

## Abstract

he deployment of autonomous vehicles in real-world scenarios requires thorough testing to ensure sufficient safety levels. Driving simulators have proven to be useful testbeds for assisted and autonomous driving functionalities but may fail to capture all the nuances of real-world conditions. In this paper, we present a snapshot of the design and evaluation using a Cooperative Adaptive Cruise Control application of virtual reality platform currently in development at our institution. The platform is designed so to: allow for incorporating live real-world driving data into the simulation, enabling Vehicle-in-the-Loop testing of autonomous driving behaviors and providing us with a useful mean to evaluate the human factor in the autonomous vehicle context.

## Introduction

n the last years, there has been a rapid development of various Advanced Driver Assistance Systems (ADAS), which includes autonomous parking, adaptive cruise control, lane keeping, and others. This progress has increased the autonomy of vehicles which are currently able to drive according to SAE level 3 [1]. At present, autonomous vehicles in level 4 and 5 in the SAE scale are being developed by various carmakers. Such vehicles need to be exercised in real-world environments where they will ultimately be deployed.

In this context, simulation testing is an effective way to test and validate different features of autonomous vehicle technology such as performances, human factors, stability, and environmental adaptability. Simulator testing is also a costeffective solution. Indeed, this would allow for experimentation of different driving condition, by just implementing multiple virtual environments. In addition, these means of testing cancel the safety risks for pedestrians, drivers, vehicles, related to the testing of the autonomous vehicle in real-world traffic conditions. Indeed, in such situation, any failures pose serious risks to both the passengers and surrounding vehicles. Such failures are likely to happen since new algorithms may behave sub-optimally when facing unknown and unforeseen real-world scenarios. Although there are a series of advantages, these simulations cannot always adequately capture all the nuances of real-world driving.

In this paper, we present the virtual reality driving simulator currently in development by our institution. The platform will be designed so to allow for multiple uses. Specifically, it will be designed for the testing of the autonomous driving algorithm and the evaluation of human response to specific autonomous control behavior.

Autonomous vehicle algorithms eventually need to be tested in the real world. However, such testing always poses a risk to the occupants of the vehicle, other drivers, and pedestrians. Our system is defined to integrate live world real traffic data provided by the real circulating vehicle. The use of such data allows for realistic testing of the autonomous algorithm, without incurring any real-world risk.

Many human-machine interaction issues between drivers and autonomous vehicles are currently not well understood. A realistic and interactive simulation testbed is required to empirically and safely examine these issues and to facilitate rapid prototyping of novel interaction metaphors. In order to evaluate and improve such human-machine interaction between the autonomous vehicle and either the passenger or other drivers of different non-autonomous vehicles circulating in the same road, we designed a platform which provides live information of the simulated autonomous vehicle to the realvehicle driver.

The paper is organized as follows. In section 2, we review the state of the art regarding existing driving simulators, highlighting their limitations which our simulator aims to solve. Section 3, describes in detail the software architecture and the technical implementation of the system. In section 4 we introduce the method for the empirical evaluation of the system time delay. In section 5 we present the results of the platform when used for a Cooperative Adaptive Cruise Control (CACC) application. Finally, in section 6, we will draw our conclusions on the current state of the simulator and discuss further developments and potentials for future research.

# **Research Background**

The use of a driving simulator to test autonomous functionalities of a vehicle is not a new approach. With more than a decade developing, there have been many commercial/opensource tools designed to build unmanned system models and simulation environments.

### **Related Works**

Driving simulators have to be designed and evaluated with respect to the user experience. In recent years there have been many works whose results allow to gain a better understanding of the properties a good and accurate simulator must have. For example, in [2] they used an immersive VR based driving simulator to study driver behavior, techniques, and adaptability. Other works [3] study and compare the error between on-road real driving and driving in a simulated environment. Such works although very helpful for testing the user driving experience are not meant to include the simulation of autonomous vehicles (AV).

In the field of autonomous vehicle simulation and more generally in mobile robots, examples of simulators include USARSim [4], Gazebo [5] and Microsoft Robotics Studio [6], which provide a virtual environment for the testing of autonomous driving algorithms. More recently, simulators have been developed to increase the level of realism of the virtual environment. For example, in [7] they propose an environment modeling approach using image sequences acquired with both GPS and camera mounted on the car and road GIS data for Autonomous Vehicle simulators. They contribute to the improvements of simulator environments which are modeled based on the real-world data, with the result of being more realistic and therefore more effective than the currently existing ones based on Gazebo and USARSim.

A good simulator environment must be able to: replicate complex urban environment, simulate realistic host vehicle, model traffic participants. In [8, 9, 10] they discussed the design and implementation of their simulators accounting respectively for these three fundamental aspects. Specifically, [8] discusses a distributed simulation platform for autonomous driving cloud system, [9] considers the implementation of intelligent actors in simulation and [10] examines ways to control the weather, sensing and traffic control.

In the last years, there has been a rapid development of various ADAS systems. This progress has increased the autonomy of vehicles which are now able to drive according to SAE levels 2 and 3. In this stage, the vehicle autonomously drives for most of the time. However, the passenger must always be on alert during driving, since the control can pass to him in case of an unexpected situation. This would affect both the passengers and the vehicles around it since the reaction of the driver is unpredictable. In this context, the testing of autonomous driving algorithms must include feedback from both the passenger and other vehicles involved in the driving scenario. A possible way to account for such unexpected behavior and the effects on the person in the simulation is to include the software in the hardware simulation loop. Gechter et al. [11] discuss a hybrid autonomous vehicle simulator that is closest to our work. They use an RTK GPS device to record data for actors and try to introduce software simulation in the hardware loop by using a virtual sensor which senses the objects in the virtual environment as if perceived in the real vehicle. Their approach represents a good step in the direction of this new kind of simulators. However, differently from the proposed simulator, it does not allow to reproduce actual vehicles driving in real traffic situation while performing an immersive VR simulation.

## Characteristics of Vehicle Simulators

The existing simulators differ in the provided features depending on the field and interest of study. Below we highlight what features are the most important when it comes to the testing of autonomous vehicles algorithm.

According to Mudd (1968) [12] and McCormick (1970) [13], pioneers in the validation of driving simulators, the first characteristic that a simulator must achieve is the realism of the simulation. In the context of autonomous vehicles, a high level of realism requires the development of multiple features. First of all, the behavior of the autonomous vehicle is affected by the environment. This creates the need for simulator environments which are modeled based on the real-world data as the one presented in [7]. Second, in order to achieve useful results for real-world uses cases, it is important to have an accurate physical model of the simulated vehicle which not only accounts for its kinematic and dynamics but also for the possible imperfect information coming from sensors. Third including the feedback of the typical passenger in the autonomous vehicles is also essential to validate and improve autonomous vehicle algorithms. In this sense, the introduction of the software in the hardware loop could help to obtain and to improve such feedback from the simulation.

In the research presented in this paper, we tried to assess these three characteristics, and we introduced a novel platform that allows for realistic virtual traffic scenario creation in a virtual reality simulation with real-time traffic data provided by driving the real car on a real-world road. Our simulator aims at creating an immersive virtual reality where a person can experience being a passenger in an autonomous vehicle. In particular, the VR experience is generated by using data coming from a real-world vehicle that runs in a real-world traffic situation during the simulation. For such a reason the level of realism and the timeliness of shared real-time information are crucial aspects. To achieve this objective, we not only defined a realistic representation of the environment using data collected in real-time from actual vehicles, but we also defined a system that allows the user to experience a simulation which integrates the virtual and real worlds.

To the best of our knowledge, there are no other testbeds that allow integrating a VR experience where the virtual reality relies on data collected in real time by actual vehicles driving in real traffic scenarios. Indeed, the originality of this simulator is the real-time connection between the real and the virtual world during simulations. The overall platform consists of the following components: 1) the VR simulation environment to visualize the vehicle 2) a control system which produces the dynamic of the virtual vehicle according to some autonomous driving algorithm 3) a system that allows communication from the real-world car to the visualization module and vice versa.

## The Proposed Simulator

The platform presented in this paper is currently in development at our institution. The main objective of this simulator is to provide a testbed for testing of autonomous driving algorithms and enable research on human-machine interaction and user experience evaluation.

The simulator aims at testing and evaluating autonomous algorithms and human experience of being a typical passenger of an autonomous vehicle. The simulator consists in a virtual autonomous-driven vehicle that from now on we define as virtual-vehicle, and one human-driven real vehicle, defined as real-vehicle. The driver in the real-vehicle can see in real time the behavior of the virtual-vehicle and thus adjust its maneuver depending on it. This interaction between the driver of the real-vehicle and the virtual-vehicle is created using tablet devices mounted on the cars' dashboards and will be referred to as vehicle-in-the-loop. There can be two different cases, either the real-vehicle precedes the virtual-vehicle, or it follows it. In the first case, the tablet will act as a rear-view mirror and will show the virtual-vehicle which follows the real-vehicle according to some autonomous algorithms. In the second instead, the tablet will show the virtual-vehicle from the front view of the realvehicle. The two scenarios are represented in Figure 1.

### FIGURE 1 Vehicle-in-The-Loop concept.



### System Architecture

The VR platform testbed has various system components that are seamlessly integrated via various network communication technologies to operate collectively as a real-time high-fidelity simulation system. The testbed architecture is illustrated in Figure 2.

<u>Figure 2</u> depicts the testbed architecture which is constituted by Visualization Module (VM), Control Module (CM) and data transmission. The real-time data transmission collects the information from the *real-vehicle* and passes them to the other modules. The control module defines the AV behavior of the *virtual-vehicle* in the simulation environment. The VM comprises of the computer-generated environment hosting the *virtual-vehicle* and supports visualization of the CM's response for the *virtual-vehicle* either in VR using a stereoscopic Head Mounted Display (HMD) or on the tablet mounted on the *real-vehicle*. The CM and the VM are located within the same Local Area Network and exchange information with each other over a bi-directional network socket link using User Datagram Protocol (UDP).

### Real-Time Data Communication

As anticipated, the distinctive feature of the simulator is the ability to integrate real-world live data into the simulation and to provide feedback to the real-world from the virtual one.

The *real-vehicle* is equipped with different sensors. In particular, a precise GPS is coupled with an inertial measurement unit (IMU) to provide accurate positioning information about speed, acceleration and heading direction. This information is then streamed into the simulator environment. The real-time transmission of the *real-vehicle*'s data to the simulator relies on an ad-hoc Connected Vehicle Testbed (X-CVT) developed at our institution. The X-CVT is a Dedicated





Short-Range Communication (DSRC) infrastructure setup that allows for Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication capabilities [14]. The DSRC infrastructure of X-CVT comprises communication nodes of two types: Fixed Edge Nodes, and a Mobile Edges Node. The first ones are installed along a 1-mile stretch of a road, the second one is placed inside the *real-vehicle* [14]. The Fixed Edge Nodes are linked to the university campus's Local Area Network via Optical Fiber and Wi-Fi backhaul links. Mobile Edge Node consists of an On-board Unit (OBU) that facilitates communication with Fixed Edge Nodes using DSRC technology. The OBU also supply the required GPS location and estimated speed data for the *real-vehicle* as it is driven on the road.

Data collected from the *real-vehicle* is published to a Communication System Node running Messaging Queue Telemetry Transport (MQTT) broker. MQTT is a lightweight publish-subscribe based messaging protocol. The VM subscribes to the MQTT broker to receive the *real-vehicle*'s published information. The overall system allows for real-time data communication from the vehicle to the simulation platform with reduced latency.

### Visualization and Control Module

The Visualization Module uses packets received from the MQTT broker to recreate and render the vehicle motion of the *real-vehicle* in a predefined simulation environment. The core of the VM consists of a virtual reality simulation environment built using the popular game engine, Unity3D. This consists of a real-world track established on a public road and reconstructed in the virtual world. The real-world test track is shown in <u>Figure 3</u>. The track was selected so to represent a common driving scenario in an urban environment. The track consists of a mile-long stretch of a 2-lane road along which multiple intersections and traffic lights.

**FIGURE 3** Top view of the modeled track



**FIGURE 4** Sample view of the simulation environment



© 2019 SAE International. All Rights Reserved

The reconstruction of the environment includes 3D models of trees, buildings, traffic lights so to be as close as possible the real-world scenario. The virtual track was constructed starting from the GPS coordinates of the road center, extracted from Google Maps and later verified by taking multiple measurements along the track with a high precision GPS. A transformation was established between real-world GPS coordinates and the X-Y-Z coordinates of the Unity virtual environment. This is a necessary step in order to map the location of the real-world vehicles into the simulation. A sample view from the developed 3D scene is shown in Figure 4.

The Visualization Module receives the information about the *real-vehicle* state i.e. position, acceleration, and velocity and communicates them to the Control Module. The CM is developed through co-simulating CarSim, and MATLAB/ Simulink. It uses the information from the VM and defines the new *virtual-vehicle*'s state according to specific autonomous algorithms. The state of the *virtual-vehicle* is communicated to the VM which renders the vehicle into the simulation environment according to the new states defined by the CM.

The environment, as well as the rendered vehicles, can be visualized either through an HMD or through the tablet mounted on the vehicle. The VM runs on a high-end graphics machine powered by NVIDIA GTX 1080, enabling it to render an immersive stereoscopic view of the interior of the car and surrounding environment at a stable frame rate of 90 FPS.

## **Baseline Empirical Evaluation**

Our test bed is a virtual reality-based simulation whose main purpose is to facilitate human factors evaluation of autonomous vehicle behavior algorithms. The main innovative contribution of this simulator consists in the use of real-time world data from the *real-vehicle*, to update the position of the *virtual-vehicle* in the virtual environment according to some autonomous driving algorithms and provide a feedback to the driver of the *real-vehicle*. Indeed, the new position of the *virtual-vehicle* is showed on the tablet mounted inside the *real-vehicle* to create a virtual-real world feedback as described in Figure 1.

The overall output of the platform strongly depends on the intrinsic delay between the different elements of the network. Indeed, in case of high delay, the virtual-vehicle will be out of time displayed with respect to the dynamic of the real-vehicle, and this would negatively affect the realism of the whole simulation. It is crucial to guarantee a limited amount of time to transfer information from the car to the Visualization and Control Modules and vice versa. In order to evaluate the platform's time performances, we decided to assess the different elements of the platform separately. In particular, we decided to test the platform for the following performance measures: backend latency and network latency. The first one represents the average time it takes for a packet of information to travel between the Visualization and the Control module. The second instead is the average time in seconds it takes for a GPS packet to reach the VM.

## **Network Latency**

The objective of the latency tests was to measure the network latency during data transmission observed between the Visualization Module and the *real-vehicle*. The *real-vehicle* utilizes an experimental testbed consisting of three edge nodes spanning a road length of approximately 1 mile [14]. The edge nodes consist of DSRC capable Cohda Road Side Units (RSUs) and Intel NUC computing devices with Linux OS. The *real-vehicle* is equipped with DSRC capable On-Board Units (OBUs) that can communicate with the edge nodes. A central server connects the edge nodes to the machine housing the VM. While the communication between the OBU and RSU is conducted wirelessly via DSRC, the RSU and the central server are backhauled via an optical link over the network of our institution.

The *real-vehicle* is equipped with an OBU that allows it to communicate vehicle data with the VM. The vehicle data is a JSON string consisting of relevant vehicular information such as positional information (latitude, longitude), mobility information (speed, acceleration), sent message counter, vehicular ID and the timestamp at the time of transmission. For the purpose of these tests, the OBU's MAC address was considered as the vehicle's unique ID. This data is broadcasted to the RSUs where a gateway application dynamically converts the received data into a message publishable to the central server using the MQTT protocol. A Mosquitto broker running on the server receives this data and forwards it to the VM that is subscribed to the same broker. To communicate via the broker, care is taken that the RSUs and VM use the same topic to publish and subscribe to the broker respectively. Unix timestamps of each packet's transmission from the OBU and the RSU, and the final receiving timestamp at the VM are recorded to allow insight into the end-to-end latency and the latency observed over the wired and wireless link of the physical network. End-to-end latency for each packet is measured as the time difference between the timestamps recorded at the OBU and the VM. Wireless link latency is measured as the time difference between the timestamps recorded at the OBU and the RSU, whereas the wired link latency is measured as the time difference between timestamps recorded at the RSU and the VM which incorporates the latencies associated with message publish protocol. Typically, each test involved broadcasting approximately 1000 packets from the OBU and recording the average latency for each test and the average of individual latency observed for each correctly received packet.

Table 1 summarizes the results obtained from tests conducted using the experimental setup described earlier. Various testing configurations were used for testing where the varying factors were vehicle motion and packet rate. The stationary test utilizes the real-vehicle communicating with the VM using 1 RSU. Tests with the real-vehicle in motion broadcasting packets utilized all 3 RSUs. The broadcast of the real-vehicle data allowed multiple RSU's in the DSRC communication range of the vehicle to receive and forward the messages. This technique allowed efficient transfer of communication from one RSU to another in the direction of the vehicle's mobility. However, it proposed duplicate messages being received. Therefore, at the VM care was taken to disallow duplicate messages identifying each unique message based on timestamp and message counter. The packet size of each JSON string being sent from the OBU was 224 bytes. The packet rate requirement for the VM was 40 packets/second, and this rate was used to send the packets from the OBU. Additionally, testing was done with a packet rate of 80 packets/second to demonstrate the network's capability to support the higher packet rate as well. The latency tests observed were satisfactory for the Virtual Reality Simulation tested. These results were also consistent when tests were repeated at a later date. Tests 1 was conducted on June 1, 2018, and tests 2 and 3 were conducted on May 19, 2018.

<u>Figure 5</u> shows the graphical representation for latency observed for each packet transmitted from the OBU in test 1.

#### **TABLE 1** Latency Test Results.

Test	Rate (s/sec)	Size (bytes)	Stationary/ Moving	Average Latency (sec)
1	40	300	Moving	0.008303
2	40	300	Moving	0.006746
4	80	300	Stationary	0.006925
			2010 CAE Internetions	

2019 SAE International. All Rights Reserved.

#### FIGURE 5 Overall latency



As we can see from the picture, the average value is consistent with the shown in the Table. The overall latency is the sum of the latency between the OBU and the Fixed Node and the one between the Fixed Node and the VM.

<u>Figure 6</u> shows the comparison of the latency observed over the wired and wireless link for the same test. The red one represents the delay between the OBU and the RSU. While the blue represents the latency between the RSU and the VM. As we can see the overall latency is equally distributed in the two delays. From the calculated latency, it was possible to extract a probability distribution density, which is reported in <u>Figure 7</u>. The probability distribution density has been computed by employing the MATLAB function *ksdensity*.

### **Backend Latency**

Backend latency test measures the latency between the Visualization and the Control Module. The two modules











communicate with each other through JSON string which contains information about the state of the simulated car such as latitude, longitude, and velocity. In order to calculate the overall latency, we add the timestamp value to the string. In particular, when the two modules communicate, both the delivery and the receiving timestamp are included into the string. In this way, both the latency from VM to CM and from CM to VM can be calculated as the difference between the delivering and receiving timestamp. The overall latency between the two modules is calculated as half of the sum of the two delays (from VM to CM and from CM to VM) of the two modules. This calculation is repeated for the whole simulation and from it, we extracted the probability density function shown in Figure 8. The probability distribution density has been computed by employing the MATLAB function ksdensity.

From the picture, we can see that the average latency between the Control and Visualization Module resulted to be around 5ms which represent a satisfactory result.

# CACC Application Study

To test and evaluate the performances of the visualization and control modules from a user point of view we decided to set up a Co-operative Adaptive Cruise Control (CACC) application. The CACC allows a platoon of cars to move one after another in a smooth flow and such that each car maintains a desired distance from the preceding one, knowing the position and dynamic of both preceding and following car. In such an application, each car communicates its position to the other cars through a wireless connection.

We present the simulator set up which uses the HDM as represented in Figure 9 to ten different participants. The group consisted of 7 males and 3 females in the age range of 18 to 25. We ask them to fill out different questionnaires to evaluate multiple aspects of the virtual simulators: simulator sickness, the presence of virtual reality experience, quality of the experience, the human factor of the CACC response.

#### AN IMMERSIVE VEHICLE-IN-THE-LOOP VR PLATFORM FOR EVALUATING HUMAN-TO-AUTONOMOUS



### **User Experience Measures**

The main objective of such simulation is to test the CACC controller performances with respect to the user experience. To do so, we defined a platoon of two different cars. The trajectory of the leading car is generated from the real-vehicle. The test consists in driving the real-vehicle along the real track, collects its GPS, velocity, acceleration and orientation signal and reproduce (off-line) the path of the real-vehicle in the virtual environment. The trajectory of the second car has been defined from the Control Module which receives the information of the leading car from the visualization module and outputs the new position of the second car, the *virtual-vehicle*, according to the CACC controller. The motion and location of real-vehicle and virtual-vehicle in the virtual environment are shown to the driver only through the HMD and not through the tablet. In particular, during the simulation, the user experience being in a car as if it sits in the virtual-vehicle in Figure 9.

In order to assess the simulator sickness effects, we present to the user a pre- and post-Simulator Sickness Questionnaire (SSQ) before and after experiencing the virtual simulator. The results were analyzed according to the standard introduced in [<u>15</u>]. Specifically, the virtual simulator reports a mean total SSQ score in the postcondition equal to 14.2 as shown in Figure 10 with orange stripes. Such a result is very promising according to the scale provided by [<u>15</u>].





Furthermore, in order to understand the sense of presence as the subjective sense of being in a virtual environment, we used an Igroup Presence Questionnaire (IPQ). The IPQ allows evaluating the sense of presence according to four different criteria: general presence, spatial presence, involvement and experienced realism [16]. The results are reported in Figure 11, and these indicate the high overall presence in the virtual reality simulator.

Lastly, we evaluate the human factor of the CACC controller by asking the users to rate different aspects of the *virtual-vehicle* behavior such as stopping and following distance, acceleration, etc. From such responses, we were able to estimate the six most relevant qualities of an autonomous vehicle virtual reality simulator, which according to [<u>17</u>] include: comfort, enjoyability stress, interest expectation, and frustration. The result of the questionnaire is reported in <u>Figure 12</u> where the two negative criteria, i.e., frustration and stress are reported as inverse because a low absolute score indicates a high quality of experience. The scale for the assessment is from 0 to 6. As reported in <u>Figure 11</u> almost all the criteria reach the maximum value on the scale, indicating a positive user experience.

### FIGURE 11 IPQ questionnaire scores







AN IMMERSIVE VEHICLE-IN-THE-LOOP VR PLATFORM FOR EVALUATING HUMAN-TO-AUTONOMOUS

# Conclusion and Future Work

The work presented discusses the design and implementation of a novel VR platform for the testing of autonomous vehicles algorithm and analysis of human response. The innovative contributions with respect to the currently used driving simulator are two, in sequence: the ability to integrate real-time and real-world data into the simulation and the possibility to provide feedback to the real-world from the virtual vehicle. The system is composed of different elements which are seamlessly integrated via various network communication technologies to operate collectively as a real-time high-fidelity simulation system.

The paper presents the development and the testing of such different elements. The main requirement for these elements is real-time cooperation. Thus, we were interested in testing and proving their operation capability in terms of time delay. The presented results showed that the system overall delay, considered as the sum of the calculated delays of single elements, is of the order of milliseconds. Such a result is very promising and represents a good starting point for the implementation of the overall simulation platform.

Furthermore, the VR platform has been tested with respect to the user experience. In particular, we were interested in understanding if the level of realism provided by the virtual environment created through the visualization and control module interaction were satisfactory. The results obtained offers a first user evaluation of the overall platform. The user experience is a fundamental factor that must be taken into consideration for the implementation of the overall platform. One of the main goals is to investigate the user experience offered by the autonomous vehicle and allows researching human factors aspects associated with it.

## References

- 1. SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor Vehicles," 2016.
- 2. Taheri, S.M., Matsushita, K., and Sasaki, M., "Virtual Reality Driving Simulation for Measuring Driver Behavior and Characteristics," *Journal of Transportation Technologies* 7(02):123, 2017.
- 3. Meuleners, L. and Fraser, M., "A Validation Study of Driving Errors Using a Driving Simulator," *Transportation Research Part F: Traffic Psychology and Behaviour* 29:14-21, 2015.
- Carpin, S. et al., "USARSim: A Robot Simulator for Research and Education," in *Robotics and Automation*, 2007 IEEE International Conference on. IEEE, 2007.
- Koenig, N.P. and Howard, A., "Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator," *IROS*. 4, 2004.

- Jackson, J., "Microsoft Robotics Studio: A Technical Introduction," *IEEE Robotics & Automation Magazine* 14:4, 2007.
- 7. Zhang, C. et al., "Road View: A Traffic Scene Simulator for Autonomous Vehicle Simulation Testing," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on.* IEEE, 2014.
- 8. Tang, J. et al., "Distributed Simulation Platform for Autonomous Driving," in *International Conference on Internet of Vehicles*. Springer, Cham, 2017.
- Gu, T., and Dolan J.M., "A Lightweight Simulator for Autonomous Driving Motion Planning Development," 2015: 94.
- Best, A. et al., "Autono Vi-Sim: Autonomous Vehicle Simulation Platform with Weather, Sensing, and Traffic Control," in *Neural Information Processing Systems*, 2017.
- 11. Gechter, F. et al., "Towards a Hybrid Real/Virtual Simulation of Autonomous Vehicles for Critical Scenarios," in *The Sixth International Conference on Advances in System Simulation* (SIMUL 2014), 2014.
- 12. Mudd, S., "Assessment of the Fidelity of Dynamic Flight Simulators," *Human Factors* 10(4):351-358, 1968.
- 13. McCormick, E.J., *Human Factors Engineering* (McGraw-Hill, 1964).
- 14. Rayamajhi, A. et al., "Thin Gs in a Fog: System Illustration with Connected Vehicles," in *Vehicular Technology Conference (VTC Spring), 2017 IEEE 85th.* IEEE, 2017.
- Kennedy, R.S. et al., "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness," *The international journal of aviation psychology* 3(3): 203-220, 1993.
- 16. Schubert, T., Friedmann F., and Regenbrecht H., *Igroup Presence Questionnaire*, 2001.
- 17. "User Experience Questionnaire Online," [Online]. Available: <u>http://www.ueq-online.org</u>/.

## **Contact Information**

### Dr. Pierluigi Pisu,

Associate Professor at Automotive Department of Clemson University, Greenville SC 29607, USA. <u>pisup@clemson.edu</u>. Website: https://cecas.clemson.edu/pisugroup/index.html

## Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under grant No. CNS-1544910. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

<sup>© 2019</sup> SAE International. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE International.

Positions and opinions advanced in this work are those of the author(s) and not necessarily those of SAE International. Responsibility for the content of the work lies solely with the author(s).