

Mixed-reality Automotive Testing with SENSORIS

Balázs Varga^{1*}, Mátyás Szalai¹, Árpád Fehér¹, Szilárd Aradi¹, Tamás Tettamanti¹

¹ Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, H-1111 Budapest, Stoczek street 2, Hungary

* Corresponding author, e-mail: varga.balazs@mail.bme.hu

Received: 03 March 2020, Accepted: 11 March 2020, Published online: 03 August 2020

Abstract

Highly automated and autonomous vehicles become more and more widespread changing the classical way of testing and validation. Traditionally, the automotive industry has pursued testing rather in real-world or in pure virtual simulation environments. As a new possibility, mixed-reality testing has also appeared enabling an efficient combination of real and simulated elements of testing. Furthermore, vehicles from different OEMs will have a common interface to communicate with a test system. The paper presents a mixed-reality test framework for visualizing perception sensor feeds real-time in the Unity 3D game engine. Thereby, the digital twin of the tested vehicle and its environment are realized in the simulation. The communication between the sensors of the tested vehicle and the central computer running the test is realized via the standard SENSORIS interface. The paper outlines the hardware and software requirements towards such a system in detail. To show the viability of the system a vehicle in the loop test has been carried out.

Keywords

test environment, mixed reality, autonomous vehicles, SENSORIS, V2X

1 Introduction

Automation of vehicles and transport infrastructures is continuously increasing (Földes and Csiszár, 2018; Torok et al., 2018). This also means that the testing and validation needs of automotive world are growing intensely. Accordingly, the testing of connected and highly automated vehicles requires novel testing methods (Szalay et al., 2019a). Autonomous driving and related intelligent infrastructure developments open immense possibilities for scientific and technological advances.

Autonomous vehicles can be divided into separate layers in terms of system architecture: perception layer, decision layer, navigation layer, and action layer (Huang et al., 2016). These layers require different testing procedures. There are several test suites for analyzing the different layers. However, more complex scenarios require the combination of these software (i.e. co-simulation) or real-world testing. Real-world testing has several limitations too: it is costly to measure every relevant state of the vehicle, it is hard to reproduce, and the testing of corner cases are often impossible. Corner cases occur rarely in reality and are often dangerous to test. Safety drivers of autonomous vehicles instinctually take over if they feel uncomfortable with what the car is doing. Therefore, it

cannot be fully tested how the vehicle would react in a critical situation (Bolduc, 2019).

Virtual test environments such as Software-in-the-Loop (SiL), Hardware-in-the-Loop (HiL), Vehicle-in-the-Loop (ViL), or Scenario-in-the-Loop (SciL) technologies, are becoming increasingly common (Szalay et al., 2018). In the development of autonomous vehicles, new challenges, such as the use of vehicle models and sensor models, must also be tackled (Eichberger et al., 2017). Co-simulation and mixed-reality testing of AD (Autonomous Driving) and ADAS (Advanced Driver-Assistance Systems) are subjects of undergoing intense studies (Butenuth et al., 2017; Maier et al., 2018; Son et al., 2019). In mixed reality, some parts of the test are real, and some components exist only in simulation. The interconnection of the two realities assumes the existence of some sort of digital twin, i.e. the same object must exist both in virtuality and reality in parallel and in sync. Scenario-In-The-Loop (SCIL) approach implements mixed-reality in an automotive proving ground context (Németh et al., 2019).

The virtualization of tests guarantees reproducibility and repeatability. Since the vehicle moves on a real road surface, it is more realistic than purely simulation-based

testing. Mixed-reality testing implies the existence of a digital twin. Since the real and the virtual VUT (Vehicle Under Test) are intertwined, both react similarly, even so, some objects are either present in reality or in the virtual world. This enables virtualizing the reality or employing virtual objects (e.g. surrounding traffic) in the test cases.

Highly autonomous vehicles will communicate directly with nearby vehicles, the infrastructure as well as other wireless devices primarily through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Using standard communication protocols in automotive testing is essential as it enables interfacing between the products of different OEMs and suppliers.

Automated vehicles are equipped with various sensors continuously measuring vehicle states, position, and the environment (perception). Sharing this data among other vehicles or infrastructure elements is an appealing way to increase connectedness, thus improving self-driving capabilities. The standards SENSORIS and ADASIS (Advanced Driver Assistance Systems Interface Specifications) aim at defining data structures for wirelessly communicating object detection and HD map data respectively (ADASIS, 2020; SENSORIS, 2020). Pillmann et al. (2017) proposed a cloud-based framework for sharing vehicle data. The goal of the database is enabling trading with aggregated data to provide cross-sectoral services. Kutila et al. (2019) tested 5G-based C-V2X (Cellular vehicle-to-everything) communication and analyzed bandwidths and latencies. The paper also discusses communication with cloud-based databases (e.g. SENSORIS, 2020).

In this paper, the virtualization of radar sensor detection through standard communication interface is presented. An automotive radar sensor is mounted on a car. A differential GPS serves as a localization unit for the vehicle. The vehicle and test environment have accurate virtual models (i.e. realizing the digital twins). Detections from reality are injected into the virtual world in real-time.

The paper is organized as follows. In Section 2 the hardware and software architecture of the system and the role of each component are described. Emphasis is laid on the implemented communication interface. Section 3 presents a real-world demonstration of the mixed-reality framework.

2 System description

Section 2 describes the realization of a vehicle in the loop test environment employing only open-source software. The role and interconnection of each component is

described in detail. Emphasis is laid on the communication between the sensors of the vehicle and the test server, as it is done on a standard interface.

The system combines various hardware and software in order to achieve a real-time mixed-reality test environment (Fig. 1). The core of the system is a central control computer that runs a control software that handles the tasks in the virtual reality while interfacing with the real objects participating in the test. The control software establishes continuous communication between the components. The main inputs to the system from the real-world are the positions and other relevant system states (e.g. velocity, brake status, traffic signal states) of the real-world objects. Virtual objects can either solely exist in virtually or in mixed-reality. The trajectory of virtual objects can be prescribed manually or with the help of a microscopic traffic simulator software, e.g. Alvarez Lopez et al. (2018). The traffic simulator can accurately model large scale traffic and describe the behavior of drivers and pedestrians. In addition, the simulator gives access to traffic light states at modeled signalized intersections. The VUT exists in the traffic simulator as a virtual twin too, so simulated vehicles can respond to it accordingly. An additional module of the SCIL is the 3D visualization module. There, every controlled object exists in a predefined world and moves based on the commands given by the SCIL. Therefore, matching the network layout in the traffic simulator and the visualization module is crucial.

This paper focuses on a subset of the SCIL automotive testing framework detailed in Horváth et al. (2019) and Szalay et al. (2019b). The goal of this work is to feed the real objects detected by a sensor mounted on the EGO vehicle to the central module and provide accurate visualization. Furthermore, this goal shall be achieved by employing a standard communication interface and open-source software.

2.1 Hardware architecture

The EGO vehicle is equipped with two sensors participating in the test. A Continental ARS-408-21 Premium Long Range Radar Sensor (77 GHz) is responsible

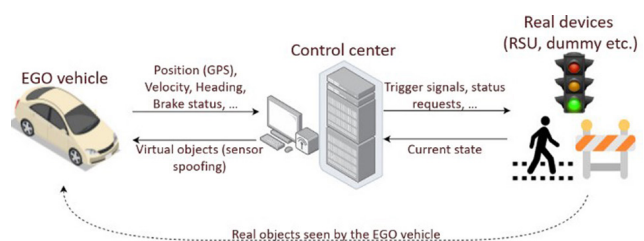


Fig. 1 Main elements of SCIL

for detecting objects in front of the vehicle. A self-developed RTK-GPS provides localization information of the EGO vehicle. This device consists of two u-blox ZED-F9P GPS modules and an LTE GSM module. Due to the rover-base structure, GPS modules provide heading information in addition to velocity and position data. The radar connects to the on-board PC via CAN through a Vector CanCaseXL. The GPS uses a serial port for communication. An on-board PC is responsible for interfacing between the sensors and establishing wireless communication with the central computer. The wireless communication can be realized in multiple ways. It is possible to use wireless V2x devices (e.g. Cohda Wireless MK5 OBU V2X) or 5G New Radio Technology. The hardware architecture is shown in Fig. 2.

2.2 Software architecture

In Subsection 2.2, the software components running on the On-board PC and the central computer are outlined. The custom made software responsible for communication between the components is realized in Python 3.7. The main advantage of Python is that it has a wide variety of open-source libraries allowing rapid software design. The other part of the software outlined in this paper is the visualization module. It is based on the Unity 3D game engine. The advantage of a game engine is its flexibility and realistic visualization. The digital twin of the VUT is defined in Unity too, this digital twin inherits the motion of the VUT from the simulation. For visualization of the detected point of the sensor, a cube on the position of the detection with the given size and transparency is displayed. The degree of transparency based on the probability of the existence of the sensed object. The game engine has 50 Hz update frequency, therefore for smooth visualization interpolation of the measurements is needed.

The On-board software continuously receives and processes the UBX messages and the NMEA sentences with 5 Hz frequency. In parallel, on a separate thread it reads the object detections too from the CAN bus through the CAN interface. The object detection has higher update

rate, 16 Hz, thus synchronization of the two data streams is needed. It is done with the help of two single-slot blocking queues which are always updated if new data is available. The two queues are read one after another, starting with the smaller frequency one. Thus, it can be assumed that the data from the two streams are approximately in sync.

Localization and object detection data are then put into a single SENSORIS data frame. The encoded SENSORIS messages are transmitted to the central computer via a TCP/IP socket.

The Central computer acts as a TCP/IP server, allowing connections from various devices. In this specific case, the only device is the On-board computer inside the EGO vehicle, interfacing with sensors. The visualization module is also connecting through TCP/IP but on localhost. In the central control software, the received SENSORIS messages are decoded, and values are assigned to the virtual representation of the EGO vehicle and the detected objects. The appropriate coordinate transformations are made and a new message is constructed that can be parsed by the visualization module. The interconnection of the software functions is shown in Fig. 3.

Through the interface, the central computer receives the GPS coordinates of the EGO vehicle in decimal degrees, the heading in degrees. The central software maps the GPS coordinates into a scenario-specific Cartesian coordinate system. The radar detections are received in the local Cartesian coordinate system of the sensor. To achieve the global coordinates of the radar detections first, they shall be corrected by offsetting them to the car's front bumper. Then, through a simple coordinate transformation with the position and heading of the EGO vehicle, their global coordinates (which can be used for visualization) are obtained.

2.3 SENSORIS interface

Vehicle sensor data exists in many different formats across automakers, which is an obstacle to rapid development. Efficient creation of autonomous vehicle applications will require a common approach to how vehicle sensor data is gathered by connected cars and sent to the cloud for processing and analysis. SENSORIS is a standard for vehicle-to-cloud data interface. OEMs and various automotive suppliers are participating in the development of the



Fig. 2 Hardware architecture

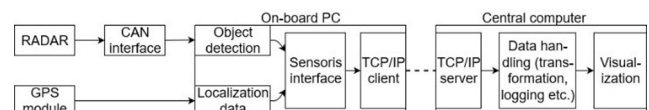


Fig. 3 Software architecture

framework. The interface of SENSORIS defines the content and encoding of the messages that are communicated between the actor roles. Data messages contain vehicle sensor data. Data messages communicated from one vehicle of a vehicle fleet to its vehicle cloud contain sensor data from one vehicle. Data messages communicated from a vehicle cloud to a service cloud contain data from individual vehicles or aggregated data from several vehicles of a vehicle fleet (ERTICO – ITS Europe, 2019, Fig. 4).

The interface can be conveniently used for other V2X applications too, omitting the cloud. In the proposed SCIL framework, the central computer takes over the role of the vehicle cloud and the service cloud. Messages are encoded with the help of protocol buffers (Varda, 2008), allowing efficient compression and modular data messages. In this work, two event groups are used: localization category and object detection category. The localization category stores the GPS coordinates, heading, and velocity of the EGO vehicle alongside their accuracy. The object detection category encompasses the status, type, bounding box etc. of the target object.

3 Demonstration in the university campus

The demonstration was carried out at the Budapest University of Technology and Economics campus parking lot (Fig. 5).

The test area has an accurate (few cm accuracy) model in Unity 3D based on laser scanning. The locations of the buildings trees etc. are accurate, however mobile objects such as cars at the parking lot differ between the model and the reality. The test vehicle was a Toyota Prius 3 equipped with a long-range radar and a differential GPS, outlined in Subsection 2.1. The vehicle was accurately placed in the simulation with the help of its GPS coordinates and heading, realizing the digital twin, see Figs. 6–7.

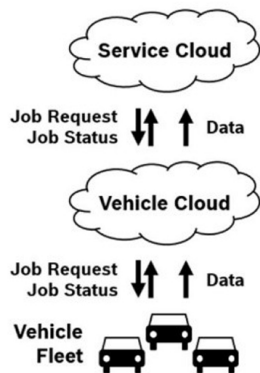


Fig. 4 SENSORIS actor roles (Source: ERTICO – ITS Europe, 2019)

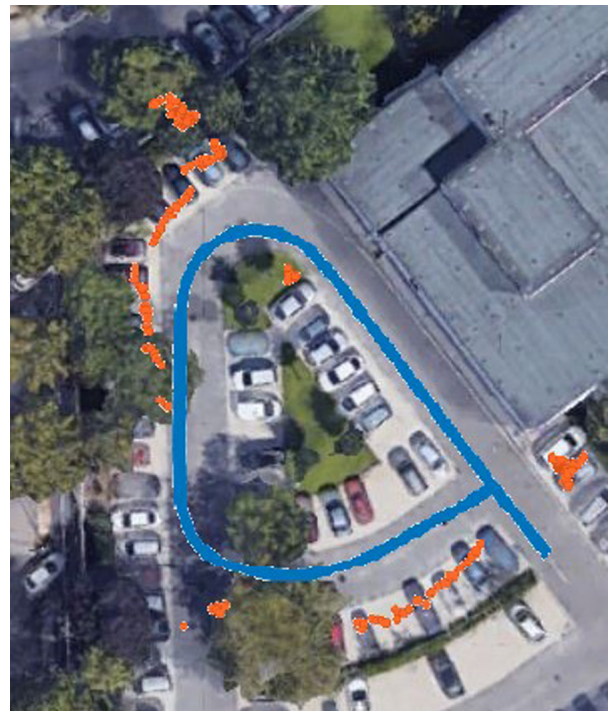


Fig. 5 Campus parking lot at Budapest University of Technology and Economics (Source: Google Maps: 47.478921, 19.056274 – Google, 2020)

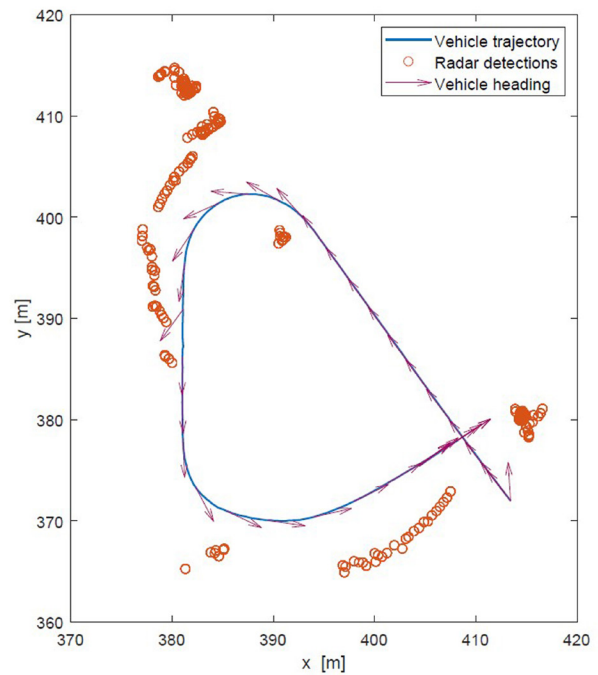


Fig. 6 GPS trajectory and radar detections

The car took one lap in the parking lot and detected the stationary cars. The trajectory of the vehicle and the locations of radar detections are shown on the map in Fig. 5. The GPS and the radar feed were synchronized, thus the detections could be virtualized in real time. Fig.



Fig. 7 Digital twin of the test vehicle and the university campus.
 The detected object is represented by the green cube.

6 depicts the trajectory of the vehicle with the radar detections in the simulator coordinate system. Despite the sparse sampling rate of the GPS, the trajectory is smooth. The radar detections are noisy but the positions of the detected objects can be recognized.

In conclusion, the demonstration substantiated that the radar sensor virtualization using standardized interface can be successfully realized for mixed-reality automotive testing.

References

- ADASIS AISBL "ADASIS", [online] Available at: <https://adasis.org/> [Accessed: 15 February 2020]
- Alvarez Lopez, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y. P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wiessner, E. (2018) "Microscopic Traffic Simulation using SUMO", In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, pp. 2575–2582. <https://doi.org/10.1109/ITSC.2018.8569938>
- Bolduc, D. A. (2019) "NVIDIA exec maps out steps to autonomous vehicles", Automotive News Europe, [online] 27 November 2019. Available at: <https://europe.autonews.com/suppliers/nvidia-exec-maps-out-steps-autonomous-vehicles> [Accessed: 15 February 2020]
- Butenuth, M., Kallweit, R., Prescher, P. (2017) "Vehicle-in-the-Loop Real-world Vehicle Tests Combined with Virtual Scenarios", ATZ worldwide, 119(9), pp. 52–55. <https://doi.org/10.1007/s38311-017-0082-4>
- Eichberger, A., Markovic, G., Magosi, Z., Rogic, B., Lex, C., Samiee, S. (2017) "A Car2X sensor model for virtual development of automated driving", International Journal of Advanced Robotic Systems, 14(5), pp. 1–11. <https://doi.org/10.1177/1729881417725625>

4 Conclusions and future work

In this paper, the virtualization of the perception layer sensor signals was presented. The approach is based on the digital twin of the test vehicle and the test network. The system can be constructed from a differential GPS and a perception sensor and two wirelessly connected PCs. From the software side, the system was realized with open source software (i.e. Python and Unity 3D). The communication between the two computers were carried out using the SENSORIS communication standard (ERTICO – ITS Europe, 2019). The main challenge was to synchronize the two sensor feeds (GPS and perception). Simulation results suggest that the digital representation of the vehicle moves accurately in the virtual world in sync with reality. The detected objects can be adequately positioned in the simulation too.

In this demonstration, the system worked in an open-loop fashion, there was no data sent from the central computer to the test vehicle. The next step is closing the loop with the help of sensor spoofing and virtual sensor models. The virtual perception sensors scan the virtual world and transmit the detections to the car, making the car think those objects are real.

Acknowledgments

The research reported in this paper was supported by the Higher Education Excellence Program in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/FM).

- ERTICO – ITS Europe (2019) "SENSORIS Interface Architecture. Version 1.0.0" [online] Available at: <https://sensor-is.org/presentations/> [Accessed: 15 February 2020]
- Földes, D., Csiszár, Cs. (2018) "Framework for planning the mobility service based on autonomous vehicles", In: 2018 Smart City Symposium Prague (SCSP), Prague, Czech Republic, pp. 1–6. <https://doi.org/10.1109/SCSP.2018.8402651>
- Google "Google Maps" [online] Available at: <https://www.google.com/maps/@47.4786585,19.0564618,19.25z> [Accessed: 15 February 2020]
- Horváth, M. T., Tettamanti, T., Varga, B., Szalay, Zs. (2019) "The Scenario-in-the-Loop (SciL) automotive simulation concept and its realization principles for traffic control", In: Proceedings of the 8th Symposium of the European Association for Research in Transportation (hEART 2019), Budapest, Hungary, pp. 1–6.
- Huang, W., Wang, K., Lv, Y., Zhu, F. (2016) "Autonomous vehicles testing methods review", In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, pp. 163–168. <https://doi.org/10.1109/ITSC.2016.7795548>

- Kutilla, M., Pyykönen, P., Huang, Q., Deng, W., Lei, W., Pollakis, E. (2019) "C-V2X Supported Automated Driving", In: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, pp. 1–5.
<https://doi.org/10.1109/ICCW.2019.8756871>
- Maier, F. M., Makkapati, V. P., Horn, M. (2018) "Environment perception simulation for radar stimulation in automated driving function testing", *e & i Elektrotechnik und Informationstechnik*, 135(4), pp. 309–315.
<https://doi.org/10.1007/s00502-018-0624-5>
- Németh, H., Hány, A., Szalay, Z., Tihanyi, V., Tóth, B. (2019) "Proving Ground Test Scenarios in Mixed Virtual and Real Environment for Highly Automated Driving", In: Proff, H. (ed.) *Mobilität in Zeiten der Veränderung*, Springer Gabler, Wiesbaden, Germany, pp. 199–210.
https://doi.org/10.1007/978-3-658-26107-8_15
- Pillmann, J., Wietfeld, C., Zarcu, A., Raugust, T., Alonso, D. C. (2017) "Novel common vehicle information model (CVIM) for future automotive vehicle big data marketplaces", In: 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, pp. 1910–1915.
- SENORIS "SENSORIS", [online] Available at: <https://sensor-is.org/> [Accessed: 15 February 2020]
- Son, T. D., Bhawe, A., Van der Auweraer, H. (2019) "Simulation-Based Testing Framework for Autonomous Driving Development", In: 2019 IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, pp. 576–583.
<https://doi.org/10.1109/ICMECH.2019.8722847>
- Szalay, Zs., Hamar, Z., Simon, P. (2018) "A Multi-layer Autonomous Vehicle and Simulation Validation Ecosystem Axis: ZalaZONE", In: International Conference on Intelligent Autonomous Systems 15 (IAS 2018), Baden-Baden, Germany, pp. 954–963.
https://doi.org/10.1007/978-3-030-01370-7_74
- Szalay, Zs., Hamar, Z., Nyerges, Á. (2019a) "Novel design concept for an automotive proving ground supporting multilevel CAV development", *International Journal of Vehicle Design*, 80(1), pp. 1–22.
<https://doi.org/10.1504/IJVD.2019.105061>
- Szalay, Zs., Szalai, M., Tóth, B., Tettamanti, T., Tihanyi, V. (2019b) "Proof of concept for Scenario-in-the-Loop (SciL) testing for autonomous vehicle technology", In: 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, pp. 1–5.
- Torok, A., Derenda, T., Zanne, M., Zoldy, M. (2018) "Automatization in road transport: a review", *Production Engineering Archives*, 20(20), pp. 3–7.
<https://doi.org/10.30657/pea.2018.20.01>
- Varda, K. (2008) "Protocol Buffers: Google's data interchange format", In Google Open Source Blog [online] Available at: <https://developers.google.com/protocol-buffers/> [Accessed: 15 February 2020]