Proceedings of the Estonian Academy of Sciences,
2021, **70**, 4, 413–421
https://doi.org/10.3176/proc.2021.4.07
Available online at www.eap.ee/proceedings

MANUFACTURING
ENGINEERING

Autonomous vehicle safety evaluation through a high-fidelity simulation approach

Mohsen Malayjerdi[a*], Barış Cem Baykara[a], Raivo Sell[b,a] and Ehsan Malayjerdi[a]

[a] Mechanical and Industrial Engineering Department, Tallinn University of Technology, Tallinn, Estonia
[b] FinEst Twins Smart City Center of Excellence, Tallinn University of Technology, Tallinn, Estonia

Received 15 June 2021, accepted 13 July 2021, available online 2 November 2021

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. For solving this issue, the method should be complemented with simulation. The primary goal of this research was to develop advanced techniques in the safety validation area by using end-to-end simulation technologies. In this study, we present a simulation approach for safety evaluation of an AV shuttle, iseAuto, currently operating at the Tallinn University of Technology campus. We created a virtual environment by using geospatial data from the specified path on the university campus that includes all relevant features. Then, we converted the map to a 3D format applicable for the SVL simulator. Also, we provided the AV 3D model to use in the simulation and equipped it with the SVL virtual sensors to provide data for the Autoware perception algorithms, which is the control software of the shuttle. To show the efficiency of the proposed method, we designed two overtaking scenarios and observed the AV behaviour under the test. Finally, we demonstrate how the system enables us to evaluate AV's decision-making performance and safety in different situations.

Key words: autonomous vehicle, simulation, safety validation, high-fidelity simulator.

1. INTRODUCTION

Development of autonomous vehicles is one of the top trends in the automotive industry and the technology has been evolved to make them safer. Thus, engineers are facing new challenges, especially in moving toward Levels 4 and 5 of the Society of Automotive Engineers (SAE). To place autonomous vehicles (AVs) on roads and evaluate the reliability of their technologies, they have to be driven billions of miles [1]. It would take a long time to achieve this, 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 controlling these vehicles, also enabling safety validation before real-road driving [2–5]. Different high-resolution virtual environments can be developed for simulators by using cameras or lidars to simulate the scenarios as close to the real world as possible [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 a digital twin [7,8]. Extensive research and development, such as in [9,10] or [11], has been performed on AVs in recent years involving simulation. However, most of that has employed a low-fidelity simulator that cannot be a reliable reference for safety validations.

In this paper, we focus on the utilization of a high-fidelity simulator for an AV shuttle at Tallinn University

* Corresponding author, momala@taltech.ee

of Technology (TalTech), Estonia. The TalTech AV research group is well known for its AV shuttle, iseAuto [12], which is operational on the campus for experimental research purposes (Fig. 1). The vehicle was designed and developed from scratch by implementing the previously proposed mechatronic design methodology [13–15] with a special focus on early design stages. The first prototype development was a joint venture with TalTech and the local industry Silberauto [16]. This shuttle is controlled by Autoware [17], a Robotic Operating System (ROS) based platform for self-driving vehicles.

The overall research project was planned to be executed in two stages. First, the virtual environment was built based on the campus AV road area, where most of our real experiments take place, to create the simulation framework. We used geospatial images to generate the environment as a Unity terrain. Among different modern AV simulators such as CARLA [18], LGSVL (in 2021 the name was changed to SVL) [19] and Gazebo, we opted for SVL to be our simulator due to its compatibility with our control software (Autoware) and our terrain generation platform Unity. Another reason was to create different scenarios and perform software-in-the-loop (SIL) simulation by connecting Autoware with SVL. This enables us to find a better sensor 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 vehicle manufacturing, particularly for mechanical behaviour and dynamical analysis. However, AVs demand more due to their specific nature. Simulation in various complex environments and scenarios involving other road users with different sensor combinations and configurations 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 engine features such as Unreal and Unity, which give the power to create a complex virtual environment and realistic rendering.

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

Figure 2 shows a full map of the simulation workflow and different layers in the simulator as well as the control software (Autoware). Vehicle 3D model and the virtual environment, which were built inside Unity, were imported to the simulator. The simulator allows cus-

Fig. 1. TalTech iseAuto – an AV shuttle.

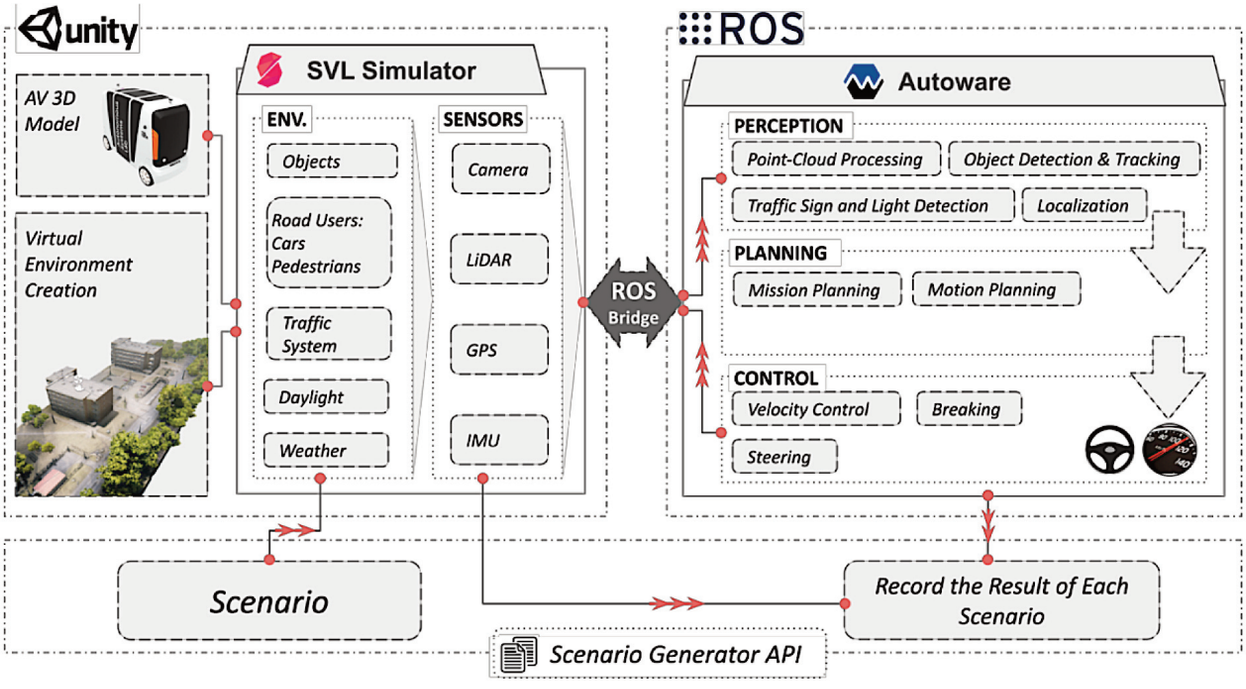

Fig. 2. High-level architecture of the simulation and the AV system.

tomizing the environment to create different scenarios such as adding/removing other road users, inserting traffic systems, adjusting the time of day and the weather of the scene. There is a scenario generator API that connects to the simulator and creates various scenarios according to the user definition. Then, the virtual sensors used in the AV provide information for the perception of the environment. This information is transferred via a ROS bridge to our control software platform to use in the perception algorithms for the localization and detection. Perception results are used in the Autoware planning section which makes the control commands for the AV. These control commands are sent back to the simulator via the ROS bridge to navigate the vehicle inside the simulator. Furthermore, in the case of any failure in any scenario, some sensor data and vehicle navigation commands are recorded for further study.

The iseAuto 3D model and its lidar sensors are illustrated in Fig. 3. A Velodyne VLP-32 was installed at the top front of the shuttle and a VLP-16 at the top back. Two Robosense Bpearl were installed at the left and right sides of the vehicle. Furthermore, to cover the blind zone in front of the vehicle, a RS-LiDAR-16 was installed in the front bumper. This lidar configuration creates a good point-cloud coverage around the vehicle for perception purposes.

Fig. 3. iseAuto simulated model with different lidars installed.

3. VIRTUAL ENVIRONMENT CREATION

The fierce competition in the gaming industry nowadays has generated many features for game engines. These engines can simulate physics and thus be exploited as simulators aside from game development. SVL and others have already taken advantage of the aforesaid engines and created a framework for testing autonomous vehicles within such physics simulators. Even though these simulators provide some basic tools and assets to get started, it is still not sufficient. To make it more realistic, we need to have real-world terrains simulated.

3.1. Workflow

In order to create a terrain for simulation, the area to be simulated has to be mapped. There are certain steps to follow:

- Data Collection and Processing;
- Terrain Generation.

Data is collected by aerial photography and processed further to obtain a dense point-cloud of the area to be mapped. The point-cloud is then processed through a process called segmentation. Lastly, it is fed into Unity as an input for terrain generation.

3.2. Data collection and processing

Aerial imagery of the area to be mapped has to be captured with a camera drone. The images are captured at a grid flight path, which ensures that the captured images cover different sides of a subject. In order to make sure that the images have maximum coverage, the flight path is followed three times from different camera angles but at a constant altitude. Taking aerial photos 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, which may disturb the photogrammetric process. The images taken are georeferenced by the drone and if necessary, a stationary Real Time Kinematic (RTK) device can be utilized to mitigate errors and shift the positioning data stamped on the pictures. The onboard IMU provides the pictures with 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 performed 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.

3.3. 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 performed 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 colour map to utilize in conjunction with the Unity's terrain engine to create realistic environments.

4. SIMULATION AND SAFETY ASSESSMENTS

Based on the simulation architecture illustrated in Fig. 2, the AV can be run inside the virtual environment. In 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

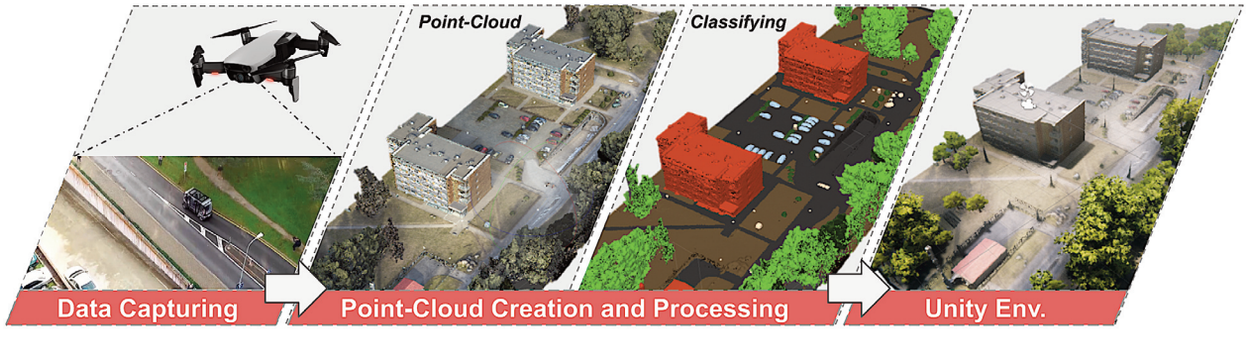

Fig. 4. Steps for virtual environment generation.

[20]. Now, by using a high-fidelity simulator we can simulate different scenarios close to real life in order to evaluate the control algorithm performance and safety. In terms of defining these scenarios, SVL provides a Python API for spawning different objects such as 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 SVL environment while picture (b) illustrates 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 processing. 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 are 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. In this way, simulations can help first to improve our perception and detection system, and then to improve 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 common point-cloud filtering is ground removal, in which some part of the point-cloud defined as ground will be separated. Each lidar point-cloud can be filtered separately or once after 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 Fig. 6. Filtering was applied to Fig. 5b. As a result, the ground, which can be seen in the figure, is almost removed from the point-cloud (see Fig. 6a). However, the NPC points remained and they were clustered as an object in Fig. 6b. Filtering accuracy results in high-performance object detection and safe decision making [21]. Figure 7 illustrates 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. Figure 7b shows that the NPC is

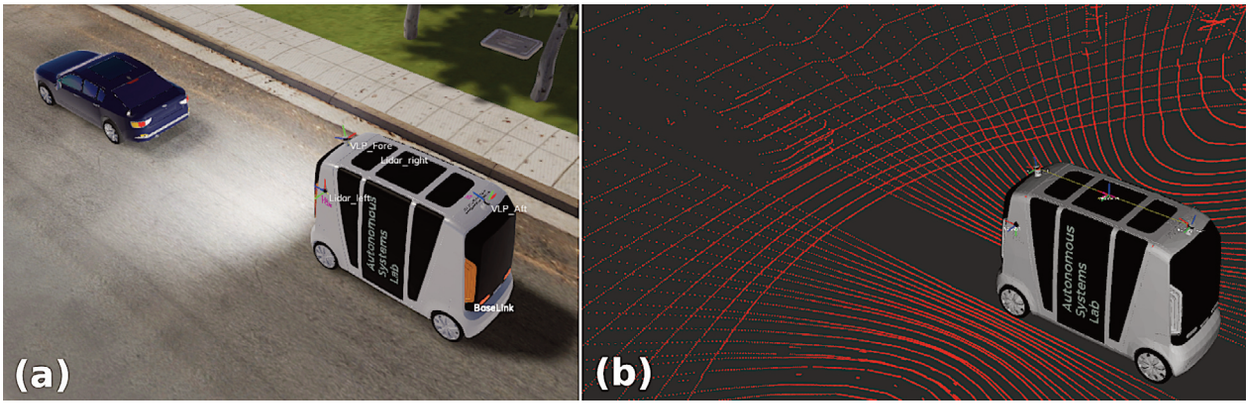

Fig. 5. (a) SVL environment versus (b) Rviz point-cloud visualization.

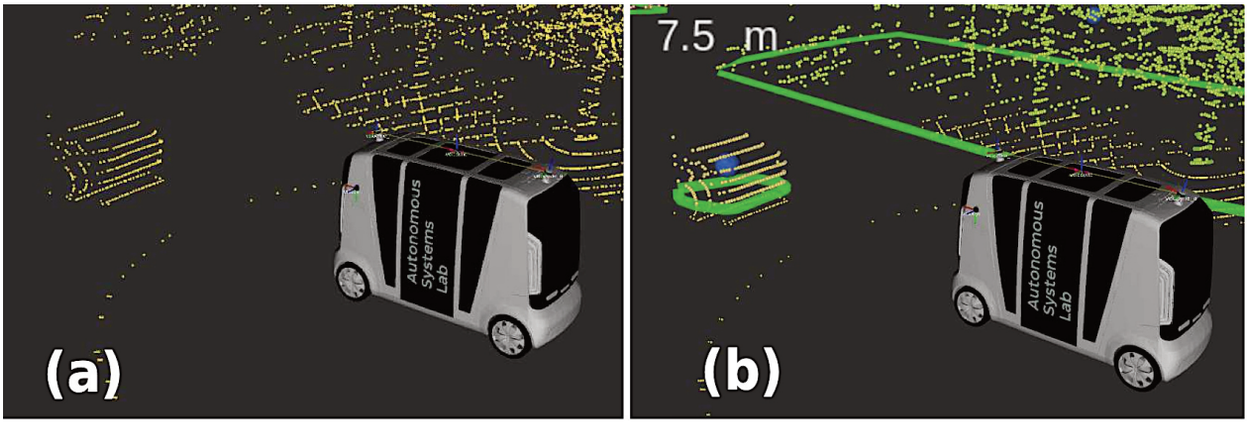

Fig. 6. (a) The ground filtering of the point-cloud and (b) applying of Euclidean clustering.

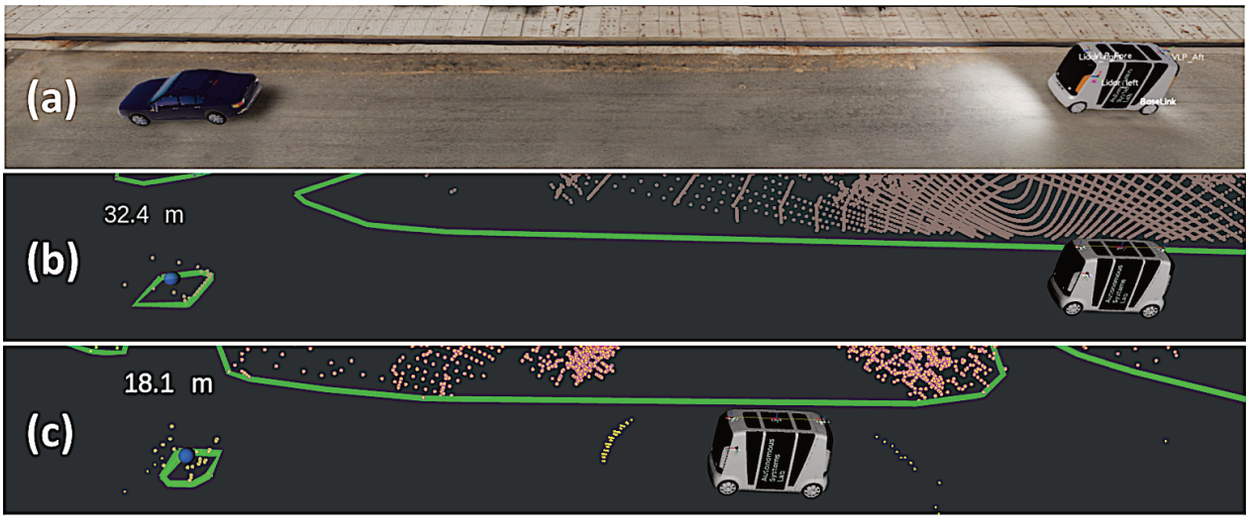

Fig. 7. Maximum distance for detecting a stopped NPC after filtration with different filtering parameters.

detected by the AV shuttle from the distance of 32 metres but picture (c) demonstrates that the maximum distance enabling to detect an object has decreased to 18 metres. 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-cloud concatenation.

4.1. Scenario definition

Scenarios are plans for studying simulations effectively. A good scenario generator can help to validate the whole control system faster in a more reliable way, guaranteeing to cover all the corner cases that might cause failure in the system. There are several methods for generating the scenarios such as human designed, grid search and optimized searching. For example, in [22], the authors implemented a learning method to find safety-critical scenarios for specific tasks. In this paper, for showing the simulation workflow, two main and simple overtaking scenarios were studied. Figure 8 demonstrates two different situations in overtaking: scenario A shows a stopped car that is overtaken by our shuttle while scenario B shows the same mission with an additional car, already starting to overtake the two others.

4.2. Running simulation

In this section, the two described scenarios are simulated inside the simulator and shuttle behaviour is monitored.

- Scenario A

In this scenario, the shuttle is passing a stopped vehicle by generating an alternative local waypoint. The overtaking algorithm is enabled after the shuttle has detected an obstacle in its path. Five different frames of this scenario simulation are shown in Fig. 9. First, the AV follows the way and detects the obstacle (step 1), then stops 15 metres before the object (step 2) and generates a new waypoint (step 3). Then, it starts to follow the new waypoint, and finally, after passing the obstacle, it changes the lane back to the initial path (step 4) and continues its former route (step 5).

By simulating scenario A several times in different areas, the overtaking algorithm for passing a static object was initially evaluated and verified. But to investigate more challenging situations, various road users such as other vehicles and pedestrians should be involved in the scenario. For this, another scenario was designed by adding another vehicle driving forward from behind in the opposite lane.

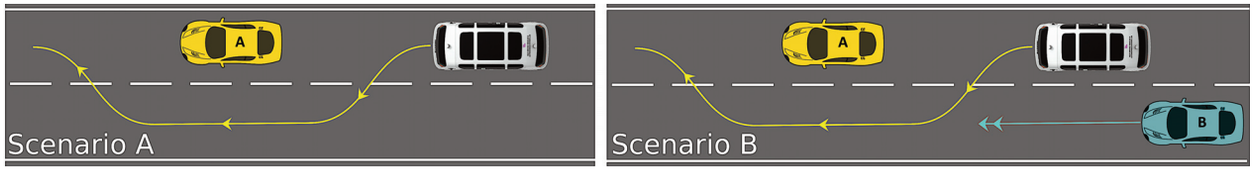

Fig. 8. Two different scenarios for overtaking.

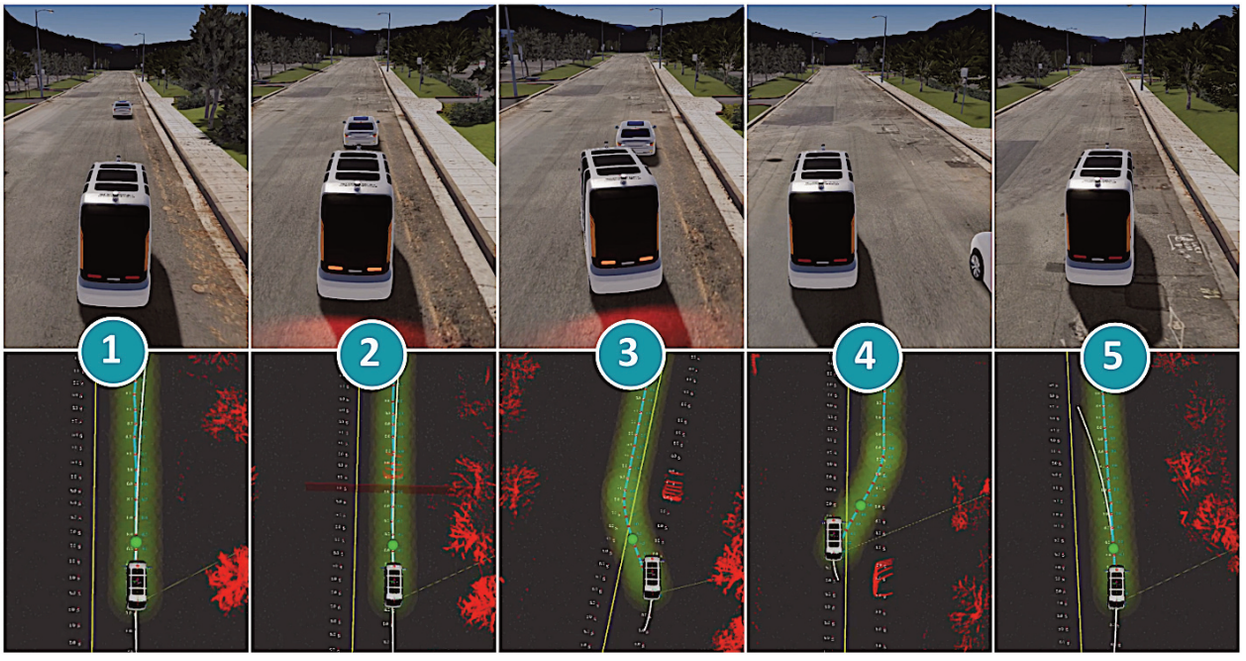

Fig. 9. Five different steps of the scenario A simulation in the SVL simulator (top) and in the Rviz (bottom).

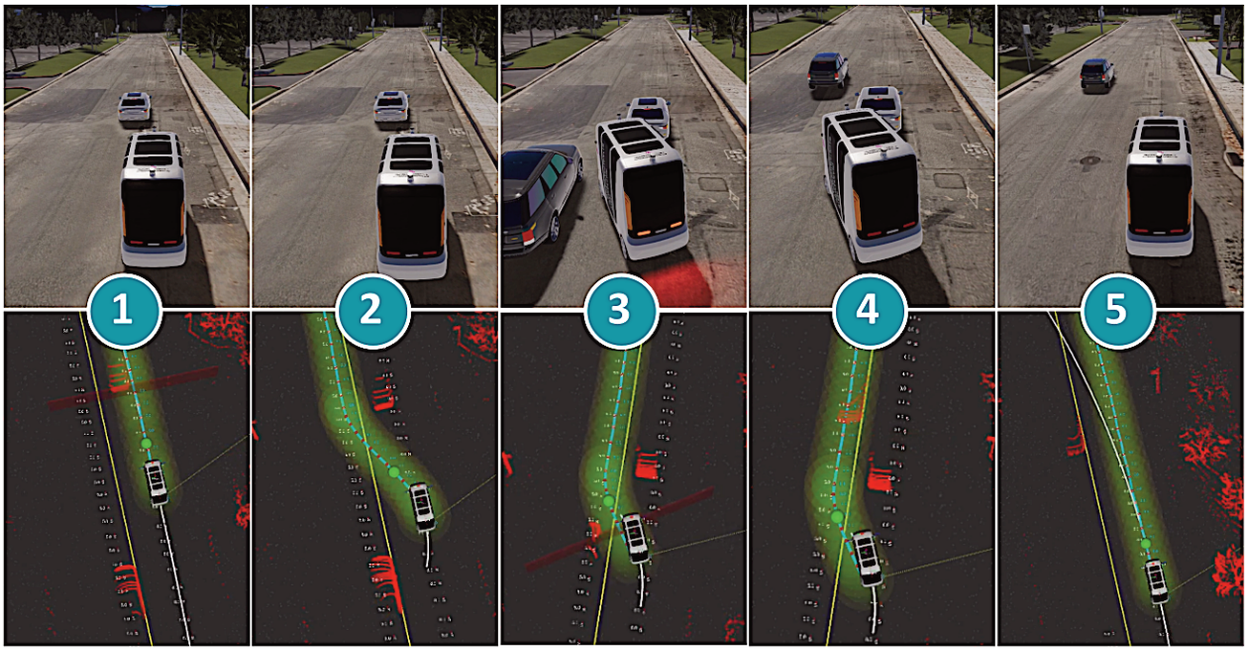

Fig. 10. Different steps of an overtaking process.

- Scenario B

Figure 8 shows the scenario B scheme, that a third vehicle is overtaking the shuttle and the stopped vehicle. It is expected that the shuttle prevents collision and considers the opposite lane traffic. Similar to the former scenario, five steps of scenario B are recorded in Fig. 10. As seen in the simulation, the AV reaches the static object and stops to prepare for overtaking (step 1). The moving vehicle is visible in the Rviz software (frame 1 image below) as a red point-cloud cluster. It is expected that the shuttle prevents collision and considers the opposite lane traffic while overtaking. In step 2 the shuttle starts to overtake and the new path is generated. Before the shuttle changes the lane, it meets the moving vehicle in the green

area (collision area, any object inside it is an obstacle), then the shuttle stops before the collision happens. Finally, after the moving vehicle drives more than 15 metres along the green area, the shuttle starts to follow the route and changes the lane back to its initial path.

This scenario was simulated with a different value for variables such as the speed of the moving vehicle and the lateral position of each vehicle on the road. The results recorded collision in some cases and investigations showed that due to the limited size of the green area and lack of an efficient motion prediction while shifting lanes, the AV can collide with other road users that are not considered. Therefore, using the current overtaking algorithm without any added prediction feature is rejected and it is not safe to be implemented in the real shuttle.

5. CONCLUSIONS

Safety validation is crucial for most of the AV developments and deployments. The simulation as a validation approach presented in this paper offers a practical and effective way to evaluate the safety in different levels. This paper provides the simulation architecture of iseAuto with SIL testing, which shows how the virtual environment and vehicle model are used in combination with Autoware to simulate different scenarios. As an illustration, two overtaking scenarios were studied and the control algorithm was examined based on its safe performance. In conclusion, the development and utilization of this testing scheme will enable the development of safety improvement and autonomous vehicle performance.

ACKNOWLEDGEMENTS

This research received funding from two grants: the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No. 2014-2020.4.01.20-0289. The publication costs of this article were covered by the Estonian Academy of Sciences and Tallinn University of Technology.

REFERENCES

1. Kalra, N. and Paddock, S. M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract.*, 2016, **94**, 182–193.
2. Chao, Q., Jin, X., Huang, H. W., Foong, S., Yu, L. F. and Yeung, S. K. Force-based heterogeneous traffic simulation for autonomous vehicle testing. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 20–24, 2019*. IEEE, 8298–8304.
3. Aeberhard, M., Rauch, S., Bahram, M., Tanzmeister, G., Thomas, J., Pilat, Y., Homm, F., Huber, W. and Kaempchen, N. Experience, results and lessons learned from automated driving on Germany's highways. *IEEE Intell. Transp. Syst. Mag.*, 2015, **7**(1), 42–57.
4. Anderson, S. J., Peters, S. C., Pilutti, T. E. and Iagnemma, K. Design and development of an optimal-control-based framework for trajectory planning threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In *Robotics Research* (Pradalier, C., Siegwart, R. and Hirzinger, G., eds). Springer, Berlin, Heidelberg, 2011, 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. Unsettled technology areas in autonomous vehicle test and validation. SAE Technical Paper Series, 2019.
6. Malayjerdi, M., Kuts, V., Sell, R., Otto, T. and Baykara, B. C. Virtual simulations environment development for autonomous vehicles interaction. In *Proceedings of the ASME 2020 International Mechanical Engineering Congress and Exposition, Portland, OR, USA, November 16–19, 2020.* ASME, **2B**, IMECE2020-23362.
7. Kuts, V., Modoni, G. E., Otto, T., Sacco, M., Tähemaa, T., Bondarenko, Y. and Wang, R. Synchronizing physical factory and its digital twin through an IioT middleware: a case study. *Proc. Est. Acad. Sci.*, 2019, **68**(4), 364–370.
8. Kuts, V., Cherezova, N., Sarkans, M. and Otto, T. Digital Twin: industrial robot kinematic model integration to the virtual reality environment. *J. Mach. Eng.*, 2020, **20**(2), 53–64.
9. Lu, B., He, H., Yu, H., Wang, H., Li, G., Shi, M. and Cao, D. Hybrid path planning combining potential field with sigmoid curve for autonomous driving. *Sensors*, 2020, **20**(24), 7197.
10. Andersen, H., Schwarting, W., Naser, F., Eng, Y. H., Ang, M. H., Rus, D. and Alonso-Mora, J. Trajectory optimization for autonomous overtaking with visibility maximization. In *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, October 16–19, 2017*. IEEE, 1–8.
11. Easa, S. M. and Diachuk, M. Optimal speed plan for the overtaking of autonomous vehicles on two-lane highways. *Infrastructures*, 2020, **5**(5), 44.
12. Sell, R., Leier, M., Rassõlkin, A. and Ernits, J. Self-driving car ISEAUTO for research and education. In *Proceedings of the 2018 19th International Conference on Research and Education in Mechatronics (REM), Delft, Netherlands, June 7–8, 2018*. IEEE, 111–116.
13. Christophe, F., Sell, R., Bernard, A. and Coatanéa, E. Opas: Ontology processing for assisted synthesis of conceptual design solutions. In *Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering* Conference*, San Diego, USA, August 30–September 2, 2009*. ASME, **49026**, 249–260.

14. Sell, R., Coatanéa, E. and Christophe, F. Important aspects of early design in mechatronic. In *Proceedings of the 6th International Conference of DAAAM Baltic Industrial Engineering, Tallinn, Estonia, April 24–26, 2008*, 177–182.
15. Wang, R., Sell, R., Rassõlkin, A., Otto, T. and Malayjerdi, E. Intelligent functions development on autonomous electric vehicle platform. *J. Mach. Eng.*, 2020, **20**(2), 114–125.
16. Rassõlkin. A., Sell, R. and Leier, M. Development case study of the first estonian self-driving car, iseauto. *Electr. Control Commun. Eng.*, 2018, **14**, 81–88.
17. Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y. and Azumi, T. Autoware on board: Enabling autonomous vehicles with embedded systems. In *Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, April 11–13, 2018*. IEEE, 287–296.
18. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. and Koltun, V. Carla: An open urban driving simulator. 2017, *arXiv:1711.03938*.
19. Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S. et al. LGSVL simulator: A high fidelity simulator for autonomous driving. 2020, *arXiv:2005.03778*.
20. Medrano-Berumen, C., Malayjerdi, M., Akbaş, M. I., Sell, R. and Razdan, R. Development of a validation regime for an autonomous campus shuttle. In *Proceedings of the 2020 SoutheastCon, Raleigh, NC, USA, March 28–29, 2020*. IEEE, 1–8.
21. Habermann, D., Hata, A., Wolf, D. and Osório, F. S. 3d point clouds segmentation for autonomous ground vehicle. In *Proceedings of the 2013 III Brazilian Symposium on Computing Systems Engineering, Rio de Janeiro, Brazil, December 4–8, 2013*. IEEE, 143–148.
22. Ding, W., Chen, B., Xu, M. and Zhao, D. Learning to collide: An adaptive safety-critical scenarios generating method. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, October 25–29, 2020.* IEEE, 2243–2250.

Autonoomse sõiduki turvalisuse hindamise suure täpsusega simulatsiooni meetod

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

Autonoomsete sõidukite tööstus planeerib strateegilisi lahendusi, et kindlustada turvalisus enne, kui autonoomsed sõidukid viiakse masstootmisse. Turvalisuse saavutamiseks on vajalik läbi viia väga erinevaid teste. Kõikide testide tegemine reaalse sõidukiga reaalses linnaruumis on pigem ebapraktiline ja võtaks aega aastaid. Selle probleemi vältimiseks kasutatakse simulatsioone. Antud artikli eesmärgiks on välja pakkuda metoodika ja tehnoloogia turvalisuse valideerimise simulatsioonideks autonoomsete sõidukite testimisel. Artiklis on välja pakutud turvalisuse hindamise meetod, mis on realiseeritud TalTechi linnakus tegutseva TalTechi iseauto autonoomse sõiduki platvormil. On loodud virtuaalne mudel linnaku testalast, mis sisaldab eri objekte ja mis on konverteeritud 3D-kaardiks Unity keskkonnas. Loodud virtuaalne mudel on omakorda sisendiks SVL-simulaatorile, mis ühendab endas virtuaalsete andurite simulatsiooni ning Autoware algoritmid, mis juhivad TalTechi iseautot. Demonstratsioonlahendusena on kirjeldatud simulatsioonijuhtu, kui isejuhtiv sõiduk peab tegema möödasõidu seisvast autost, mis blokeerib sõidurea. Lõpuks on näidatud, kuidas antud lahendus võimaldab hinnata isejuhtiva sõiduki otsuste tegemise võimekust ja turvalisust eri situatsioonides.