

LIDAR MODELLEZÉS AUTONÓM JÁRMŰVEK RENDSZERSZINTŰ TESZTELÉSÉHEZ

LIDAR MODELLING FOR SYSTEM LEVEL TEST OF AUTONOMOUS VEHICLES

Erőrdi Zakariás*, Papp János, Bári Gergely

Járműtechnológia Tanszék, GAMF Műszaki és Informatikai Kar, Neumann János Egyetem, Magyarország

Kulcsszavak:

Lidar
Szimuláció
Validáció
Autonóm
Önvezető

Keywords:

Lidar
Simulation
Validation
Autonomous
Self-Driving

Cikktörténet:

Beérkezett 2018. október 24.
Átdolgozva 2019. február 24.
Elfogadva 2019. március 19.

Összefoglalás

Az egyes algoritmusok fejlesztésére különféle keretrendszerek állnak rendelkezésre (ROS, MATLAB/SIMULINK), melyekhez különféle szimulációs környezetek léteznek, viszont az autonóm rendszer, rendszerszintű tesztjére általában csak a tesztpályán kerül sor. Célunk egy olyan szimulációs környezet kialakítása, melyben az autonóm rendszert teljes egészében tesztelni lehet az érzékeléstől kezdve, az aktuáláson át a járművet menet közben befolyásoló különböző dinamikai hatásokig. Jelen kutatás egy ilyen lehetséges szimulációs környezetet mutat be, ezen belül kiemelve egy lidar szenzor modellezési és validálási folyamatát, illetve, hogy a szenzor modellezése hogyan segíthet az önvezető algoritmusok rendszerszintű fejlesztésében.

Abstract

There are various frameworks available for the development of each algorithm (ROS, MATLAB / SIMULINK) for which different simulation environments exist, but the autonomous system's system test usually only takes place on the test track. Our goal is to create a simulation environment in which the autonomous system can be fully tested from perception, through actuation with various dynamic impacts affecting the vehicle while driving. This paper presents a possible simulation environment, highlighting the modeling and validation process of a lidar sensor and how the sensor model can help system-level development of self-driving algorithms.

1. Bevezetés

Az autonóm járművekkel kapcsolatos kutatások már a 90-es években kezdetüket vették, az elmúlt pár évben pedig számtalan, a témához kapcsolódó tanulmány született. A közúti járművekben is kezd szépen lassan megjelenni ez a technológia, elég hogyha csak az olyan vezetőt segítő rendszerekre gondolunk, mint a sávkövetés, holttér érzékelés vagy az automatikus távolságszabályozó. [1] A fejlődési irány is ki van jelölve, teljesen önvezető autó létrehozása, amely el tud végezni bármilyen körülmény között bármilyen a feladatot, amit egy sofőr tudna. [2]

Ehhez a célhoz vezető algoritmus fejlesztés során pedig elengedhetetlen a tesztelés. Az autonóm rendszer rendszerszintű tesztelésére a legkézenfekvőbb mód természetesen az éles tesztelés, tesztautón, viszont ez rendkívül költséges, elég, hogyha csak az autóra felszerelendő

* Kapcsolattartó szerző. Tel.: +36 301974973
E-mail cím: zacherosdi@gmail.com

szenzorokra, aktuátorokra gondolunk. Jogilag is korlátozottak a dolgok, ugyanis különböző engedélyek nélkül nem lehet önvezető autót a közutakon tesztelni. [3] Továbbá számos esetben a fejlesztés során a tesztelés reprodukálhatósága elengedhetetlen, elég hogyha csak a behangolandó szabályzóparaméterekre vagy az algoritmus különböző - szélsőséges körülmények közötti viselkedésére gondolunk. Ezek miatt a jövőben kiemelt szerepe lesz a szimulációnak.

Jelenleg számos szimulációs program létezik, viszont kimondottan autonóm járművek fejlesztésére szánt környezet még nincs a piacon. Van amelyik program kimondottan az autó menetdinamikai vizsgálatára van kifejlesztve, mint például az IPG CarMaker, van ami kimondottan a járműveken használatos érzékelő szenzorok modellezésére lett tervezve, például a Pro-SIVIC. Van olyan aminek a közúti forgalom szimulálása a célja, mint például a SUMO [4], illetve vannak a különböző, robotok fejlesztésére szánt szimulációs környezetek, mint a például a Gazebo, ami integrálódik a manapság leginkább használatos robot-fejlesztési keretrendszerbe, a ROS-ba.[5]

A választott szimulációs környezet számos szigorú követelménynek kell megfeleljen és egy modellezési folyamat után validálni is kell a kapott eredményeket. Egyáltalán nem egyértelmű a választás, viszont hogyha valóban az a cél, hogy a robot minden körülmény között leválassza az embert akkor elengedhetetlen az autó menetdinamikájának a pontos szimulálása is, hiszen a hirtelen, váratlan helyzetekben is megfelelő modellre van szükségünk, elég hogyha csak olyan esetekre gondolunk, amikor az autó a tapadási határ környékén van például egy hirtelen irányváltás, vagy vészfék esetén. Ezekre pedig a CarMaker egy megfelelő választásnak tűnik.[6] Továbbá az IPG szoftverének újabb verzióiban már különböző érzékelő szenzorok modellezésére is lehetőség van, így ebben a környezetben kivitelezhető lehet az önvezető autók rendszerszintű szimulációja.

A továbbiakban CarMaker által biztosított szimulációs szoftver által nyújtott szenzor modellezés lesz bemutatva, egy LiDAR szenzor példáján, valós mérési eredményekkel összehasonlítva, illetve ennek a modellnek az autonóm rendszer fejlesztésébe való integrációja.

2. Szenzor modellezés

A LiDAR szenzor a kibocsátott lézerefény hullámhossza, illetve a kibocsátott és visszaverődő fény fáziskülönbsége alapján érzékeli a szenzor közelében elhelyezkedő objektumokat nagy pontossággal. Az autonóm járművek fejlesztésénél a LiDAR szenzor az egyik alapfelszerelés a kifejezetten tesztelésre szánt autók körében [1], ahol egyik legelterjedtebb algoritmus, amihez használják a szenzort, az a SLAM (simultaneous localization and mapping). [7]

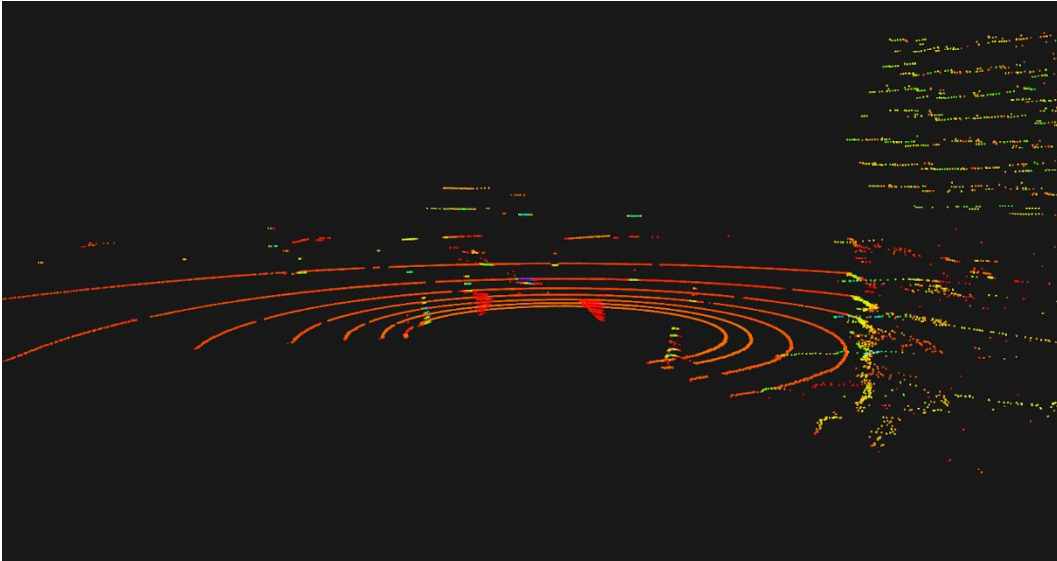
2.1. LiDAR Mérés

A modellezett LiDAR egy Velodyne VLP-16 volt, mellyel először méréseket végeztünk.

1. Táblázat. A szenzor fontosabb adatai

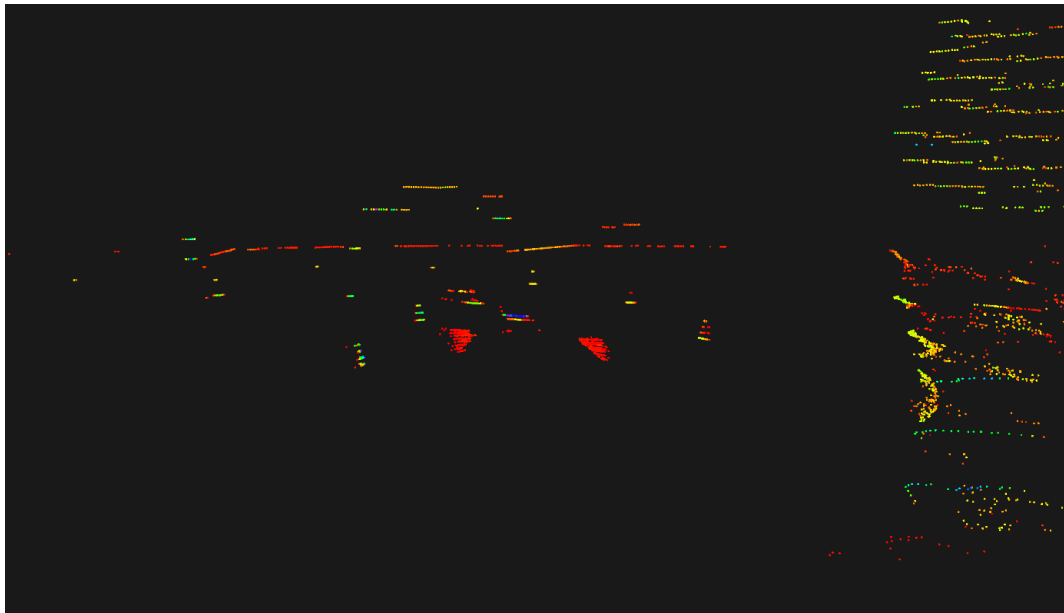
Mérési tartomány	100 m-ig
Pontosság	+/- 3 cm
Függőleges látószög	30° (+/-15°)
Szögfelbontás (függőleges)	2°
Szögfelbontás (vízszintes)	0.1° - 0.4°

A szenzor az autó elejére volt felszerelve, és egy bójákkal kirakott pályán teszteltük az érzékelését. A kapott eredményeket szerettük volna reprodukálni a szimulációs környezetben.



1. ábra. Raw data

A fenti képen látható a különböző objektumokról visszaverődő fénysugarak alapján létrejövő pontfelhő. A különböző színek különböző fényintenzitáshoz tartoznak. Az első szembetűnő dolog a sík érzékelése, ugyanis az autó vízszintes tengelyéhez képest lefelé induló fénysugarak mind visszaverődnek a talajról, ennek a jelenségnek a modellezése viszont nem feltétlenül szükséges, ugyanis a síkszűrés egyszerűen megvalósítható, és elég csak a ténylegesen az autó útjában álló objektumokkal foglalkozni.



2. ábra. Pontfelhő a síkszűrés után

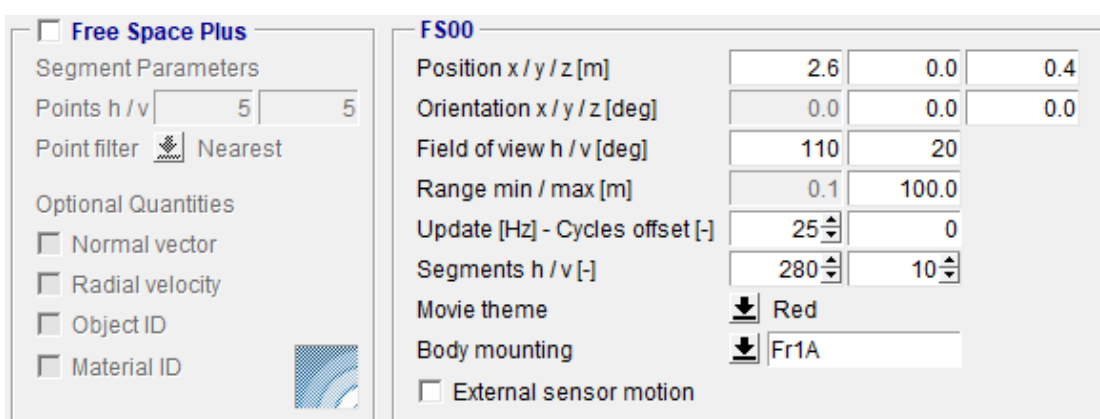
A síkszűrés utáni képen már kivehetőek a bóják, látható egy épület a kép előterében, a bóják között az autó vázáról visszaverődő pontok piros színnel, illetve a távolból érkező zajok. Egy ehhez hasonló pontfelhő reprodukálása az, amit a szimulációs környezettől elvárunk.

3. Szimulációs környezet

3.1. Carmaker

A Carmaker nyújt számunkra egy alap járműmodellt, melyet tetszés szerint paraméterezhetünk. Ha a modell valamelyik részével nem lennénk elégedettek, akkor pedig ezt akár teljesen újra is írhatjuk más környezetben, ami aztán tud integrálódni a Carmakerben. A legkézenfekvőbb környezet ehhez a MATLAB/Simulink, ugyanis ez a program eleve a modell alapú fejlesztésre lett kitalálva [8], egyszerűen integrálható Carmakerbe [9], illetve manapság a szabályzókat többnyire Simulinkben fejlesztik [10]. Az autonóm algoritmusok fejlesztése lényegében úgy tud megvalósulni, hogy a jármű-vezető modelljét írjuk meg mi magunk. Egy ilyen modell bemenete a szenzoradatok, kimenete pedig az aktuátorok alapjele. A Munkám során Carmaker által felkínált LiDAR modellt elemeztem.

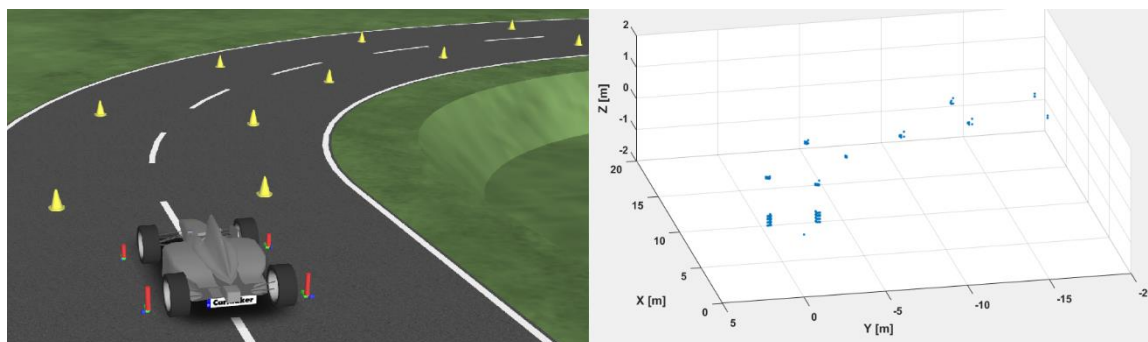
3.2. LiDAR modellezés Carmakerben



3. ábra. Lidar modell változtatható paraméterei

Megadhatjuk a szenzor elhelyezkedését az autón, illetve egyéb paramétereket, amiket az 1)-es táblázatban kiemeltem, viszont az alap modellnek van egy komolyabb megkötése. Ahhoz, hogy a szimuláció sebesség ne csökkenjen drasztikusan, a pontfelhő nem állhat több, mint 3000 elemből. Ezt a Segments h/v mezőben lehet megadni. Ahhoz, hogy a szenzor felbontását meg tudjam tartani, a függőleges látószöveget 20° -ra, a vízszintes látószöveget 110° -ra csökkentettem, amiket rendre 10 illetve 280 részre oszt fel a szenzor, így jön ki a 2° -os függőleges felbontás, és a 0.4° -os vízszintes felbontás. Rendelkezésre áll egy bonyolultabb szenzormodell is, a free space sensor plus, viszont ez külön szoftverintegrációt igényel.

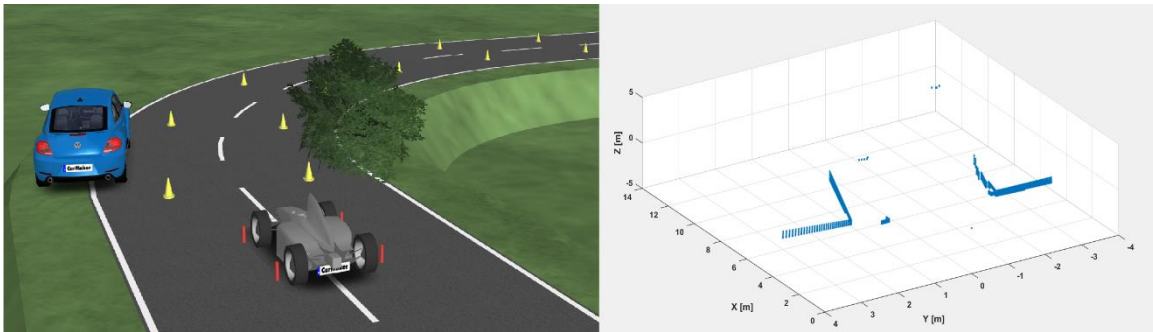
A felparaméterezett szenzor alapvetően előre elkészített objektumokat képes érzékelni, olyan objektumokat, amivel az autó a közutakon is szembetalálkozhat, viszont 3D modellező programmal készíthetünk mi is saját objektumot. Első körben a bóják lettek bemodellezve, illetve a pályán kirakott teszt esetet igyekeztünk szimulálni.



4. ábra. Szimulált pálya és a kapott pontfelhő

A kapott pontfelhő megfelelő, egyszerű bójadetektáló algoritmust lehet fejleszteni ebben a szimulációs környezetben, ami alapján az autó autonóm módban elnavigálhat a bóják között. Adódik a kérdés viszont, hogy a bonyolultabb közlekedési szituációkat hogyan kezeli le a program, ugyanis az alap szenzormodell az objektumokat hasábokkal közelíti. Látható a 4. ábrán is, ahogyan a bója egyszerűsítve van.

Ennek érdekében, a bójás pályát kiegészítettük egyéb tereptárgyakkal, ami egy személyautó közúton történő közlekedése közben gyakorta előjön. Parkoló autó illetve növényzet hozzáadása után is megnéztük a szimuláció eredményeit.



5. ábra. Eredmények bonyolultabb közlekedési szituáció esetén

Bonyolultabb közlekedési szituációkban már jól látszanak a modell korlátai, viszont alap útvonal-kereső algoritmusok fejlesztésére, egyszerűbb SLAM, illetve térképező algoritmusok fejlesztésére megfelelő ez a szimulációs környezet.

Járókelők, valós, veszélyes közlekedési helyzetek modellezésére viszont már nem alkalmas a modell. Az 5. ábrán látható, például ahogy nem lehet különbséget tenni a kapott pontfelhő alapján a parkoló autó, illetve a bokor között. Olyan döntési helyzetek elé nem lehet tehát állítani az algoritmust, amikor annak morális döntést kellene hoznia. [11]

4. Összefoglalás

Az a cél, hogy egy olyan szimulációs környezetet találjunk, amiben az autonóm járművet „vezető” algoritmus tesztelhető az érzékeléstől kezdve az aktuálásig. Az IPG által felkínált Carmaker lehetőségeit vizsgáltuk, azon belül is a LiDAR modellezést. A kapott eredmények alapján már tesztelhetőek, fejleszthetőek különböző útvonal tervező algoritmusok, viszont a modell egyszerűségéből adódóan bonyolultabb közlekedési helyzetek nem modellezhetőek.

Az egyik fejlesztési lehetőség, hogy integráljuk a MATLAB/Simulinkbe a Carmaker másik, bonyolultabb lidar modelljét. Ez már leköveti az egyes objektumok geometriáját, nem egyszerűsíti azokat hasábkkal, illetve egyéb paramétereket is ad az egyes pontokról (pl. Intenzitás)

Egy másik fejlesztési lehetőség a kamerák modellezése, és akkor már tesztelhető szenzorfüzió, képdetektáló algoritmus, illetve így a szimulációs környezet méginkább le fogja fedni az igényt, amit az autonóm járművek tesztelése kíván.

Köszönetnyilvánítás

A cikk kutatásaihoz az Új Széchenyi Terv keretein belül az EFOP-3.6.2-16-2017-00016 számú projekt biztosított forrást. A kutatás az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Irodalomjegyzék

- [1] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, Homayoun Najjaran, Autonomous vehicle perception: The technology of today and tomorrow, *Transportation Research Part C: Emerging Technologies*, Volume 89, 2018, Pages 384-406, ISSN 0968-090X, <https://doi.org/10.1016/j.trc.2018.02.012>.
- [2] Federal Automated Vehicles Policy, Accelerating the Next Revolution In Roadway Safety, September 2016 , Page 9
- [3] Nynke E. Vellinga, From the testing to the deployment of self-driving cars: Legal challenges to policymakers on the road ahead, *Computer Law & Security Review*, Volume 33, Issue 6, 2017, Pages 847-863, ISSN 0267-3649, <https://doi.org/10.1016/j.clsr.2017.05.006>.
- [4] ssia Belbachir, Jean-Christophe Smal, Jean-Marc Blosseville, Dominique Gruyer, Simulation-Driven Validation of Advanced Driving-Assistance Systems, *Procedia - Social and Behavioral Sciences*, Volume 48, 2012, Pages 1205-1214, ISSN 1877-0428, <https://doi.org/10.1016/j.sbspro.2012.06.1096>.
- [5] Mostafa Sharifi, XiaoQi Chen, Christopher Pretty, Don Clucas, Erwan Cabon-Lunel, Modelling and simulation of a non-holonomic omnidirectional mobile robot for offline programming and system performance analysis, *Simulation Modelling Practice and Theory*, Volume 87, 2018, Pages 155-169, ISSN 1569-190X, <https://doi.org/10.1016/j.simpat.2018.06.005>.
- [6] Widner, A & Bári, G. (2018). Investigating the effects of roll center height in simulation, for safety-margin research. *IOP Conference Series: Materials Science and Engineering*. 393. 012034. 10.1088/1757-899X/393/1/012034.
- [7] Tobias M. Quack, Matthias Reiter, Dirk Abel, Digital Map Generation and Localization for Vehicles in Urban Intersections using LiDAR and GNSS Data, *IFAC-PapersOnLine*, Volume 50, Issue 1, 2017, Pages 251-257, ISSN 2405-8963, <https://doi.org/10.1016/j.ifacol.2017.08.042>.
- [8] <https://www.mathworks.com/products/simulink.html>
- [9] Abhishek Bhat, Akshay Vishwanath Malghan, Modelling of propulsion system for complete vehicle verification through simulation, *Master's Thesis 2016:72*, ISSN 1652-8557, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology
- [10] A.S. Martyanov, E.V. Solomin, D.V. Korobotov, Development of Control Algorithms in Matlab/Simulink, *Procedia Engineering*, Volume 129, 2015, Pages 922-926, ISSN 1877-7058, <https://doi.org/10.1016/j.proeng.2015.12.135>.
- [11] Yochanan E. Bigman, Kurt Gray, People are averse to machines making moral decisions, *Cognition*, Volume 181, 2018, Pages 21-34, ISSN 0010-0277, <https://doi.org/10.1016/j.cognition.2018.08.003>.