Safety Assessment and Simulation of Autonomous Vehicle in Urban Environments

Mohsen Malayjerdi, Barış Cem Baykara, Raivo Sell and Ehsan Malayjerdi

Mechanical and Industrial Engineering Department, Tallinn University of Technology, Tallinn, Estonia
momala@taltech.ee

Abstract. The Autonomous Vehicle (AV) industry aims to design strategic plans to ensure the safety of the developed systems before their mass deployment. Real-road testing is shown to be impractical for validating these systems as it requires many years if not decades of testing in different environmental conditions. Therefore, this method must be complemented with simulation for a feasible solution. The primary goal of this research is to develop advanced techniques in the safety validation area by using immersive simulation technologies. This study led to a simulation approach for safety evaluation of an AV shuttle, iseAuto, currently operating at the TalTech campus. First, we create a virtual environment based on the specified path on the university campus including all relevant features. The environment is created by using geospatial data which is collected by a drone, then they are converted to a 3D map applicable for the LGSVL simulator. Next, the developed shuttle 3D model is imported to the simulation environment and with the aim of the LGSVL, sensor data are provided for the Autoware perception algorithms, which is the control software of the shuttle. This system enables us to evaluate AV's decision-making performance and safety in different situations.

1. Introduction

Autonomous vehicles development is one of the top trends in the automotive industry and the technology behind them has been evolving to make them safer. In this way, engineers are facing new challenges especially moving toward the Society of Automotive Engineers (SAE) levels 4 and 5. To put Autonomous Vehicles (AV) on roads and evaluate the reliability of their technologies they have to be driven billions of miles [1], which takes a long time to achieve unless with the help of simulation. Furthermore, due to the past real crash cases of AVs, a high-fidelity simulator has become an efficient and alternative approach to provide different testing scenarios for control of these vehicles, also enabling for safety validation before real road driving [2, 3, 4, 5]. Different high-resolution virtual environments can be developed based on the real world for simulators by using cameras or lidars to simulate the scenarios as close as possible to the real [6]. Also, virtual environment development enables us to customize and create various urban backgrounds for testing the vehicle. Creating a virtual copy of an existing intelligent system is a common approach nowadays called digital twin [7, 8].

In this paper, we focus on the utilization of simulation for an AV shuttle at Tallinn University of Technology (TalTech), Estonia. TalTech AV research group is well-known for its AV shuttle, iseAuto [9] that is operational in the campus for experimental research purposes (figure 1). The
vehicle is designed and developed from scratch by implementing previously proposed mechatronic design methodology [10, 11, 12] with a special focus on early design stages. The first prototype development was a joint venture with TalTech and local industry Silberauto [13]. This shuttle is controlled by Autoware [14], a Robotic Operating System (ROS) based platform for self-driving vehicles.

The overall research project is planned to be executed in two stages. First, the virtual environment was built from the campus AV road area, where most of our real experiments take place, to build the simulation framework. We use geospatial images to create the environment as a Unity terrain. Among different modern AVs simulators like CARLA [15], LGSVL [16] and Gazebo we took LGSVL as our simulator due to compatibility with our control software (Autoware) and our terrain generation platform, Unity. Second, creating different scenarios and performing software-in-the-loop (SIL) simulation by connecting Autoware with LGSVL. This enables us to find better sensors configuration and settings in addition to the verification of the decision-making system that leads to safety assessment.

2. Simulator

Simulation has been widely used in vehicles manufacturing, particularly for mechanical behaviour and dynamical analysis. However, AVs need more than that due to their nature. Simulation in various complex environments and scenarios included other road users with different sensors combination and configuration enables us to verify their decision-making algorithms. One of the most popular robotic simulator platforms is Gazebo. It is based on ROS and utilizes physics engines and various sensor modules suitable for autonomous systems. Nevertheless, Gazebo lacks modern game engines features like Unreal and Unity which gives the power to create a complex virtual environment and realistic rendering.

On the other hand, CARLA and LGSVL are modern open-source simulators based on the game engines, Unreal and Unity respectively, which also have good compatibility with our AV stack Autoware. Although, comparing these two is beyond our discussion but we selected the LGSVL as our simulator because of compatibility with our terrain generator which is Unity.

Figure 2 shows a full map of the simulation workflow and relation between Autoware and the simulator. Vehicle 3D Model and the virtual environment, which are created inside the unity, are imported to the simulator. The simulator allows customizing the environment to create different scenarios such as adding/removing other road users, putting traffic systems and adjusting the time of day and the weather of the scene. Then, virtual sensors provide information for the perception of the environment. This information is transferred via a ROS bridge to our control software platform to use in our perception algorithms for localization and detection. Perception results are used in the Autoware planning section which makes the control commands for the

![Figure 1. TalTech iseAuto - an AV shuttle](image-url)
vehicle. These control commands are sent back to the simulator via the ROS bridge to navigate the vehicle inside the simulator.

![High-level architecture of simulation and the AV system](image1)

**Figure 2.** High-level architecture of simulation and the AV system

The iseAuto 3D model and its lidar sensors are illustrated in figure 3. Two Velodyne VLP-16 are installed at the top front and back of the vehicle. Furthermore, two Robosense Bpearl are installed at the sides left and right of the vehicle. This lidar configuration creates a good point-cloud coverage around the vehicle for perception purposes.

![iseAuto simulated Model with different lidars installed](image2)

**Figure 3.** iseAuto simulated Model with different lidars installed

### 3. Virtual Environment Creation

Nowadays the fierce competition in the gaming industry has brought many features to the table in terms of game engines. These engines can simulate physics and thus can be exploited as simulators aside from game development. LGSVL and others have already taken advantage of these engines and have created a framework for testing autonomous vehicles within these physics simulators. However, even though these simulators provide some basic tools and assets to get started, it is not enough. To make it more realistic, we need real life terrains simulated.

#### 3.1. Data Collection and Processing

Capture of aerial imagery with a drone, over the area to be mapped, has to be conducted. The images are captured at a grid based flight path. This ensures that the captured images contain different sides of a subject. In order to make sure the images have maximum coverage, the flight path is followed three times in different camera angles but at a constant altitude. Taking aerial photo is one of the most important steps in the mapping process as it will significantly affect the outcome of the process and the amount of work to be done to process those images.
There are also external factors that may affect the quality of the pictures taken off the ground. Weather conditions and scene lighting may create artifacts on the pictures that may disturb the photogrammetric process. The images taken are georeferenced by the drone and if necessary a stationary RTK device can be utilized to mitigate errors and shifting on the positioning data stamped on the pictures. The onboard IMU provides the pictures with the orientation so that later they can be stitched together and used for photogrammetric processing. Third party software aligns and creates the dense point-cloud from the pictures that were captured. Once the dense point-cloud is created, the segmentation and classification of the points is needed in order to separate unwanted objects and vegetation from the point-cloud data. However removing is not to be done in the point-cloud as the positional information they provide for their respective objects will aid terrain generation to spawn details. Figure 4 shows the three main steps to generate the Unity train from geospatial data.

![Figure 4. Steps for virtual environment generation](image)

3.2. Terrain generation

Digitalization of a real life environment can be used for simulating AVs in countless different scenarios without taking the vehicle out for once. Terrain generation from point-cloud is done right in Unity. In-house developed plugin reads a pre-classified point-cloud file and based on chosen parameters it creates a normal map, a heightmap and a color map to use in conjunction with the unity’s terrain engine to create realistic environments.

4. Simulation and Safety Assessments

Based on the simulation architecture illustrated in figure 2, we can run the AV inside the virtual environment. In our collaboration with Florida Polytechnic University and Embry-Riddle Aeronautical University, we developed a regime for creating edge-case scenarios for safety validation of the shuttle working on our campus pilot road [17]. Now, by using a high-fidelity simulator we can simulate different scenarios close to real to evaluate the control algorithm performance and safety. In terms of defining these scenarios, LGSVL provides a python API for spawning different objects like cars and pedestrians inside the virtual environment with different motion plans.

Figure 5 shows iseAuto facing a stopped Non-Player Character (NPC) vehicle that is spawned in front of the AV. Picture (a) is inside the LGSVL environment while picture (b) shows the lidar perception of the environment in RViz visualization tool. There is no filtering applied on this point-cloud; therefore, everything is mixed together and it is hard to distinguish objects for later processings. One of the challenging topics of self-driving development is overtaking. The way that the AV should decide for this mission and the risks that it faces is under study. Our experience with the vehicle trying to pass a stopped npc or an object has led us to focus on this topic more. This way, simulations can help us first, to improve our perception and detection system then, the mission and motion planning for a safe overtake. The first steps for detection...
are filtering and clustering the point-cloud. Autoware has some predefined features for them. One of the common point-cloud filterings is ground filtering in which some part of the point-cloud that is defined as ground will be separated. Each lidar point-cloud can be filtered separately or once after the concatenation with other lidars. Filtering parameters have an intensive effect on the detection result. Sometimes losing 10 to 20 points due to the improper filtering will result in the object not to be detected.

Filtering and clustering are illustrated in figure 6. Filtering is applied to the picture (a) and the ground, which was detected by lidars in figure 5 (b), is almost removed from the point-cloud. However, the npc points remained and they were clustered as an object in the picture (b). Filtering accuracy results in high performance object detection and safe decision making [18]. For example, Figure 7 shows how different ground filtering parameters can change maximum distance for detecting a stopped npc in front of the AV shuttle, although, both cases have similar clustering parameters. Picture (b) shows that the npc is detected by the AV shuttle from 32 meters distance but picture (c) shows that the maximum distance that it is able to detect an object decreased to 18 meters. The more distance we have for detection, the more time we have
for making a smooth control decision. In AVs with multiple lidars, filtering accuracy can be improved by performing it before point-clouds concatenation.

5. Conclusion
Safety validation is crucial for most of the AVs developments and deployments. The simulation as a validation approach in this paper offers a practical and effective way to evaluate the safety in different levels. This paper presents the simulation architecture of the iseAuto with SiL testing that shows how the virtual environment and vehicle model are used in combination with Autoware to simulate different scenarios. The development and utilization of this testing scheme will enable the safety improvement and autonomous vehicles performance development.

Acknowledgments
The research is supported by the EC H2020 project Finest Twins (grant No. 856602).

References
[1] Kalra N and Paddock S M 2016 Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice* 94 182–193
[2] Chao Q, Jin X, Huang H W, Foong S, Yu L F and Yeung S K 2019 Force-based heterogeneous traffic simulation for autonomous vehicle testing 2019 International Conference on Robotics and Automation (ICRA) (IEEE) pp 8298–8304
[3] Aeberhard M, Rauch S, Bahram M, Tanzmeister G, Thomas J, Pilat Y, Homm F, Huber W and Kaempchen N 2015 Experience, results and lessons learned from automated driving on germany’s highways IEEE Intelligent transportation systems magazine 7 42–57
[4] Anderson S J, Peters S C, Plutti T E and Iagnemma K 2011 *Robotics Research* (Springer) pp 39–54
[5] Razdan R, Lumina J, Balachandran A, Cheng C, Sreenivas S, Fernando X, Taiber J, Kalia A, Keel N, Zuby D, Krishnan K, Langer D and Sell R 2019 Unsettled technology areas in autonomous vehicle test and validation Tech. rep. SAE Technical Paper
[6] Malayjerdi M, Kuts V, Sell R, Otto T and Baykara B C 2020 Virtual simulations environment development for autonomous vehicles interaction IMECE2020 (ASME)
[7] Kuts V, Modoni G E, Otto T, Tähmemaa T, Bondarenko Y and Wang R 2019 Synchronizing physical factory and its digital twin through an iiot middleware: a case study. *Proceedings of the Estonian Academy of Sciences* 68
[8] Kuts V, Cherezova N, Sarkans M and Otto T 2020 Digital twin: industrial robot kinematic model integration to the virtual reality environment *Journal of Machine Engineering* 20
[9] Sell R, Leier M, Rassölkin A and Ernits J P 2018 Self-driving car iseauto for research and education 2018 19th International Conference on Research and Education in Mechatronics (REM) (IEEE) pp 111–116
[10] Christophe F, Sell R, Bernard A and Coatanéa E 2009 Opas: Ontology processing for assisted synthesis of conceptual design solutions International Design Engineering Technical Conferences and Computers and Information in Engineering Conference vol 49026 pp 249–260
[11] Sell R, Coatanéa E and Christophe F 2008 Important aspects of early design in mechatronic 6th International Conference of DAAAM Baltic Industrial Engineering, Tallinn
[12] Wang R, Sell R, Rassölkin A, Otto T and Malayjerdi E 2020 Intelligent functions development on autonomous electric vehicle platform *Journal of Machine Engineering* 20
[13] Rassölkin A, Sell R and Leier M 2018 Development case study of the first estonian self-driving car, iseauto *Electrical, Control and Communication Engineering* 14 81–88
[14] Kato S, Tokunaga S, Maruyama Y, Maeda S, Hirabayashi M, Kitsukawa Y, Monrroy A, Ando T, Fujii Y and Azumi T 2018 Autoware on board: Enabling autonomous vehicles with embedded systems 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS) (IEEE) pp 287–296
[15] Dosovitskiy A, Ros G, Codevilla F, Lopez A and Koltun V 2017 Carla: An open urban driving simulator arXiv preprint arXiv:1711.05906
[16] Rong G, Shin B H, Tabatabaei H, Lu Q, Lemke S, Možíček M, Boise E, Uhr G, Gerow M, Mehta S et al. 2020 Lgsvl simulator: A high fidelity simulator for autonomous driving arXiv preprint arXiv:2005.03778
[17] Medrano-Berumen C, Malayjerdi M, Akbaş M I, Sell R and Razdan R 2020 Development of a validation regime for an autonomous campus shuttle 2020 SoutheastCon (IEEE) pp 1–8
[18] Habermann D, Hata A, Wolf D and Osório F S 2013 3d point clouds segmentation for autonomous ground vehicle 2013 III Brazilian Symposium on Computing Systems Engineering (IEEE) pp 143–148