmezetten nem tartalmaz implementálható V2V/V2I eszközöket. Jellemzően ezek az interakciók külső közlekedési és kommunikációs modulok fejlesztésével történnek, amelyek együttesen szimulálják a V2V/V2I kommunikációt. Például a [18] dokumentumban a kutatás Pythonban fejlesztett különböző forgatókönyveket használ a V2V/V2I kommunikáció szimulálására a VISSIM szoftverben. A tesztelési szimuláció által demonstrálásra került a V2V üzenetek terjedése egy közlekedési helyzetben, a figyelmeztető jelzést küldő összekapcsolt járművek változó elérési rátáihoz viszonyítva. A hangsúly azon volt, hogy az összekapcsolt járművek elérhetősége egy forgalmi helyzetben, hogyan tudja pl. csökkenteni a forgalom sebességét a baleset közelében, az út-biztonság javítása érdekében.

3.3. A V2V/V2I megjelenítése szimulációban

A [17] dokumentumban található információk a PTV VISSIM 11 verziójától elérhetőek teljeskörűen. Az egyszerűbb összedolgozás néhány következményét és az automatizált járművek kommunikációs funkcióit lehetséges szimulálni a PTV VISSIM új funkcióival és a megfelelő beállítások használatával. Azonban ahhoz, hogy komplexebb összedolgozás és kommunikáció szimulálható lehessen, szükség lesz további lépésekre. Valamelyik elérhető interfészt kell használni hozzá (Pl. COM interfész, drivermodel.dll interfész vagy drivingsimulator.dll interfész).

3.4. API (Application Programming Interface) fejlesztések

A PTV VISSIM-ben három API található. Ezek közül kettő jelentős fejlesztésen esett át. Az API-ok felhasználási lehetősége:

- Saját algoritmussal rendelkező járművek szimulálása PTV VISSIM-ben
- Csatlakozás és információ csere valamilyen járművön kívüli szimulált eszközzel (Pl. érzékelő szimulátor)
- Kommunikáció és összedolgozási stratégia vagy bármilyen funkció szimulálása, amely nem érhető el közvetlenül a PTV VISSIM GUI-jában.

Az API-t széleskörben használják autógyártók és kutatók egyaránt, akik a saját irányítási algoritmusukat fejlesztik. Az API használatához szükséges programozói ismeret. Mindkét említett interfész teljes leírása megtalálható a dokumentumokban, melyek a szoftver telepítése során kerülnek a számítógépre.

3.5. Drivermodel.dll

A külső Driver Model DLL interfésszel lehetőség van a szoftverben található közlekedési magatartás megváltoztatására és helyette felhasználó által meghatározott magatartás használatára néhány, vagy az összes jármű részére a szimulációban. A felhasználó által meghatározott algoritmust be kell illeszteni a DLL-be C/C++ programozási nyelven megírva. A szimuláció során a VISSIM behívja a DLL kódot minden egyes érintett jármű esetében, minden időbeni lépés során a szimulációban, hogy meghatározza a jármű magatartását. A VISSIM elküldi a jármű és környezete aktuális állapotát a DLL-nek és a DLL monitorozza a gyorsulását/lassulását és a mellette lévő jármű magatartását is (főleg sáv váltás esetén), majd a jármű frissített állapotát visszaküldi a VISSIM-nek. Ez a külső közlekedési modell minden jármű típushoz külön aktiválható a „Use external driver model” paranccsal, emellett szükséges választani egy közlekedési modell DLL fájlt is. Amennyiben ez a beállítás megfelelő, akkor az adott járműtípus összes járművének közlekedési magatartása a kiválasztott DLL alapján kerül kiszámításra. A vissim.exe könyvtárban létre kell hozni egy alkönyvtárat (DriverModelData\), a VISSIM indulásakor megjelenő figyelmeztető üzenet elkerülése érdekében. A következő fejlesztések kerültek megvalósításra:

- A külső Driver Model DLL-ek most már több szálon futó szimulációkban is alkalmazhatók, amennyiben az összes DLL támogatja a többszálon futást
- Ha a DLL kéri, a Vissim elküldi az összes olyan közeli jármű adatait, amelyet az ego-jármű (a szimuláció során elsődlegesen megfigyelt jármű) lát az aktuális közlekedési magatartás szerint
- Felhasználó által meghatározott jármű tulajdonsági értékek küldhetők a DLL-nek és módosíthatóak a DLL segítségével.
- A közeli járművek elejének és végének a koordinátáit a VISSIM elküldi a DriverModel.DLL-nek
- Az ego jármű aktuális sávjának vonallánca szintén elküldésre kerül a DLL-nek

3.6. Drivingsimulator.dll

A PTV VISSIM kiegészítő modulja a „Driving Simulator Interface” lehetővé teszi a VISSIM csatlakoztatását egy közlekedési szimulátorhoz. Ez a szimulátor lehet egy ember által

használt hardware vagy egy szoftver részlet, amely az autonóm jármű algoritmusát képviseli. A VISSIM lehetővé teszi, a körülvevő forgalom szimulátorban való megjelenítését és a szimulátor visszaküldi az aktuális pozícióját és orientációját a szimulált járműveknek, kerékpárosoknak, gyalogosoknak. A járművek és gyalogosok a VISSIM hálózatban ugyanúgy reagálnak ezekre a szimulált adatokra, mint az összes többi jármű és gyalogos a mikroszkopikus szimulációs modellben. Ráadásul, a VISSIM közlekedési jelállapotokat küld a DS szimulátornak vizualizációra és a DS szimulátor képes detektorokat beállítani a VISSIM-ben kifejezetten a jelzés befolyásolására. A DS szimulátornak nincs szüksége a VISSIM adat modell ismeretére. A DS-nek saját világmodellel kell rendelkeznie (a szimulációhoz és a vizualizációhoz). Mivel minden jármű és gyalogos pozíciója „cartesian” világkoordinátákra (x/y/z) cserélődik ki, a DS-nek képesnek kell lennie ilyen koordináták megadására/használatára, valamint a kétoldali hálózatok koordinátáinak megadására/használatára és az (Vissim / DS) oldalainak pontosan meg kell egyezniük. A következő fejlesztések kerültek megvalósításra [17]:

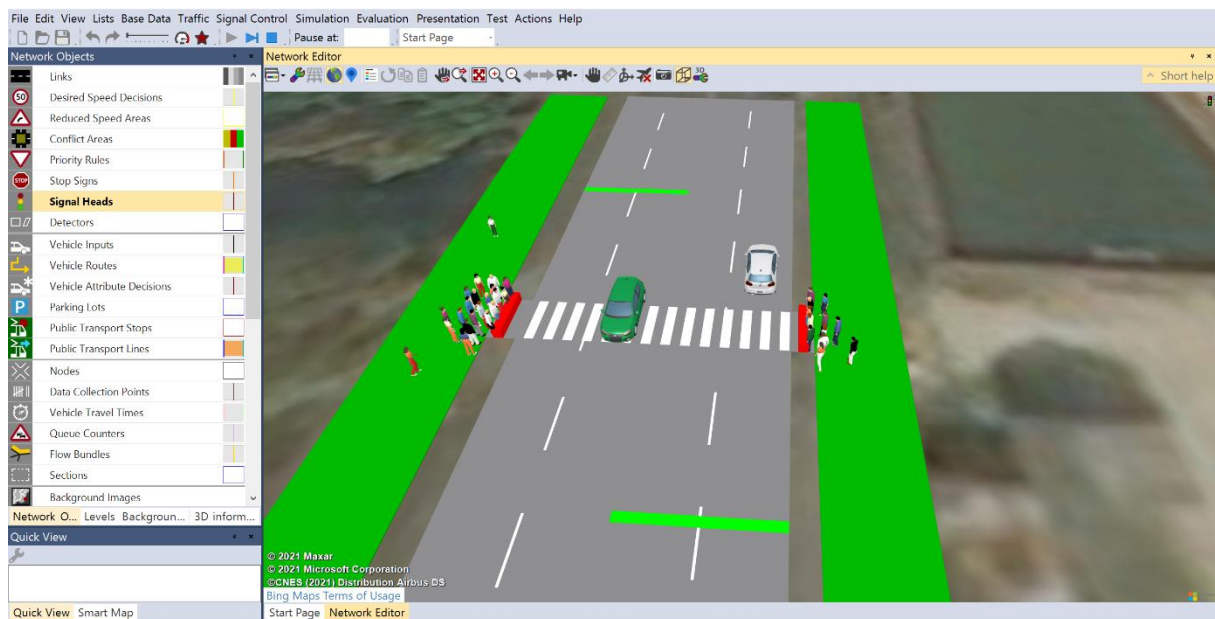
- Automatikus beszúrása a Vissim jármű/gyalogos világkoordinátáknak, orientációknak és sebességeknek a VISSIM időlépések között, amennyiben a szimulátor magasabb képkockasebességről informálja a VISSIM-et, mint a VISSIM szoftver szimulációs felbontás a VISSIM_Connect() hívásában.
- Opcionálisan a gyalogos szimulátor most is átküldhető a Vissimnek. A gyalogos típust a hálózat beállításában kell kiválasztani az új "Vezetésszimulátor" lapon. A jelölőnégyzet, a közlekedésszimulátor-interfész aktiválására szolgáló jelölőnégyzet, a járműtípus kiválasztására szolgáló négyzet és a szimulátor járművekre vonatkozó választási lehetősége is átkerült oda.
- A DS szimulátor és a Vissim között kicserélhető objektumok maximális száma minden egyes típusnál a szimulátor által a VISSIM_Connect() hívásában állítható be.
- A DS szimulátor a VISSIM_Connect() hívásában átadhat egy maximális láthatósági sugarat a Vissimnek. A Vissim járműveket és gyalogosokat csak akkor adja át a szimulátornak, ha ezen a sugáron belül vannak a szimulált jármű vagy gyalogos elülső részének középpontjától számítva

4. A SZIMULÁCIÓ MEGVALÓSÍTÁSA

Ennek a fejezetnek az első alpontjában röviden bemutatásra kerül, az előző dolgozatomban készített szimuláció legfontosabb lépései. Ezzel kicsit nagyobb rálátást biztosítva arra a folyamatra, ami ehhez a forgalom szimulációhoz és annak eredményeihez vezetett. Bár a két modell alapvetően teljesen más (a lokáció sem egyforma), mégis egy folyamaton lehet végig haladni a segítségükkel. A második alpontban már a jelenleg elkészített forgalom szimuláció és annak eredményei kerülnek bemutatásra. A jelenlegi modellben a Búza téren található Szeles utca és 26-os főút kereszteződése lett szimulálva. A fejezet harmadik alpontjában a továbbfejlesztési lehetőségek lesznek ismertetve.

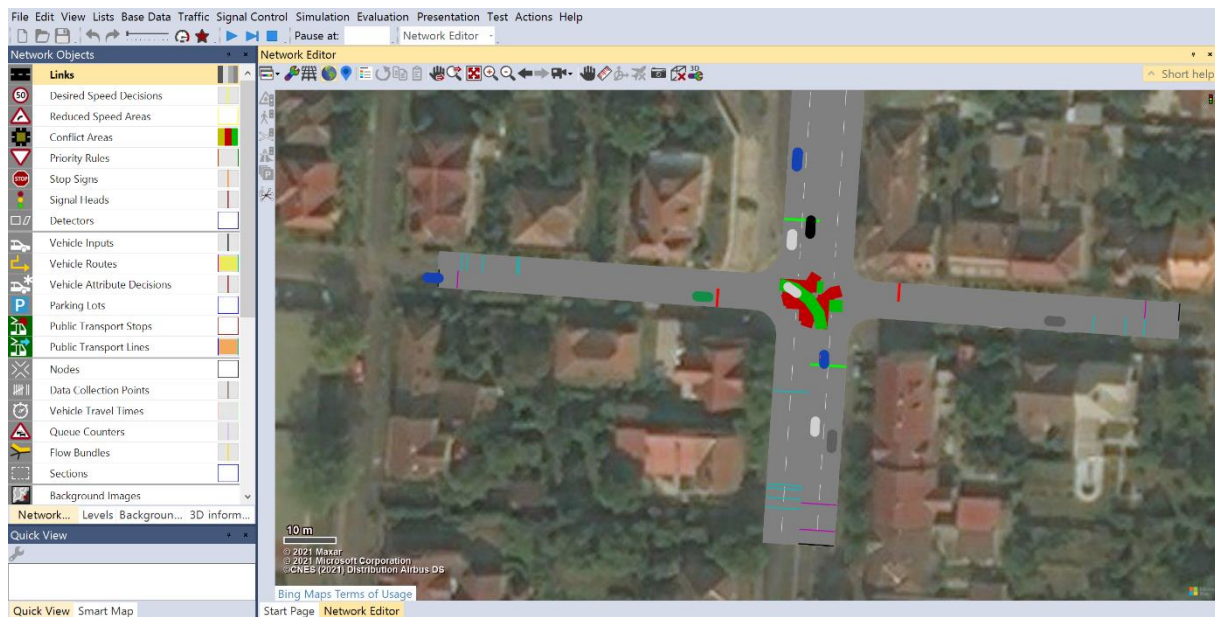
4.1. A korábbi szimuláció legfontosabb lépései [19]

A korábbi szimulációhoz lokációként Miskolcot választottuk, azon belül pedig a Szilágyi Dezső 4 sávós főutat, illetve az azt keresztező Lévy József utcát és a Farkas Jenő utcát. Ezek után a fő útpályák létrehozása volt a feladat. A „Links” paranccsal tudunk fizikai pályát létrehozni. Meghatározható milyen hosszú szakaszon szeretnénk a pályát, illetve hány sávós legyen. A szimuláció során a főút pálya a Farkas Jenő utcától korábbról indul és a Borsod Volán stadion közepéig tart. Az útpálya létrehozásakor a két sávós lehetőség lett beállítva, két irányban. Az így kialakított négy sávós főutat keresztező mindkét út fizikai pálya szakasza egy sávósra lett beállítva két irányban. A lámpacsoport már létrehozásra került a gyalogos átkelőhelyhez, azonban a gyalogos járda fizikai pályája még hiányzott ebben a fázisban, így azzal folytattuk a munkát. Az „Areas” parancs segítségével a Borsod Volán stadion közelében a négy sávós főút két oldalán lett létrehozva a gyalogos járda fizikai útpályája. Az elkészült járda útpályára gyalogosok kerültek elhelyezésre a „Pedestrian Inputs” paranccsal. Négy pontba kerültek elhelyezésre a tervezett gyalogátkelőhely mindkét oldalára, így egyenletes eloszlásban érkeznek a gyalogosok az átkelőhelyhez. Az óránkénti gyalogosok száma 500-ra lett beállítva. A gyalogosoknak két lehetséges útvonal lett beállítva, melyek az egyik oldalról a másik oldalra való áthaladás és fordítva. Úgy lett beállítva, hogy mind a kettő eset ugyanakkora arányban (1:1) forduljon elő. Következő feladat a gyalogátkelő útpálya létrehozása, amelyen keresztül átjutnak a gyalogosok a négy sávós főúton egyik oldalról a másikra. Az útpálya lehelyezése a „Links” paranccsal történik és lényeges, hogy mindkét vége beleérjen a járda útpályákba. Ezután a lehelyezett gyalogátkelő útpályára a „Pavement Markings” parancs segítségével elhelyeztük a gyalogátkelő felfestést is. Még egy fontos lépés volt hátra ebben a fázisban, ugyanis a már hamarabb létrehozott gyalogosátkelő lámpacsoportot le kellett helyezni a „Signal Heads” paranccsal. Az elkészült szimuláció 3D változata a 4. ábrán látható.



4. ábra Az elkészült szimuláció 3D változata

A szimuláció megfelelően működött és a beállított jelzőlámpa programok még a magas óránkénti járműsűrűség ellenére is hatékony forgalomirányítást valósítottak meg. Ezt követően került bővítésre a szimuláció a megkülönböztetett járművekkel. A következő kihívás a szoftverben alapértelmezett, egyedi lehetőségek bővítése a sürgősségi járművekre vonatkozóan. A bővítés a jármű főútról való balra kanyarodási lehetősége volt. Ehhez mindkét kereszteződésben a négysávos főútról balra kanyarodó csatlakozást kellett létrehozni a megkülönböztetett jármű számára. A balra kanyarodás tényleges megvalósításához, a csatlakozás mellé bizonyos beállításokat kell hozzárendelni. Ezen kívül blokkolva lett a vészhelyzetin kívül minden más járműosztály balra kanyarodás esetén. Ezek után egy másik nagyon fontos feladatot kellett megoldani. A „Conflict Areas” parancs segítségével kellett meghatározni, hogy a balra kanyarodó járműnek minden más irányból érkező járművel szemben, forgalomirányító lámpától függetlenül elsőbbségi jogosultsága legyen. Az elkészült szimulációban a 5. ábrán jól látható, ahogy a megkülönböztetett jármű (fehér színnel jelölve) balra kanyarodik és elsőbbséget kap a szemből érkező járművektől, pedig zöld fényjelzést ad nekik a forgalomirányító lámpa.



5. ábra Szimuláció futtatása, balra kanyarodás

A 6. ábrán szintén a szimuláció látható működés közben. Azt a folyamatot látjuk, ahogy a sürgősségi jármű (piros színnel jelölve) áthalad a forgalomirányító lámpa piros jelzésén.



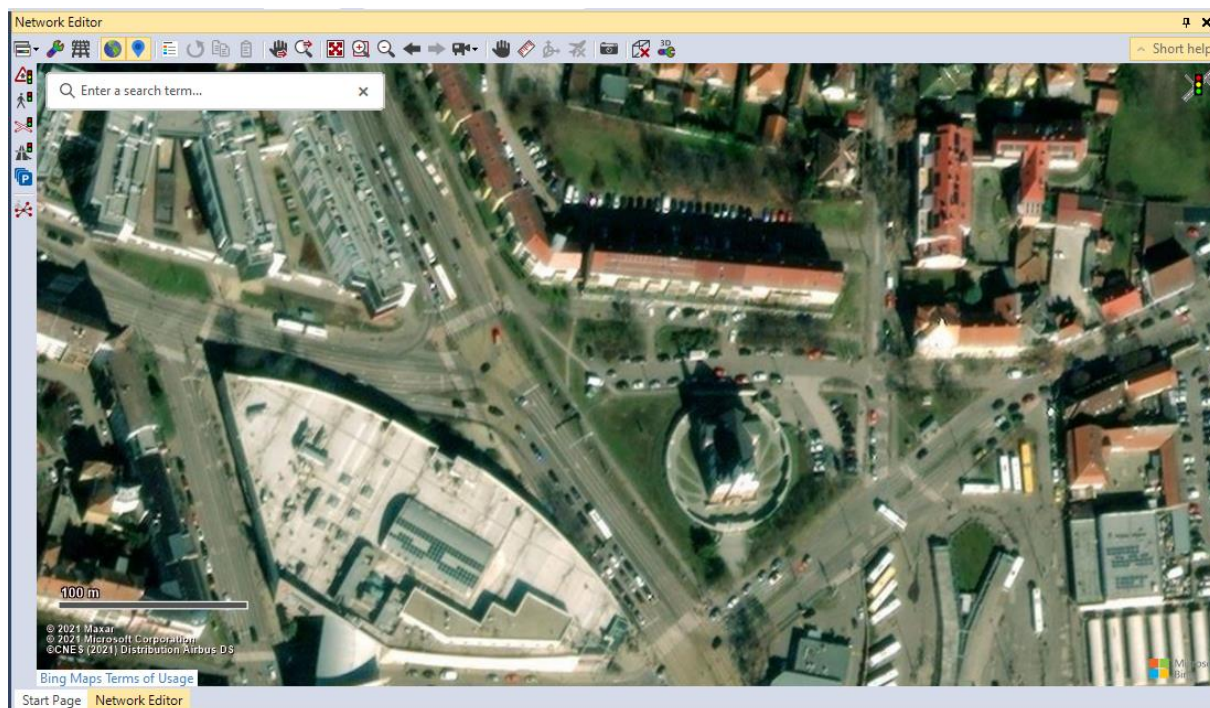
6. ábra Szimuláció futtatása, piroson áthajtás két időpillanata

Mindkét kereszteződésben sikeres működés valósult meg, a szimuláció tükrözte a valós forgalmi helyzeteket.

4.2. Jelenlegi szimuláció ismertetése

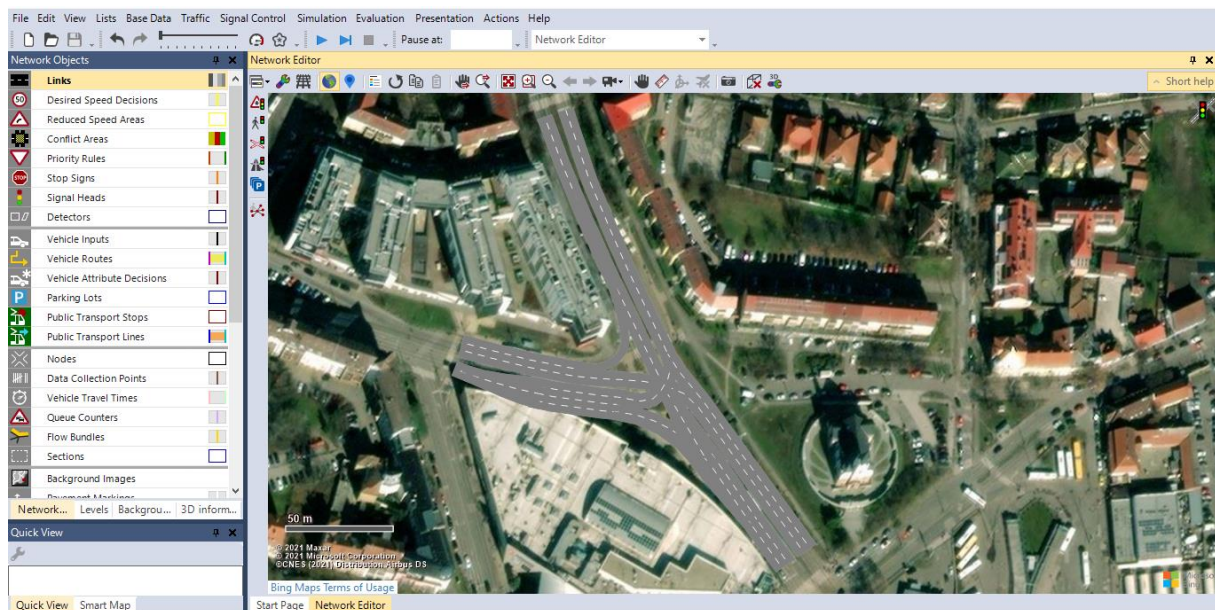
Ebben az alfejezetben kerül ismertetésre a jelenlegi dolgozatban bemutatni kívánt szimuláció. A korábbi szimulációhoz képest már egy új lokáció szolgál a szimuláció alapjául. A 2022. tavaszi TDK konferencián a bírálóktól kapott tanácsokat alapul véve választottam ki az új helyszínt, amely a miskolci Búza téren található Szeles utca-26-os főút kereszteződése. Ez Miskolc egyik legforgalmasabb kereszteződése, így egy sikeres szimuláció elkészítésével elmondható, hogy Miskolc bármelyik kereszteződésében megvalósítható a modellezés.

A szoftver új munkalap indításakor megjeleníti a világtérképet, ahol lehetőségünk van kiválasztani a szimuláció helyszínét. Ezzel a lehetőséggel valódi helyszínekre tudunk keresni, a gyorsabb helymeghatározás érdekében. A lokációként választott Búza téri kereszteződés a 7. ábrán látható.



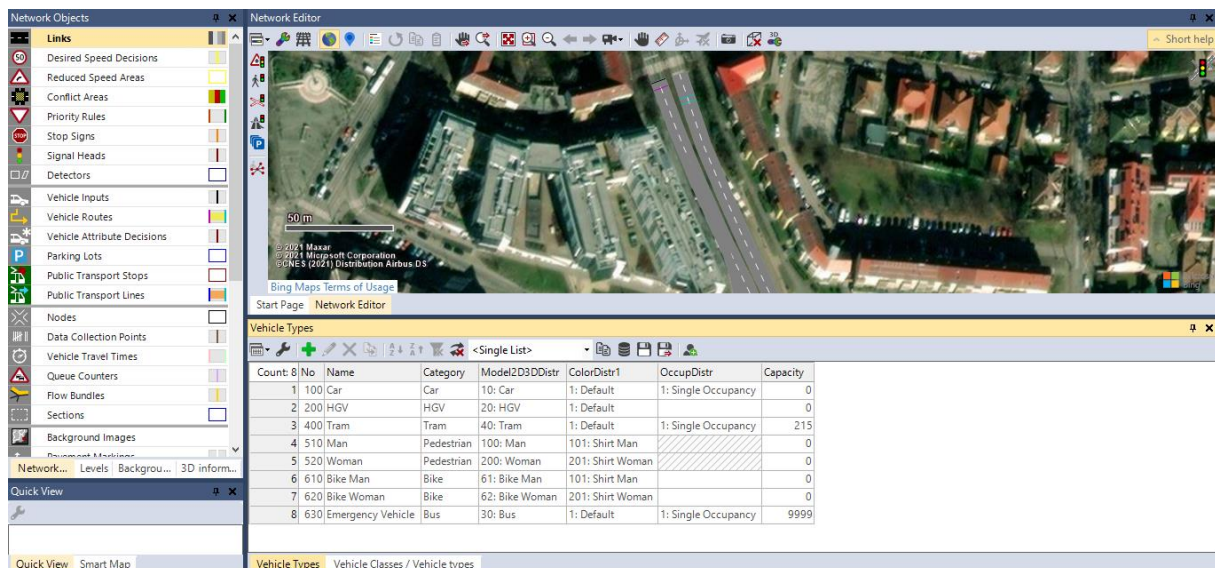
7. ábra A kiválasztott lokáció

Ezek után a fő útpályák létrehozása a feladat. A „Links” paranccsal tudunk fizikai pályát létrehozni. Meghatározható milyen hosszú szakaszon szeretnénk a pályát, illetve hány sáv legyen. Ebben a szimulációban elhelyezésre került a 26-os főúton a Nagyboldogasszony templomtól a modellezett kereszteződésig egy három sávú útpálya, amely a kereszteződésben kettéágazik, mivel balra kanyarodási lehetőség van a belső sávból, illetve a középső sávból balra és egyenesen lehet tovább haladni, ezen felül a külső sávból pedig csak egyenesen lehet tovább haladni a Szentpéteri-kapu irányába. Ebből az irányból dél felé került elhelyezésre a 26-os főúton egy kétsávú útpálya, amely egészen a Nagyboldogasszony templomig egyenesen halad tovább és a Miskolc Pláza épületét elérve három sávossá bővül. A szimulált kereszteződés előtt elágazik egy út az említett szakaszcól, amely a Szeles utca Macropolis épület előtti kétsávú útszakaszba kapcsolódik be, ezzel létre hozva egy harmadik sávot. A Szeles utca nyugati irányából jövő kétsávú útpálya négy sávossá szélesedik ki, ezek közül kettő sáv balra kanyarodik a 26-os főút Szentpéteri-kapu felé tartó két sávú útpályájára, illetve a másik két sáv jobbra kanyarodva a Pláza épülete melletti útpályába csatlakozik három sávossá bővítve azt. Az elkészített útpályák a 8. ábrán láthatóak.



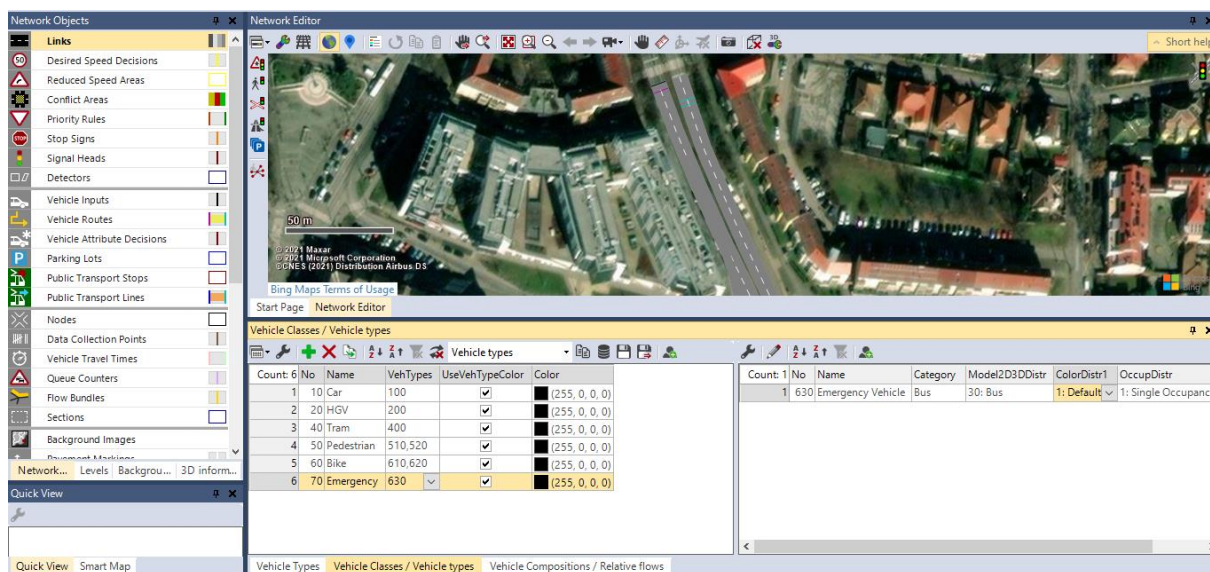
8. ábra Útpályák létrehozása

Ezek után, az új járműtípus létrehozása volt a feladat, amely a sorszámában következő 630-as típusú „Emergency Vehicle” néven kerül beillesztésre, így könnyen meg lehet különböztetni a többi járműtől. Ahogyan a 9. ábrán is látható a nyolcadik, sürgősségi járműtípus esetén a 30-as busz kategória került beállításra. Ennek a jobb szemléltetés volt az oka, mivel így könnyebben észrevehető a szimuláció futtatása során, hogy mikor érkezik sürgősségi jármű. Természetesen emiatt a valódi busz kategória kivételre került, nehogy téves információt kapjunk a szimuláció során.



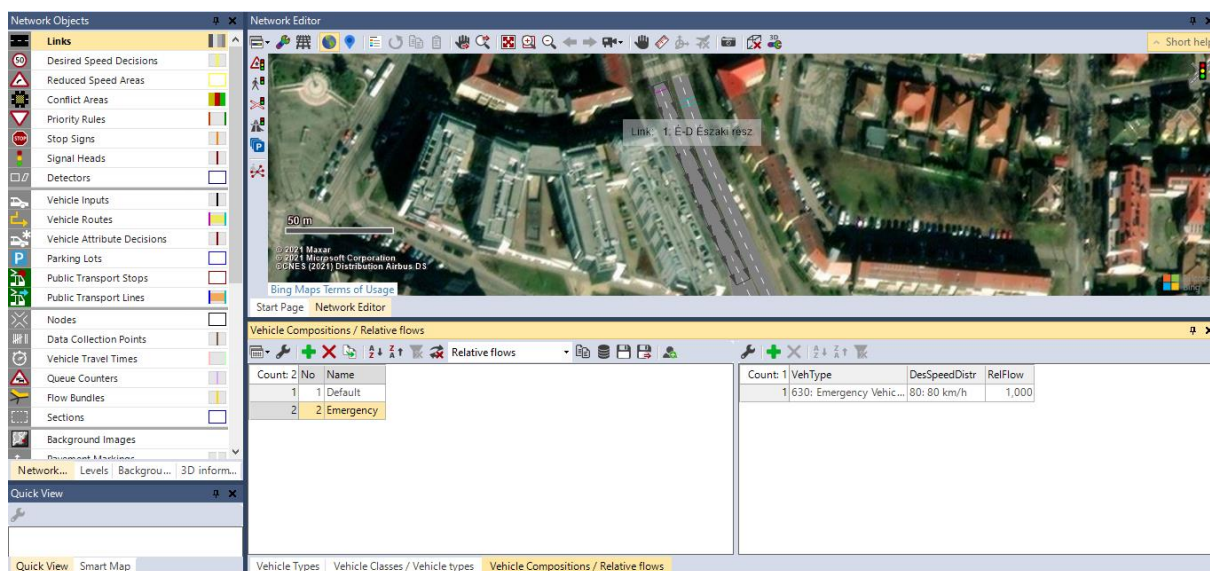
9. ábra Járműtípus meghatározása

Ezt követően a 10. ábrán látható módon meghatározásra került 70-es járműosztályként a 630-as sürgősségi járműtípus. Emellett megfigyelhető a korábban beállított „busz” kategória is.



10. ábra Jármű osztály meghatározása

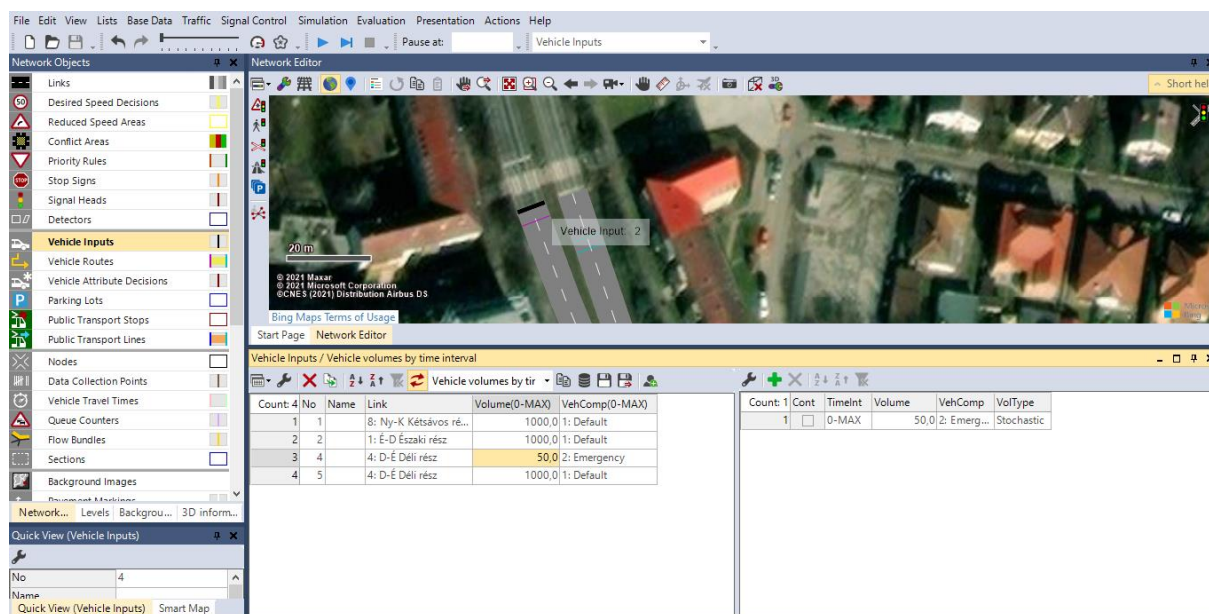
A 11. ábrán látható még egy fontos beállítás a „Vehicle Compositions” menüpontban, ahol a sürgősségi járműtípushoz 80 km/h sebesség lett hozzárendelve. Ezen könnyen lehet állítani, akár több alkalommal is a szimulációt leállítva, így különböző sebességek esetén is megfigyelhető, a modellezés eredménye. Jelen esetben ez a sebesség érték beállítás ideális útviszonyokat, illetve sürgős riasztást feltételez. Természetesen ez változhat, mivel a sürgősségi jármű sebessége függ az időjárástól, a forgalmi körülményektől és a vészhelyzet körülményei is befolyásolják.



11. ábra Jármű kompozíció beállítása

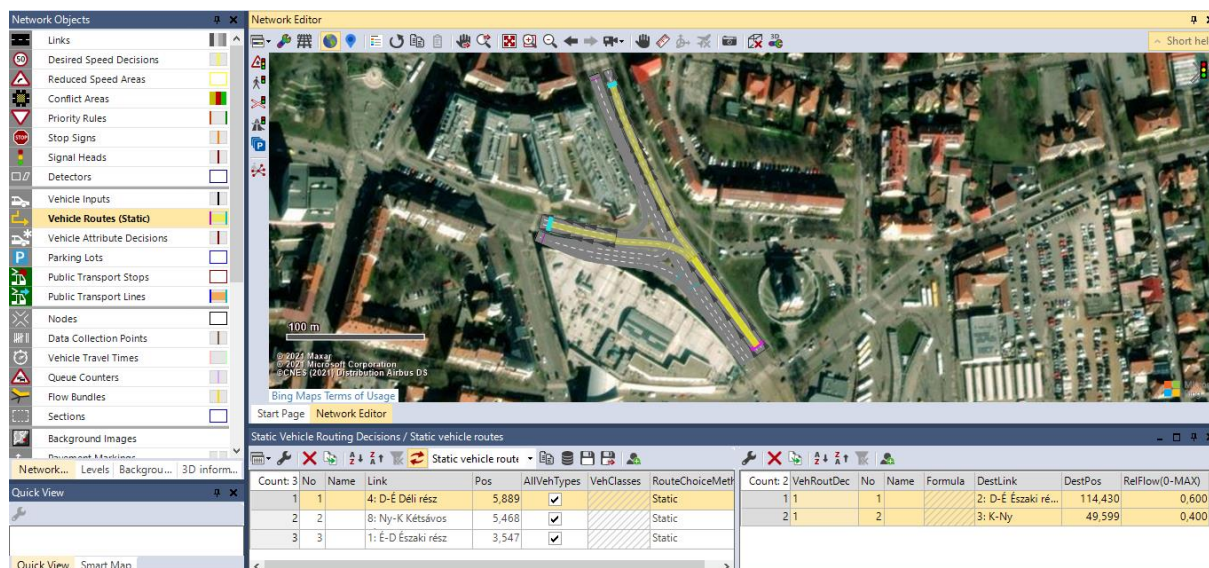
Következő lépés a jármű bemenetek meghatározása, azaz, hogy honnan fognak a szimulációba járművek érkezni. A „Vehicle inputs” paranccsal egyszerűen elhelyezhetők a bemenetek, melyek. Összesen négy jármű bemenet került meghatározásra. Az első Nyugatról Keletre irányban a Szeles utcán került beillesztésre, a második az Északról Délre tartó

útpályára, az utolsó kettő pedig a Délről Észak felé tartó útszakaszra, amelyekből az egyik az alapértelmezett jármű bemenet (amely több különböző civil jármű típust is tartalmaz), a másik pedig egy vészhelyzeti jármű bemenet. Az alapértelmezett járműveknek óránként ezer mennyiség került beállításra, amely reálisnak tűnik egy nagy forgalmú kereszteződés estén. Egyszerűsítés történt a sürgősségi járművek bemeneténél, mivel egyelőre még csak az egyik útpályán került beillesztésre, illetve óránként ötven mennyiség lett meghatározva, ezzel könnyítve az adatszerzést a szimuláció futtatása során. Az említett jármű bemenetek a 12. ábrán láthatók.



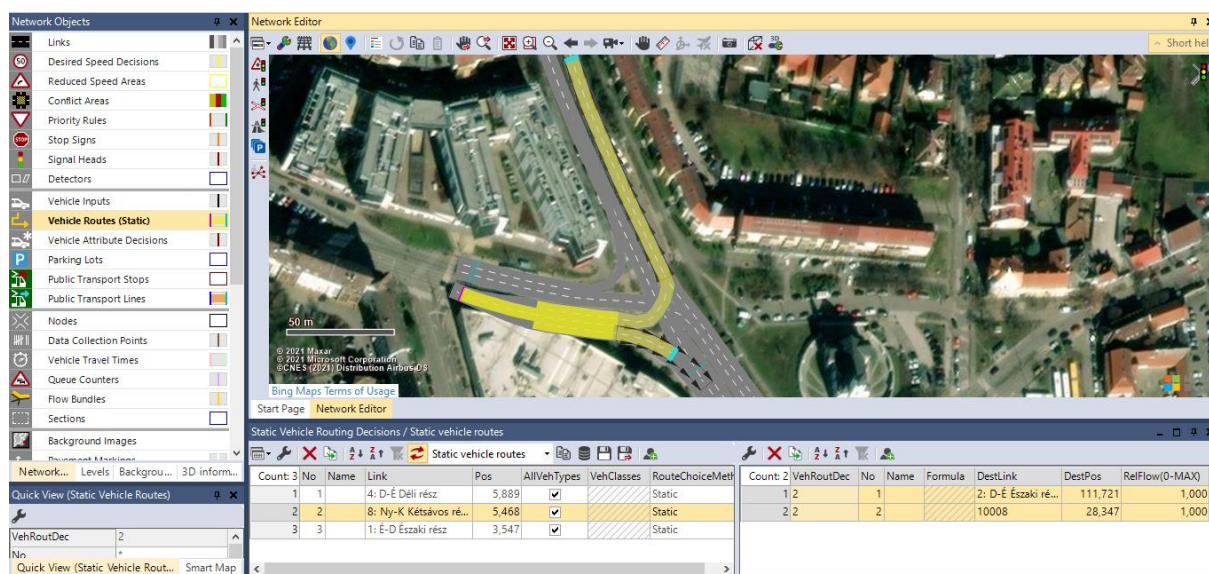
12. ábra Jármű bemenetek

A járművek elhelyezésre kerültek a szimulációban, következő lépésként pedig a lehetséges útvonalaik lettek meghatározva. A „Vehicle Routes” paranccsal elhelyezhető, hogy egy adott jármű bemenet milyen útvonalakon haladhat tovább. Ebben a szimulációban a járműveknek három féle lehetséges útvonalai lehetnek. Az első a Délről Északra tartó útpálya, amely elágazhat és így az ott beillesztett járművek haladhatnak tovább egyenesen a Szentpéteri-kapu irányában, illetve balra kanyarodva Diósgyőr irányába. Ezek az útvonal lehetőségek a 13. ábrán láthatóak.



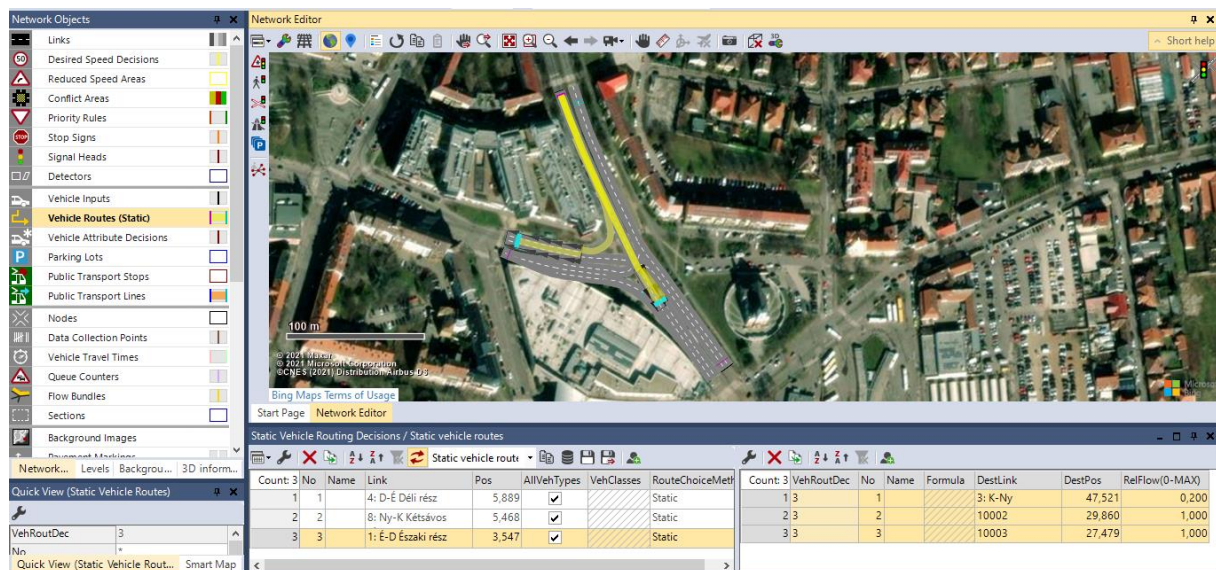
13. ábra Járművek lehetséges útvonalai (Dél-Észak; Dél-Nyugat irányokban)

A második lehetséges jármű útvonal a Nyugatról érkező Szeles utcai útpályán elhelyezett jármű bemeneteké, amelyek az útpálya négy sávossá bővülése után két felé haladhatnak tovább, balra kanyarodva a Szentpéteri-kapu irányába haladó útpályára, illetve jobbra kanyarodva a Miskolc Pláza melletti útszakasz felé déli irányba. Ezek a lehetséges útvonalak a 14. ábrán láthatóak.



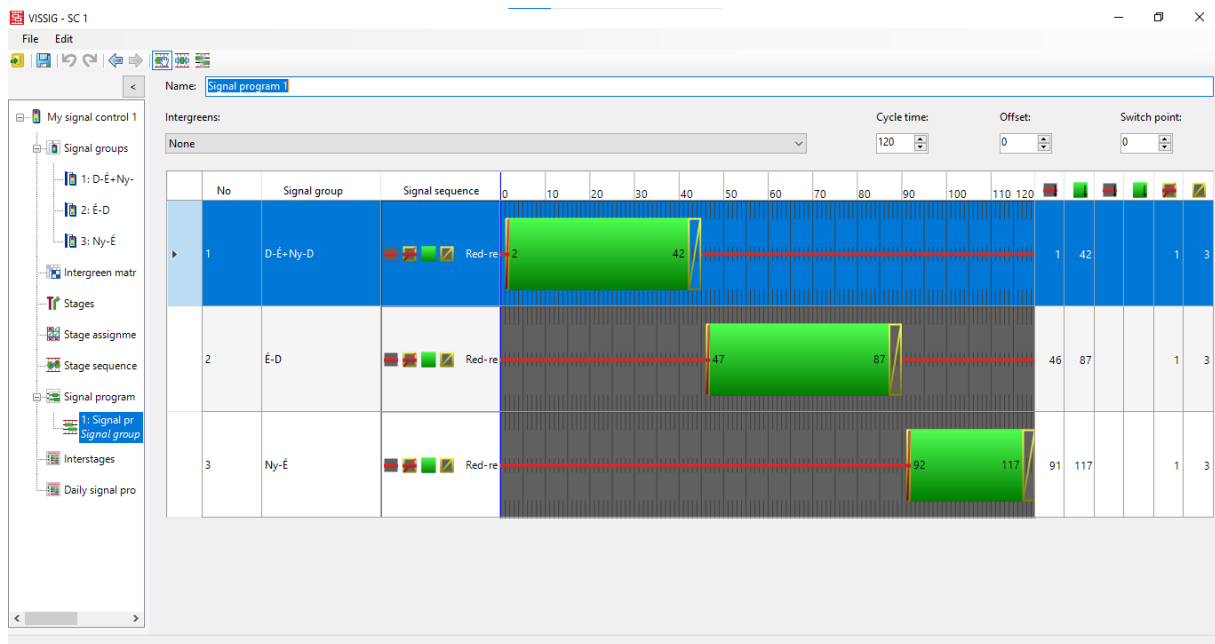
14. ábra Járművek lehetséges útvonalai (Nyugat-Dél; Nyugat-Észak irányokban)

A harmadik lehetséges jármű útvonal az Északról Délre haladó 26-os útra beillesztett járműveké, amelyek haladhatnak egyenesen tovább a kereszteződés után a Pláza melletti útszakaszra, illetve letérhetnek az útról még a kereszteződés előtt a Macropolis épülete előtt haladó Szeles utcai útpályára. Ezek a lehetőségek láthatóak a 15. ábrán.



15. ábra Járművek lehetséges útvonalai (Észak-Nyugat;Észak-Dél irányokban)

A jelen dolgozatban bemutatott szimuláció során nem volt szükség elsőbbség adási kötelezettség beállítására, mivel a modellezett kereszteződésben nincsenek „telezőld” forgalomirányító eszközök. Minden útszakaszon a valóságnak megfelelően irányított fényjelzők kerültek elhelyezésre. Ezeknek a fényjelzőknek egy úgynevezett „Signal Controller” került meghatározásra, amelyen belül három jelzőlámpa csoport lett létrehozva. Ezeknek a jelzései a 16. ábrán látható módon lettek programozva. Látható, hogy a ciklusidő 120 másodpercre lett beállítva, illetve az is, hogy a Délről Északra haladó útpálya fényjelzői és a Diósgyőr felől jövő Szeles utcai útszakasz fényjelzői egy jelzőlámpa csoportba kerültek. Így a jelzései is egyformán lettek programozva. Ennek az az oka, hogy ez a két útszakasz nem keresztezi egymást, így nyugodtan lehet azonos a jelzés programjuk.

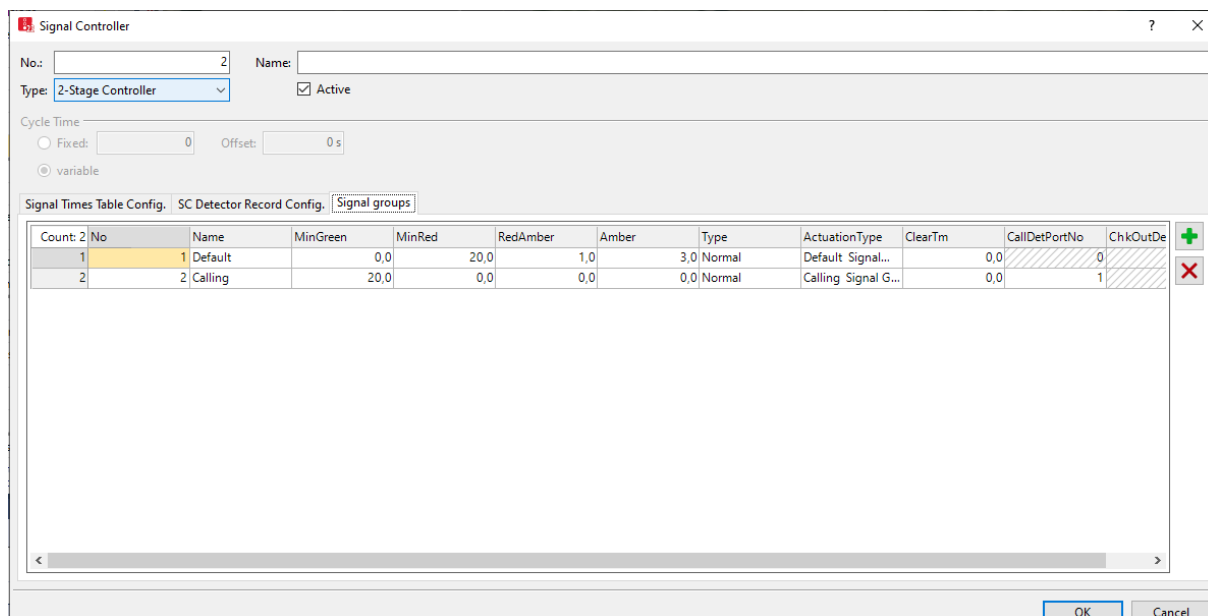


16. ábra Jelzések programozása

A beállítások megfelelőek voltak, a jelzőlámpa program jól működött, bár a jelenlegi szimulációban nem ez volt a fő cél, viszont ez is egy fontos lépés volt ahhoz, hogy a modellezés valóságghú legyen.

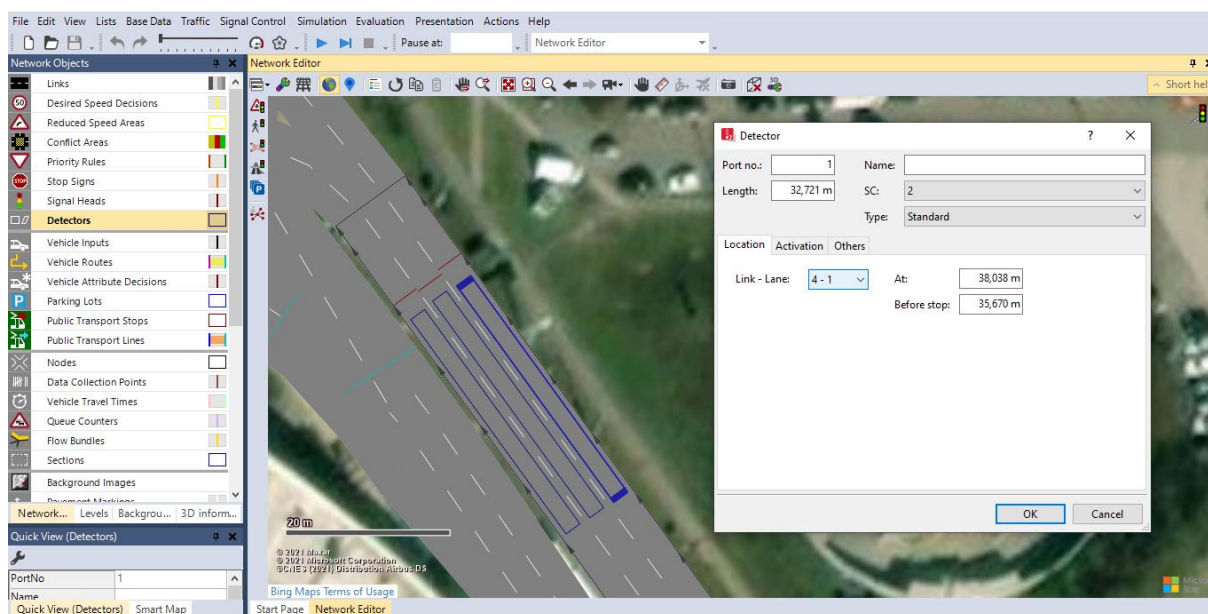
4.3. Intelligens forgalomirányító rendszer

A következő lépés már az intelligens forgalomirányító rendszer létrehozásához szükséges. Egy új „Signal Controller”, vagyis jelzés vezérlő lett létrehozva. Ez a szimuláció második jelzés vezérlője, amely az előzővel ellentétben úgynevezett „two-stage” típusú. Ez azt jelenti, hogy a forgalomirányító egységek beállításánál, meghatározható, hogy az adott irányító eszköz a második jelzés vezérlőn belül „Default”, vagyis alapértelmezett, vagy „Calling”, vagyis a hívó jelzőlámpa csoportba tartozzon. Az alapértelmezett jelzőlámpa csoportba tartozó forgalomirányító eszközök folyamatosan zöld jelzést fognak biztosítani a járművek számára, egészen addig, amíg a hívó jelzőlámpa csoportba tartozó eszközök detektorai nem érzékelnek járművet. Amennyiben érzékelnek, úgy a hívó forgalomirányító eszközök zöld jelzést biztosítanak az érkező járműveknek és az alapértelmezett forgalomirányító eszközök piros jelzésre váltanak. Erre a két különböző csoportra eltérő beállításokat lehet alkalmazni. A kezelőfelületen be lehet állítani például a zöld, vagy a piros jelzés minimum időtartamát, illetve a hívó detektor port számát. Ez azért fontos, mert a lehelyezett detektor beállításakor a meghatározott port számot szükséges használni. A bemutatott jelzőlámpa csoportok a 17. ábrán láthatóak.



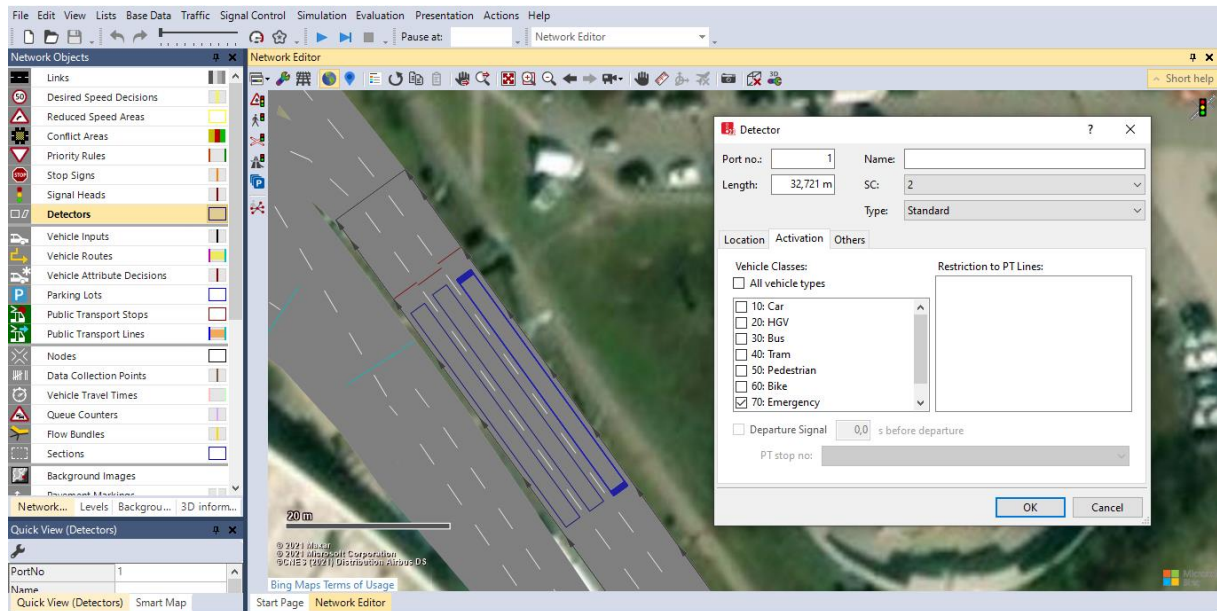
17. ábra Jelzés vezérlő beállítása

A következő lépés a detektorok elhelyezése, illetve beállítása volt. Három detektor került elhelyezésre a Nagyboldogasszony templomtól induló északi irányba tartó 26-os főút háromsávos útpálya szakaszára. Közvetlenül a forgalomirányító egységek elé kerültek és fél útpálya hosszúságban érzékelnek, ami körülbelül 32 méter távolságot jelent. Ezt a detektor kezelőfelületén be lehet állítani. Ezen kívül fontos volt beállítani a korábban is említett port számot, amely megegyezik a jelzés vezérlőnél beállított számmal. A detektor típusa „alapértelmezett”-re lett beállítva. Ezen felül még szükséges volt kiválasztani, melyik jelzés vezérlő szerint működjön a detektor. A detektorok és a beállítások a 18. ábrán láthatóak.



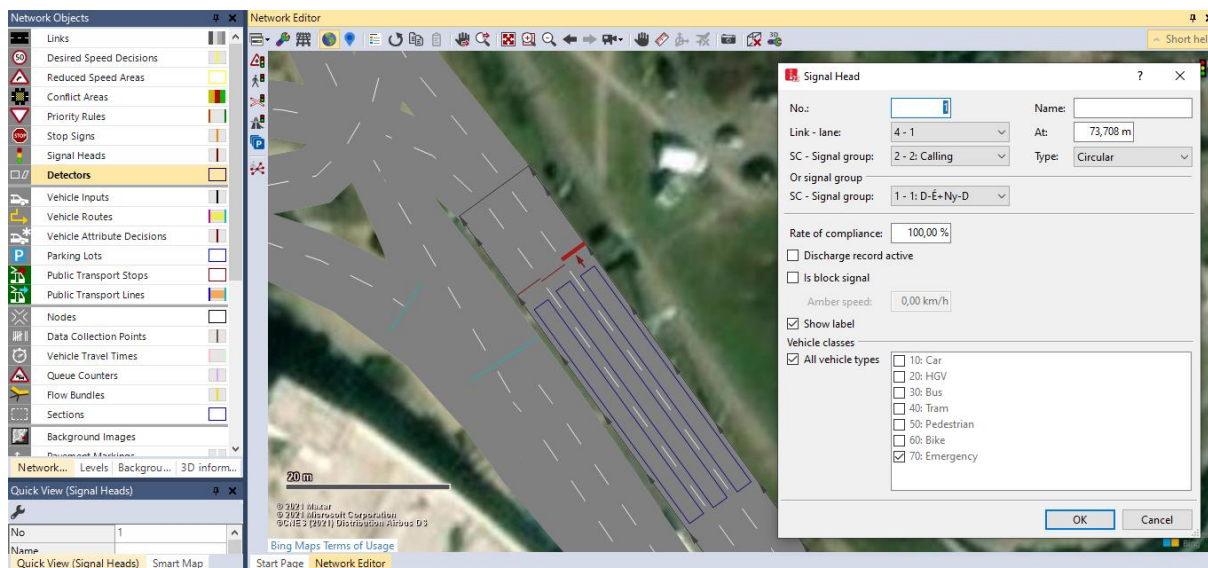
18. ábra Detektor beállítása (Lokáció)

A következő lépésben még mindig a detektoron volt szükséges állítani. A kezelő felületen az „Activation” fül alatt lehet beállítani, hogy az adott detektor milyen járműtípusokat érzékeljen, vagyis mire aktiválódjon. Ebben az esetben csak az „Emergency”, vagyis vészhelyzeti járművek lettek bejelölve, így a detektor csak sürgősségi járműtípus érkezése esetén aktiválódik. A beállítás a 19. ábrán látható.



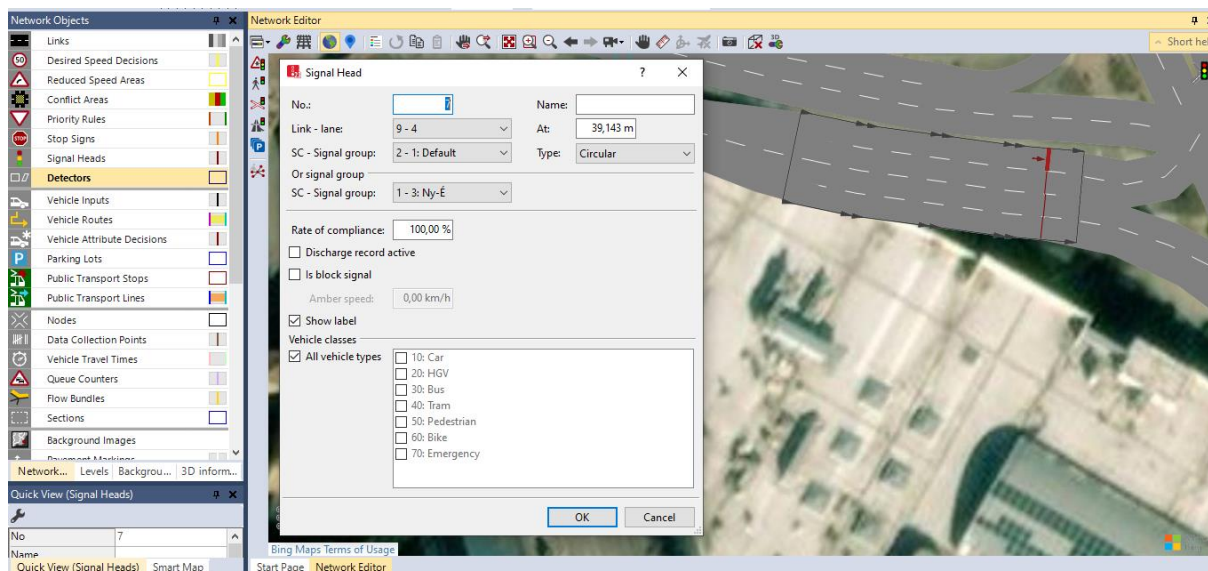
19. ábra Detektor beállítása (Aktiválás)

A detektorok beállítása után, a forgalomirányító eszközök beállítása következett. A „Calling”, vagyis a hívó jelzőlámpa csoport beállítást a Délről Északra tartó, 26-os főút útpályán elhelyezett forgalomirányító eszközök kapták. Ahogyan a 20. ábrán is látható, a „Calling” jelzőlámpa csoport mellett a korábban az 1-es jelzés vezérlőben meghatározott, jelzőlámpa csoport is hozzá lett rendelve az említett forgalomirányító eszközökhöz. Emellett az eszköz típusának a „Circular” lett beállítva. Ezen kívül látható még a jelző lámpa sorszáma, illetve, hogy melyik útpálya melyik sávján helyezkedik el.



20. ábra Forgalmirányító egységek beállítása (Calling)

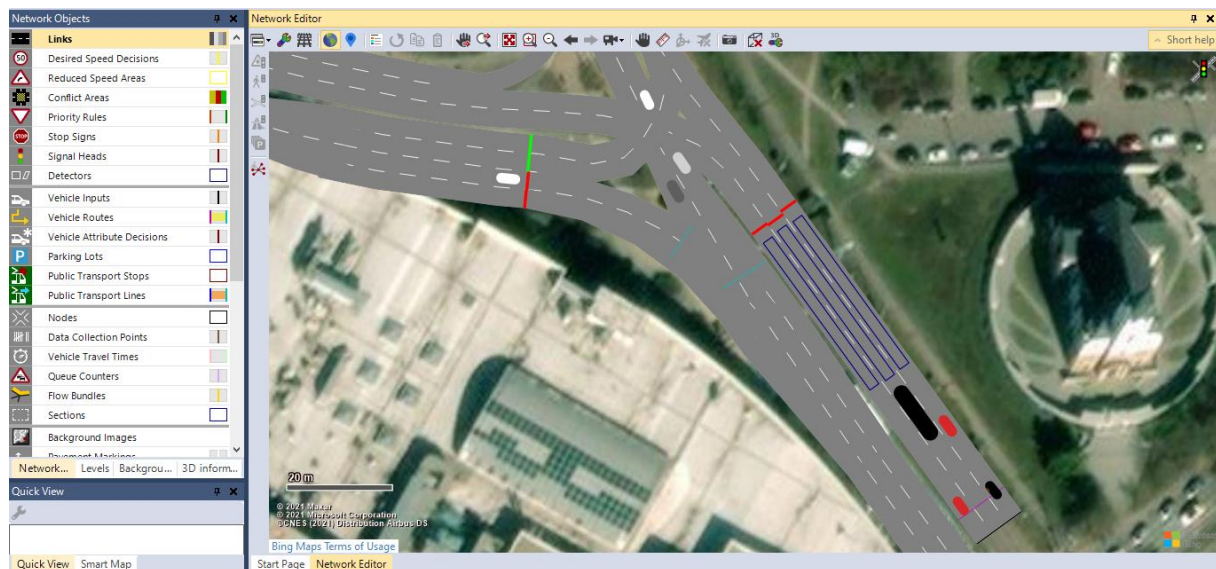
A Nyugatról érkező (először kettő) később négysávos útszakasz, balra a Szentpéteri-kapu felé kanyarodó két sávján található forgalmirányító egységek kapták a „Default”, vagyis alapértelmezett beállítást. Ezen kívül, a 21. ábrán is látható módon, a saját jelzőlámpa ciklusuk is beállításra került egy „vagy” paranccsal. Ez azt jelenti, hogy ezek az eszközök zöld jelzéssel működnek, ha a hívó eszközök piros jelzést adnak, vagy, amikor a saját jelzőlámpa ciklusa szerint zöld jelzést kell szolgáltatnia. A többi beállítás hasonlóan oldható meg, mint a korábbi egységnél.



21. ábra Forgalmirányító egységek beállítása (Default)

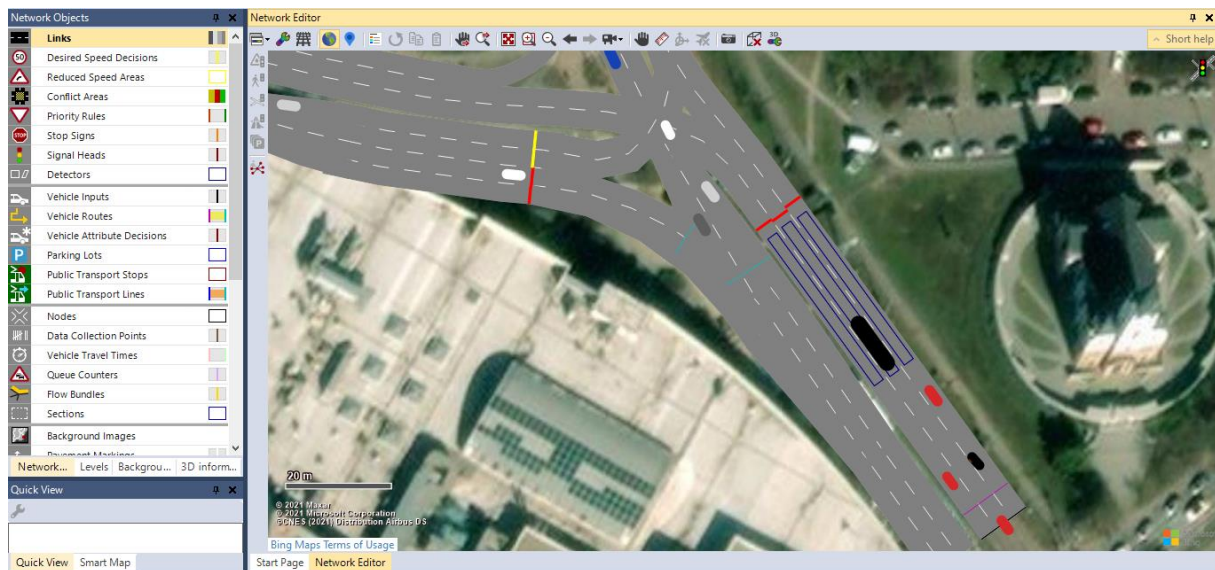
A szimulációban résztvevő maradék kettő forgalmirányító rendszernek a saját jelzőlámpa ciklusa került beállításra. A 22. ábrán látható a szimuláció futtatás közben. A jól felismerhető megkülönböztetett jármű (fekete busz) a detektor felé közeledik. Ebben a pillanatban még a

forgalomirányító lámpák piros jelzést adnak, illetve a Nyugatról Észak felé haladó útpálya forgalomirányító eszközei zöld jelzéssel működnek.



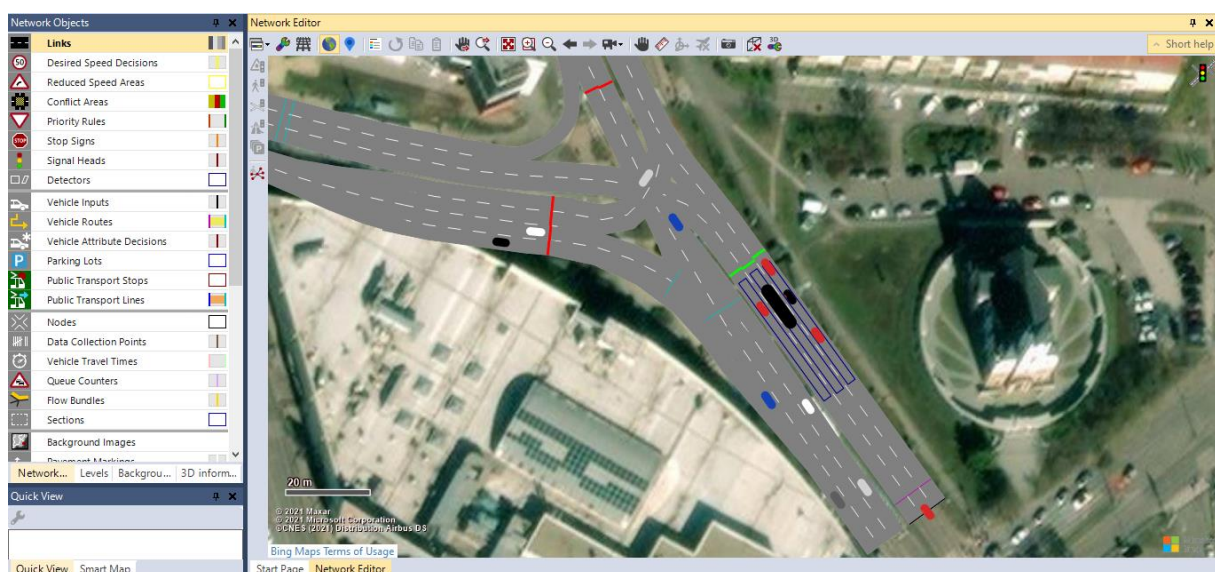
22. ábra Sürgősségi jármű közeledése (Piros-zöld jelzés)

A következő időpillanatban -amely már a 23. ábrán látható- a sürgősségi jármű már a detektor hatótávolságán belülre ér. Itt még a hívó forgalomirányító lámpák piros jelzést adnak, mivel ezeknek a jelzés-váltás ideje lassabbra van állítva. Valóban fontos, hogy minél gyorsabban szabad utat kaphassanak a civil járművek, ezzel szintén szabad utat biztosítva a sürgősségi járműnek, azonban az is fontos, hogy az utat keresztező útszakaszból érkező járművek még gyorsabban kapjanak piros jelzést. Enélkül rendkívül balesetveszélyesen és nem megfelelően működne az intelligens forgalomirányító rendszer. Látható a 23. ábrán, hogy a gyorsabb jelzés-váltási idővel rendelkező forgalomirányító egységek (Nyugatról Észak felé) a megkülönböztetett jármű érkezésekor a detektor területre, szinte azonnal narancssárgára váltanak. Ezzel már figyelmeztetik az úton érkező járműveket a megállásra. A legfontosabb, hogy a sürgősségi jármű útját keresztező útszakaszból érkező járművek hamarabb megálljanak, mint ahogyan az említett útszakaszból elindulnak a civil és a sürgősségi járművek.



23. ábra Sürgősségi jármű közeledése (Piros-narancssárga jelzés)

A 24. ábrán látható a harmadik időpillanat, amely során már minden forgalomirányító eszköz piros jelzésre váltott és a hívó detektor által aktiválódott eszköz pedig zöld jelzéssel engedi a civil járműveket, ezzel szabad utat biztosítva a megkülönböztetett járműnek.



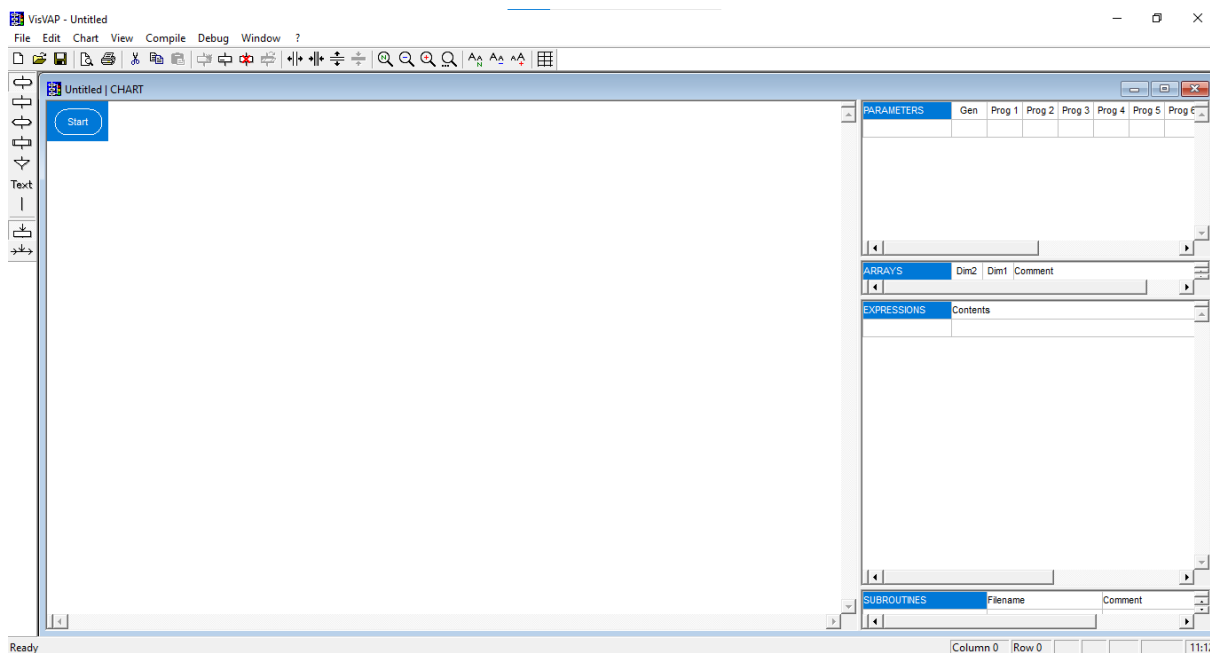
24. ábra Sürgősségi jármű közeledése (Zöld-piros jelzés)

Az intelligens forgalomirányító rendszerek szimulációja sikeresen működött, az összehangolása azoknak a forgalomirányító jelzőlámpáknak, amelyek a saját ciklusukkal lettek beállítva, megfelelő volt. Ezen felül a „Default” és „Calling” beállításokkal ellátott eszközök is a „vagy” parancs segítségével a saját jelzőlámpa ciklusukkal is képesek voltak működni.

4.4. További fejlesztési lehetőségek

A szimuláció az elvártaknak megfelelően működött. Azonban voltak olyan szituációk a szimuláció futtatása során, amelyek további fejlesztést igényelnek. Ahogyan már korábban

említésre került, a forgalomirányító eszközök „vagy” paranccsal vannak ellátva. Ez a parancs viszont azt határozza meg, hogy az eszközök mikor adjanak zöld jelzést és nem azt, hogy mikor váltsanak át piros jelzésre. A modellezés során volt olyan eset, hogy a hívó forgalomirányító eszköz a saját jelzőlámpa ciklusa alapján zöld jelzésre váltott, ezt azonban az alapértelmezett forgalomirányító eszköz nem érzékelte, mivel nem a hívó detektor aktiválta a forgalomirányító lámpát, így az is zöld jelzésen működött tovább. Ezt sajnos nehéz kiküszöbölni, hiába a jó jelzőlámpa ciklus beállítás, illetve a működő „two-stage” jelzés vezérlő. Ugyanakkor az sem megoldás, hogy a hívó forgalomirányító lámpacsoport nem kap „vagy” parancsot, hiszen az nem lenne életszerű, mivel ebben az esetben csak akkor adna zöld jelzést az eszközcsoporthoz, ha sürgősségi jármű érkezik. A jövőben programozással történne a forgalomirányító rendszer egy részének a beállítása. Ez megoldást jelent az előbb említett problémára, mivel programozás segítségével meghatározható az is, hogy bizonyos forgalomirányító eszközök mikor váltsanak piros jelzésre. A programot a VisVAP modul segítségével lehet létrehozni. A 25. ábrán látható a VISSIM könyvtárszerkezetéből megnyitott VisVAP modul, amelyen belül folyamatábra készítésére van lehetőség, aminek megfelelően működtethetők a forgalomirányító rendszer eszközei.



25. ábra VisVAP modul

5. ÖSSZEFOGLALÁS

A dolgozatban bemutatott technológia egy gyakran kialakuló veszélyhelyzet megakadályozására szolgál, amellyel a személygépjárművel közlekedők gyakran találkozhatnak. Az első fejezetében egy korábbi munkámban fejlesztett eszköz került röviden bemutatásra, amely megkülönböztetett járművek közeledését jelző rendszer civil járműveken belül. Az eszköz felhasználásának másik lehetősége, hogy nem csak a civil járműveknek jelzi közeledését, hanem az intelligens forgalomirányító rendszereknek is. A dolgozat irodalmi háttereként többek között egy olyan kísérlet került bemutatásra a második fejezetben, amelyből jól látszódott, hogy mennyi időt veszít egy veszélyhelyzeti jármű a zsúfolt városi környezetben a nem megfelelő forgalomirányító rendszerek miatt. Ezek után olyan járműkommunikációs rendszerek kerültek bemutatásra, melyek hatékonyan segítik az intelligens forgalomirányító rendszerek működését. Emellett még számos intelligens forgalomirányító technika került ismertetésre szintén a második fejezetben. A harmadik fejezetben a PTV Vissim közlekedésszimulációs szoftver került bemutatásra. Először általánosan, majd a jelen dolgozatban bemutatott téma szemszögéből is. Több olyan lehetőség is bemutatásra került, amelyet az aktuális modellezésben sikerült felhasználni, vagy a jövőbeni továbbfejlesztésekhez hasznos lehet. A negyedik fejezetben röviden ismertetésre került egy korábban elkészített szimuláció. Ezzel megismerhető az a folyamat, amelyre a jelenlegi modellezéshez való eljutásig szükség volt. Ezt követően részletesen ismertetésre került a jelenlegi intelligens forgalomirányító rendszer szimulációja. Emellett az eredmények is bemutatásra kerültek, illetve ábrák segítségével nyomon lehetett követni a megkülönböztetett jármű mozgását a szimuláció során. Majd a fejezet végén a továbbfejlesztési tervek kerültek bemutatásra.

Köszönetnyilvánítás

A TDK dolgozatban ismertetett kutató munka az NTP-SZKOLL-21-0026 azonosítószámú „Szabad utat a tehetségnek! - a Terplán Zénó Szakkollégium tehetséggondozó programja” elnevezésű projekt keretében valósult meg az Emberi Erőforrások Minisztériuma és az Emberi Erőforrás Támogatáskezelő támogatásával.

6. IRODALOMJEGYZÉK

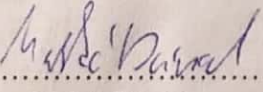
- [1] Makó, D.: Sürgősségi járművek közeledésének egyéb járművön belüli jelzésére szolgáló rendszer megvalósítása, TDK dolgozat, 2021 november 15. TDK konferencia, Miskolci Egyetem, Miskolc, 2021
- [2] Bosquez, C.; Lovato, M.; Lucas, L.: Ambulance alert system step in frequency modulated using SDR, in ANDESCON, IEEE, 2014
- [3] Nono R.; Asudais R.; Alshmrani R.; Alamoudi S.; Aljahdali A.: Intelligent Traffic Light for Emergency Vehicles Clearance, ADCAIJ, 2020
- [4] CallioVision: Intelligens forgalomfigyelő és -szabályozó rendszerek, 2020-2021
- [5] Yusuf, A. N. A.; Arifin, S. A.; Zulkifli, F.Y.: Recent development of smart traffic lights, IAES International Journal of Artificial Intelligence (IJ-AI), 2021
- [6] Rego, A., et al.: Software Defined Network-based Control System for an Efficient Traffic Management for Emergency Situations in Smart Cities, *Futur. Gener. Comput. Syst.*, vol. 88, pp. 243-253, 2018, doi: 10.1016/j.future.2018.05.054
- [7] Naik, T.; Roopalakshmi, R.; Jain, D. R. N, P.; Sowmya, B. H.: RFID-Based Smart Traffic Control Framework for Emergency Vehicles, 2018 Second Int. Conf. Inven. Commun. Comput. Technol., pp. 398–401, 2018
- [8] Gillis, A. S.: What is internet of things (IoT)?, TechTarget, 2021
- [9] Kovács, Sz.; Járműinformatika Bevezetés; Miskolci Egyetem, 2020
- [10] Jeong, J.; Shen, Y.; Xiang, Z.: Vehicular Neighbor Discovery for IP-Based Vehicular Networks, Sungkyunkwan University, 2019
- [11] V2X: What is Vehicle to Everything?, Thales, 2021
- [12] Almuraykhi, M. K.; Akhlaq, M.: STLS: Smart Traffic Lights System for Emergency Response Vehicles, IEEE, 2019
- [13] Aleko, R. D.; Djahel, S.: An Efficient Adaptive Traffic Light Control System for Urban Road Traffic Congestion Reduction in Smart Cities, *Information*, MDPI, 2020
- [14] Schmidt, S.: Smart traffic lights enable emergency vehicle priority, PTV Group, 2021
- [15] Budapesti Műszaki és Gazdaságtudományi Egyetem Közúti Közlekedésirányítási Laboratórium, 2021
- [16] Skapinyecz, R.; Erdei, L.: Korszerű közlekedésszimulációs-és tervező szoftverek alkalmazási lehetőségei az oktatásban és a kutatásban; XXIX. Nemzetközi Gépészeti Konferencia, 2021

- [17] Sukennik, P.: PTV Vissim extension-new features and improvements-final version, PTV Group, 2020
- [18] Khan, S.; Khalid, S.; Maoh, H.; Tepe, K.: Making informed route choices using V2V/V2I communications: The case of Highway401/402 Corridor, CTRF, 2018
- [19] Makó, D.: Sürgősségi járművek közeledésének jelzésére szolgáló rendszer alkalmazása forgalom irányítására, TDK dolgozat, 2022 április 25. TDK konferencia, Miskolci Egyetem, Miskolc, 2022

Eredetiség nyilatkozat

Alulírott **Makó Dávid** (Neptun kód: IRNKK6), a *Miskolci Egyetem Gépészmérnöki- és Informatikai Karának alapszakos járműmérnök* hallgatója büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírással igazolom, hogy ezt a dolgozatot saját magam készítettem, a benne leírt vizsgálatokat – ha ezt külön nem jelzem – magam végeztem el, és az ismertetett eredményeket magam értem el. Adatokat, információkat csak az irodalomjegyzékben felsorolt forrásokból használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

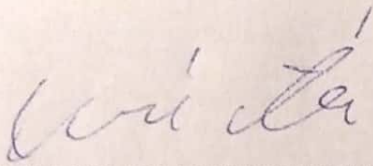
Miskolc, 2022. november 8.


.....
Makó Dávid

A konzulens nyilatkozata

Alulírott, **Dr. Cservenák Ákos**, a Miskolci Egyetem Logisztikai Intézetének oktatója, **Makó Dávid** alapszakos járműmérnök hallgató **Sürgősségi járművek mozgásának szimulációja PTV Vissim szoftverrel** című TDK dolgozatát beadásra alkalmasnak ítélem.

Miskolc, 2022 november 8.



.....
Dr. Cservenák Ákos