Modulrendszerű, önvezérlésre alkalmas, távvezérelhető, könnyű páncélvédettségű félplatós terepjáró katonai és katasztrófavédelmi célú terepjáró bázisjármű és kapcsolódó cserefelépítmények fejlesztése

Krecht Rudolf*, Pusztai Zoltán*, Kőrös Péter*

**Járműipari Kutatóközpont, Széchenyi István Egyetem*
Magyarország (Tel: +36 70 666 0478; e-mail: krecht.rudolf@ga.sze.hu).

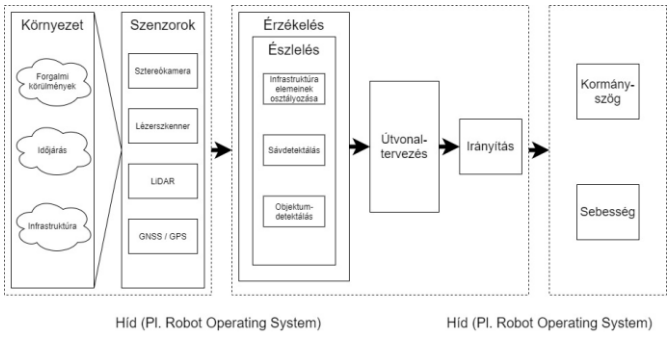
Absztrakt: Katonai és katasztrófavédelmi célú járművek esetén egyértelmű előnyökkel jár kockázatos feladatok ellátása során a jármű félautonóm, vagy távvezérelt mozgatása. Jelen cikk modulrendszerű, önvezérlésre alkalmas, távvezérelhető katonai és katasztrófavédelmi célú terepjáró távvezérelhetőségének és önvezető funkcióinak fejlesztését hivatott előkészíteni számítógépes szimulációs eszközökkel. A fedélzeti szenzoradatok alapján távvezérelt, vagy autonóm, önvezető járművek fejlesztési folyamata magas tesztigénnyel járhat hardveres és szoftveres oldalról is. Ezen tesztmérések eredményét befolyásolhatják olyan külső tényezők, mint az infrastruktúra változásai, vagy az időjárás. Továbbá említendő tényező a tesztmérések jelentős humánerőforrás-igénye is. A tesztfolyamatok felgyorsítása érdekében javasolt számítógépes szimulációkat alkalmazni adatgyűjtést célzó, valamint algoritmusvizsgálatot végrehajtó tesztek előtt. A valós mérések nem helyettesíthetők számítógépes szimulációkkal, viszont több iterációs lépés megtehető szimulációs környezetben, lehetővé téve a fejlesztési folyamat hardvermentes előkészítését. Jelen cikk elsősorban az SVL járműszimulátor alkalmazása által mutatja be a távvezérlésre, valamint autonóm járműfunkciók megvalósítására alkalmas jármű szimulációs módszereit. Az SVL szimulátoron kívül ismertetésre kerül több szimulációs eszköz is, összehasonlítva a bemutatott célra alkalmazható elérhető szimulátorokat.

1. BEVEZETŐ

Az önvezető és autonóm járművekhez és járműfunkciókhoz kapcsolódó tudományterületeken jelentős bővülés volt tapasztalható az elmúlt években. Mivel az önvezető funkciók több megoldás összetett rendszerben való alkalmazása által valósulnak meg, ezen a területen kiemelten fontos a hardver- és szoftverelemek összehangolt működése. Ezen komplex rendszerelemek megfelelő kollaborációja kiemelkedően magasnak számító tesztórák által érhető el, jelentősen megnövelve az autonóm járműfunkciókhoz kapcsolódó fejlesztések idő- és erőforrásigényét. Továbbá, a valós járműtesztek végrehajtása megköveteli a megfelelő kültéri követelmények rendelkezésre állását is. Kiemelendő továbbá, hogy egyes algoritmusfejlesztési folyamatok, irányítórendszerek paramétereinek hangolását célzó műveletek iteratívak, fontossá téve a tesztméréseket a fejlesztési folyamatok időtartama alatt is, nemcsak a fejlesztések végeztével [1]. A fedélzeti szenzorok segítségével megvalósuló teleoperáció szenzorok alkalmazása szempontjából nagyon hasonló az önvezető járműfunkciók fejlesztéséhez. Jelen cikk célja az SVL járműszimulátor alkalmazásának bemutatása katonai és katasztrófavédelmi célú terepjáró távvezérlésének és önvezető funkcióinak fejlesztéséhez. A szimuláció alkalmazásának különös szerepe van a bemutatandó fejlesztések esetében, ugyanis a fejlesztés alapjául szolgáló jármű nem áll rendelkezésre a fejlesztések előkészítési fázisában.

2. CÉLKITŰZÉS

Jelen cikk célja katonai és katasztrófavédelmi célú terepjáró teleoperációra és önvezető funkciókra kiterjedő fejlesztési folyamatának előkészítése. Az előkészítési folyamat során a fejlesztés alanyául szolgáló Gamma Komondor jármű nem áll rendelkezésre, fejlesztése és prototipizálása párhuzamosan zajlik. Ebből adódóan az előkészítési folyamat számítógépes szimulátor segítségével történik, így a fejlesztés alanyát képező jármű számítógépes szimulációjának elkészítése kulcsszerephez jut.
Az előkészítési folyamat célja a távvezérlés és az önvezetés fejlesztési irányok esetében alkalmazandó szenzorok és egyéb hardverkomponensek kiválasztása, valamint pozíciójuk meghatározása a jármű karosszériáján.
Az előkészítési folyamat céljából adódik az igény a tesztjármű számítógépes szimulációjának elkészítésére. Fontos, hogy a szenzortípusok és szenzorpozíciók kijelölése véglegesítés előtt kipróbált legyen, és mivel nem áll rendelkezésre valós jármű, ez az igény szimulátor segítségével végrehajtott teszteket követel meg. Szimulációalkotás során cél az 1. ábrán szemléltetett szenzoros és aktuátoros blokk (az ábra bal és jobb oldala) virtuális megvalósítása. Ezen komponensek megfelelő minőségű végrehajtása esetén az ábrán szereplő középső, szoftveres modul módosítás nélkül alkalmazhatóvá válik a szimulált és a valós járművön is.



1. ábra. Számítógépes szimuláció elemei

3. SZIMULÁCIÓS KÖRNYEZET KIVÁLASZTÁSA

A modellezendő jármű tulajdonságai és a fejlesztési cél alapján több számítógépes robotszimulációs eszköz is alkalmazhatónak bizonyul. Megfelelő lehet az ROS-Gazebo általános robotszimulációs eszköz, a CoppeliaSim általános robotszimulációs eszköz, az Unreal Engine 4 játékmotor, illetve a korábban bemutatott SVL járműszimulátor. A jelen cikkben ismertetett szimulációalkotási feladat megvalósítása szempontjából ideális szimulátor kiválasztása objektív pontrendszer felállításával valósul meg. A szempontok felvétele a feladat követelményei alapján történt.

1. Táblázat. Szimulációs környezetek összehasonlítása

	CoppeliaSim	Gazebo	Unreal Engine 4	SVL
ROS-kompatibilitás	7	10	2	9
Dinamikai szimuláció	10	3	1	9
Valósághű megjelenés	7	3	10	10
Szenzormodellek	8	10	4	9
Összesen	**32**	**26**	**17**	**37**

Az 1. Táblázat szemlélteti az említett szimulációs eszközök főbb tulajdonságait objektív pontrendszerrel értékelve. A célirányosan felállított pontrendszer (1-10) skálán értékeli az összehasonlított számítógépes szimulációkat annak függvényében, hogy mennyire könnyen alkalmazhatóan és mennyire pontosan teszik lehetővé az adott kiemelt szempont témaköréhez tartozó problémák, feladatok megoldását. Az értékelés nem általános érvényű, azaz nem minősíti a szimulátorokat bármilyen szimulációalkotási feladat esetében. Az értékelés kifejezetten a jelen cikk során bemutatott járműszimulációs feladat szemszögéből készült.

A *CoppeliaSim* szimulátor ROS-kompatibilitása megoldható kiegészítő szoftvercsomagok alkalmazásával, azonban ezen megoldás alkalmazhatósága limitált, az ROS egyes üzenettípusai esetében nem elérhető. Dinamikus szimuláció létrehozását moduláris, könnyen paraméterezhető komponensekkel támogatja. Megjelenése kellően valósághű, a textúrák minősége átlagos. Szenzormodellek szempontjából szintén több beépített, paraméterezhető komponenst is tartalmaz a szimulátor, viszont az ROS-kompatibilitás ezen eszközök esetében is hiányos.

A *Gazebo* szimulátor ROS-integrációja kiemelkedő, ezen szimulátor esetében nem szükséges kiegészítő szoftver a kompatibilitás megvalósításához. A Gazebo szimulátor segítségével az alkalmazott fizikai motorok miatt kellően valósághű dinamikai szimulációk készíthetőek, az alacsony pontozást a nehéz kezelhetőség és paraméterezés indokolja. A szimulátor vizuális megjelenése átlagos, valósághű modellek összeállítása nehézkes. Az elérhető szenzormodellek emuláció és ROS-kompatibilitás szempontjából valósághűek.

Az *Unreal Engine 4* játékmotor, valamint a rá épülő *Microsoft AirSim* csomag járműszimulációk esetében jó alternatívát jelenthet a rendkívül élethű vizuális megjelenés miatt, főleg kamerát is tartalmazó feladatok esetében. Ez a lehetőség jelen szimulációs feladat esetében kevésbé releváns, ugyanis korlátozott számú szenzormodellt tesz elérhetővé, valamint ROS-kompatibilitása nem megbízható.

Az *SVL* szimulátor célirányosan önvezető funkciókkal rendelkező járművek gyors és egyszerű szimulációjára készült, ebből adódóan jelen feladat esetében minden szempontra kiválóan megfelel.

4. AZ SVL JÁRMŰSZIMULÁTOR ISMERTETÉSE

Az SVL (*LG Silicon Valley Lab*) szimulátor elsődleges célja az önvezető autókkal kapcsolatos kutatások és fejlesztések megkönnyítése. A Unity játékmotorra épül, amely lehetővé teszi valósághű járművek és környezetek megjelenítését relatív alacsony számítási kapacitás-igénnyel. A Unity alkalmazása által olyan technológiák érhetők el szimulációs célokra is, mint például a *High Definition Render Pipeline (HDRP)*, amely olyan render, amely a fizika törvényeire alapozott világítással kapcsolatos megoldások megvalósítását teszi lehetővé [2].

A szimuláció a következő elemekre bontható: környezet, szenzorok, jármű (kinematika-dinamika és irányítás). A környezet szimulációja lehetővé teszi ütközési modellel rendelkező valósághű környezeti elemek (épületek, növényzet stb.) modellezését, lehetőség van tesztpályák, városrészek teljes modellezésére. Lehetőség van továbbá az infrastruktúra elemeinek modellezésre, mint közlekedési lámpák és táblák, vagy útelemek. A környezet esetében paraméterezhető az időjárás - eső, köd, napszak és a gyalogos- és járműforgalom. Az SVL szimulátor nyílt forráskódú szoftvertermék, szabadon felhasználható és továbbfejleszthető. Alkalmazása megköveteli a Unity játékmotor alkalmazását; az egyéni környezetek és járműmodellek a Unity Editor szerkesztő által készíthetők el. A járműmodellek szerkesztése futásidőben nem lehetséges, minden modellen belüli módosítás új modellkiadást (build) igényel. A kiadott modellek modellszerveren keresztül is elérhetők, vagy helyi tárhelyről is a szimulátorhoz kapcsolhatók. A modellekkel ellentétben a szenzorok paraméterezése, módosítása új modellkiadás nélkül is megoldható. A modellezett járműveken elhelyezett szenzorokat *JavaScript Object Notation (JSON)* kód formájában paraméterezhetjük. Az emulált adatokat szolgáltató érzékelők paraméterezhető tulajdonságainak listája megegyezik a valós szenzorok paramétereivel.

Az SVL szimulátor számos önvezető járművek fejlesztése során használatos rendszerrel kompatibilis, például: *Apollo, CyberRT, ROS*. Jelen feladat szempontjából kiemelkedően fontos a szimulátor *ROS (Robot Operating System)* rendszerrel való kompatibilitása, ugyanis a fejlesztendő önvezetési és teleoperációs megoldások is ROS alapokkal fognak rendelkezni. Az ROS (Robot Operating System) rugalmas keretrendszer robotszoftverek létrehozásához. Számos eszközt, könyvtárat és általánosított eljárást foglal magába, melyek segítik komplex robotok, önvezető járművek platformfüggetlen, moduláris fejlesztését [3]. Általános célú robotszoftver fejlesztése a számos hardvertípus miatt rendkívül nagy kihívás. Az ROS lehetővé teszi robot hardverelemeinek önálló fejlesztését, és a standardizált elemközi kommunikációs eljárásainak segítségével biztosítja az így fejlesztett hardverelemek kompatibilitását. Járműszimulációk fejlesztése során is célszerű az ROS alkalmazása, ugyanis a szimulátor segítségével fejlesztett és tesztelt kódrészek adott esetben módosítás nélkül implementálhatók a valós járművön.
Az ROS gráf rendszerű, a csomópontok (node) közvetlenül kommunikálhatnak egymással. Publisher/Subscriber típusú kommunikáció valósul meg közöttük. A node-ok közölhetik (publish) üzenetüket, a többi node feliratkozhat (subscribe) erre, így minden node kommunikálhat minden node-dal [4].
Az ROS jelenleg nem valós idejű, de ezen hibája kiegészítő eszközök alkalmazásával megoldható. További hiányossága a biztonsági kérdések megoldatlansága. Mindkét hiányosság megoldásra kerül a ROS 2 változat megjelenésével. Az SVL szimulátor ROS 2 támogatottsággal is rendelkezik.

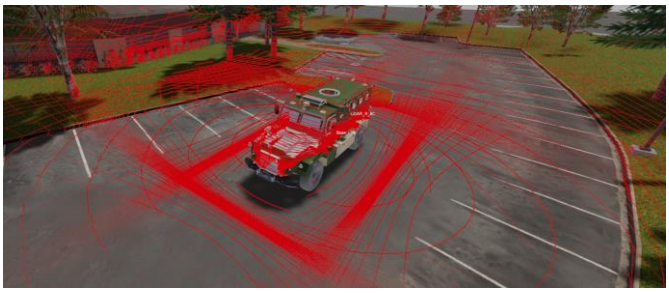
5. A GAMMA KOMONDOR JÁRMŰ SVL SZIMULÁCIÓJA

A szimuláció alanyát képező jármű a Gamma Komondor járműcsalád tagja (2. ábra). A szimulációalkotás célja a következő fedélzeti érzékelők alkalmazásának vizsgálata:

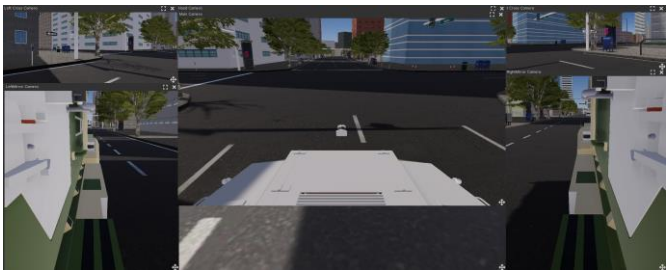
- Ouster OS1 vagy OS0 lidar érzékelők
- Allied Vision Mako színes monokamerák

2. ábra. A Gamma Komondor járműcsalád egyes járművei

A szimuláció célja a felsorolt szenzorok adatainak valósághű kinyerése, kiegészítve a jármű alacsonyszintű irányításának valósághű modellezésével, mind szenzoros, mind pedig aktuátoros oldalról. Ez azt jeleni, hogy a jármű CAN hálózatán történő adatforgalom is a szimuláció tárgyát képezi, cél ezt is élethűen kezelni.
A cikk célja ismertetni a járműmodell-alkotási feladat folyamatát SVL szimulátorban. Bemutatásra kerül a CAD modell előkészítése, a szenzorok modellezése, paraméterezése, valamint a szimulátor kommunikációs eszközökkel való kompatibilissá tétele. SVL szimulátor esetében a szimulált jármű összeállítása Unity Editor alkalmazásával lehetséges. A modellalkotás első lépése a megfelelő CAD modell előkészítése, amely jelen estben egy megfelelő elemeket tartalmazó Filmbox (.fbx) fájl. A fájl kötelező elemei a járműkarosszéria, a járműkarosszériához tartozó ütközési modell, valamint a kerekek. Ezen kívül a jármű alárendeltjeként szükséges a világítás elemeit, fényjelzéseket (index, fék stb.) is előkészíteni. Modellezés során kulcsfontosságú a megfelelő koordináta-rendszer alkalmazása.
Importálást és a textúrák megfelelő elhelyezését követően kialakítható a szimulált jármű. A kerekek esetében fontos megemlíteni, hogy a modellezés során járműhöz rendelt kerekek csak a valósághű vizuális megjelenésért felelnek, a kerekekhez saját osztály tartozik, melyeket a jármű hajtott kerekének és kormányzásának irányításához, beállításához használhatunk. Az osztály ütközési modelleket rendel a kerekek modelljeihez. A paraméterezés során több járműparaméter is megadható, beleértve a jármű tömegközéppontjára, futóművére és hajtásláncára vonatkozó adatokat. A jármű szenzorainak modellezése során fontos a tudatában lenni annak, hogy az SVL esetében a modellalkotási fázis nem tartalmaz működő szenzorokat, csak a várhatóan elhelyezendő szenzorok modelljét szükséges a járművön elhelyezni a valósághű megjelenés biztosítása érdekében. A szenzorok pozíciója és működése a JSON kód segítségével adható meg. A szenzorok az ROS-kompatibilis működés mellett meg is jeleníthetők az SVL szimulációs környezetben (3. ábra). A távvezérlés funkcióhoz használandó kamerakép megjelenítésére felhasználói felület készült (4. ábra).



3. ábra. A lidar szenzorok vizualizációja



4. ábra. A fedélzeti kamerák képének megjelenítése

6. ÖSSZEFOGLALÓ

Jelen cikk célja olyan számítógépes szimulációs eszköz biztosítása volt, amely lehetővé teszi a járműmentes fejlesztést teleoperációs és önvezetési feladatokra alkalmas katonai és katasztrófavédelmi terepjáró esetén. Bemutatásra kerültek a célra alkalmas számítógépes szimulációs eszközök, majd megtörtént objektív szempontrendszer alapján az ideális szoftver kiválasztása. Bemutatásra került továbbá a kiválasztott szoftver, az SVL autonóm jármű szimulátor, és a használával történő modellalkotás. A jármű szenzoros és aktuátoros elemeinek megfelelő szimulációja lehetővé tette a megnevezett fejlesztések során szükséges szenzorok és szenzorpozíciók kijelölését, valamint a teleoperációra és önvezetési feladatokra vonatkozó szoftveres fejlesztések indítását.

KÖSZÖNETNYILVÁNÍTÁS

A cikk kutatásaihoz a „Modulrendszerű, önvezérlésre alkalmas, távvezérelhető, könnyű páncélvédettségű félplatós terepjáró katonai és katasztrófavédelmi célú terepjáró bázisjármű és kapcsolódó cserefelépítmények fejlesztése (2020-1.1.2-PIACI-KFI-2020-00065)” projekt és a Széchenyi István Egyetem biztosított forrást.

IRODALOMJEGYZÉK

[1] R. E. Shannon, „Introduction to the art and science of simulation,” in *1998 Winter Simulation Conference. Proceedings*, Washington, DC, USA, 1998.

[2] „Unity Manual,” 10 2020. [Online]. Available: https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@10.1/manual/index.html. [Hozzáférés dátuma: 2021.09.15.].

[3] „About ROS,” Open Source Robotics Foundation, [Online]. Available: https://www.ros.org/about-ros/. [Hozzáférés dátuma: 2019.08.15.].

[4] S. Cousins, „Welcome to ROS Topics [ROS Topics],” *IEEE Robotics & Automation Magazine,* 1. kötet 17, 1. szám1, pp. 13-14, 2010.