Abstract—According to data from the United Nations, more than 3000 people have died each day in the world due to road traffic collision. Considering recent researches, the human error may be considered as the main responsible for these fatalities. Because of this, researchers seek alternatives to transfer the vehicle control from people to autonomous systems. However, providing this technological innovation for the people may demand complex challenges in the legal, economic and technological areas. Consequently, carmakers and researchers have divided the driving automation in safety and emergency systems that improve the driver perception on the road. This may reduce the human error. Therefore, the main contribution of this study is to propose a driving simulator platform to develop and evaluate safety and emergency systems, in the first design stage. This driving simulator platform has an advantage: a flexible software structure. This allows in the simulation one adaptation for development or evaluation of a system. The proposed driving simulator platform was tested in two applications: cooperative vehicle system development and the influence evaluation of a Driving Assistance System (DAS) on a driver. In the cooperative vehicle system development, the results obtained show that the increment of the time delay in the communication among vehicles (V2V) is determinant for the system performance. On the other hand, in the influence evaluation of a DAS in a driver, it was possible to conclude that the DAS’ model does not have the level of influence necessary in a driver to avoid an accident.

Index Terms—Road safety, driving Simulator platforms, control, cooperative vehicle systems, DAS.

I. INTRODUCTION

During the last 20 years, in the area of Intelligent Transportation Systems (ITS), the main objective has been to find new alternatives that help increment the levels of road safety. Some alternatives to solve this problem are: i) improve the road infrastructure for the drivers and ii) develop safety/emergency systems for the vehicles. However, the high cost of the first alternative is a problem, mainly in developing countries where the rate of death by car crashes is high [1]. Therefore, the safety and emergency systems may be considered the best choice.

In this context, Sitavancov, in [2], describes how the research in safety and emergency systems covers the technology creation and new developments using techniques of several research fields (i.e., computer vision, control systems, and communications V2X among others). These developments may enable faster responses to incidents; better guidance; collision avoidance; better energy use, and improve fleet management.

For more than three decades, the driving simulators have been an alternative to help in the development and evaluation of new safety and emergency systems. It is clear that a driving simulator cannot represent the real world with all circumstances. However, a simulator may be configured with multiple characteristics (i.e., modified car, visual system, motion system, sound, among others) that allow to best abstract the real world.

The main contribution of this study is focused on the development of a driving simulator platform. This platform allows an open and flexible way; develop and evaluate prototypes of new safety and emergency systems; use different scenarios (i.e., where it is possible to interact with other vehicles) and the use of the V2X adaptable communication networks.

Through this driving simulator platform, the users may have the possibility to adapt the structure with its tools (i.e., hardware and/or software) in the development or evaluation of new systems. This paper presents two examples of application of the driving simulator platform, which may contribute to improve the road safety.

The remainder of this paper is organized as follows. Section II reviews the state of art in driving simulation platforms; Section III explains the driving simulator platform architecture proposed. The experimental setup, results, and analysis from a development of a cooperative vehicle system, and the evaluation of DAS’ influence on a driver are presented in Sections IV and V. Finally, section VI provides conclusions and suggests future work.

II. RELATED WORK

A. History

The idea of using simulators in driving training issues originates from the first flight simulator which was developed in the early 1910s [3]. This first simulator was created to be used for training pilots and for reducing operating costs required, compared to the use of real equipment [4]. The widespread use of flight simulators inspired researchers to apply the same concept for road vehicles [3].

B. Classification of driving simulators

The significant difference among driving simulators is focused essentially in the characteristics of the components...
used in their fabrication. These components define the level of fidelity (i.e., fidelity describes the degree to which the simulator’s characteristics faithfully replicate the driving task [5]) in a driving simulator, and for this reason, there is a direct economic dependence. Therefore, the driving simulators can be classified as low, medium, and high-cost according to their acquisition cost. Sometimes they can also be referred to as low-level, mid-level, and high-level driving simulators [6].

It is important to consider that the relation of cost-benefit, in the driving simulators depends on the goal and on the level of details needed to represent the real world. In Figure 1 Smith et al. emphasized that the best test track in a simulation has to combine the realism with the controlled situations, in function of the acceptance and safety [7]. For this reason, it is common to find driving simulators used in researches with a representative cost, regarding the use of a simulator for training.

C. Driving simulators used by the carmakers

The car makers use driving simulators to test a new system on a vehicle and to analyze how the system could influence the driver. For this reason, the driving simulation platforms used by the carmakers have the highest possible fidelity. Some examples of driving simulators used by the carmakers are the ULTIMATE by Renaults, Dynamic Driving Simulator by BWM, the Virtex by Ford, the Daimler AG driving simulator by Mercedes, and the driving simulator used by Nissan.

D. Driving simulators used in Research

In researches, the driving simulators have focused on the analysis of drivers’ behavior regarding some hazard situations, considering specific factors that could affect the driver (e.g. mobile phones, psychoactive substance use, test traffic signals, test driver assistance systems, among others). Considering [8], some variables measured by the driving simulators in researches are: i) vehicle speed; ii) vehicle place on road markings; iii) distance from the vehicle in front; iv) angle of the steering wheel; and v) amount of pressure applied to the brake pedal. Therefore, the driving simulators used in researches have a high fidelity. In researches the nature of the experiments is always dynamics (i.e., it may be necessary to change some characteristics for the driving simulator platform), and for this reason, the cost and performance of the awaited results could be compromised. Therefore, when a driving simulator is used in researches, it is necessary that the system structure be re-configurable[3].

Nowadays, there are multiple driving simulator platforms to develop and evaluate safety and emergency systems. Some of them are commercial and are more prominent by its technical characteristics, such as: the exact dynamic model used in the vehicles; integration with other software tools; and the fidelity of obtained results. Moreover, some of these platforms are used by some car makers and research centers 12. However, these driving simulator platforms have considerable acquisition and maintenance costs. Furthermore, the commercial driving simulator platforms also have a closed code, which allows only a minimal tool adaptation. Among some commercial driving simulator platforms most known, it is possible to find: CarSim and TruckSim, from Mechanical Simulation; PreScan, from TASS International; and CarMaker, from IPG Automotive GmbH.

There are also state of the art open-sourced driving simulator platforms, which, in addition to the economic benefit, allow the simulator platform to adjust to the needs of the user. However, one of the biggest drawbacks of these simulator platforms is that they do not provide users with adequate technical support. For this reason, they require a longer learning time. Among some open-source driving simulator platforms, there are: the TORSC [9], which is a highly portable multi-platform car racing simulation and used as an ordinary car racing game, as an AI racing game and as a research platform; and the Delta3D 3, which is a game and simulation engine appropriate for a wide variety of simulation and entertainment applications.

The driving simulator platform presented in this study is an open-sourced tool, with a middle level of fidelity (See Section III). Its objective is to contribute to the development of new safety and emergency systems, in the first stage of modeling. Similar to other open-sourced driving simulator platforms, this tool allows the users to configure its architecture by using different software tools (i.e., also open-source) that integrate it. The integration of the driving simulator platform with different software tools could be considered its main advantage. For example, this characteristic allows the integration of a vehicular communication model V2V in the platform.

Its main disadvantage is the fact that it does not count on an own dynamic model for the vehicles used in the simulations. In fact, the development of these dynamic models is one of our activities. In Section III, the driving simulator platform architecture is presented. In this Section, it will also be described the current dynamic model used in vehicles’ platforms.

E. Advantages and disadvantages of driving simulators

A major disadvantage of a driving simulator is that it is not able to represent the complexity of real road situations. For this reason, the validity and reliability of these tools are often questioned in the research area. However, despite of this disadvantage, research centers [10]-[18] and car makers [19][20][21] have a driving simulator available based on its significant advantages, such as, facilitate the design, development, and test of their proposals.

F. Validation of driving simulators

The models used in any driving simulator have to pass through tests and validations regarding the system, entity or idea that is being represented. The objective is to obtain a level of confidence in the driving simulator [22]. Engen in [23] considers that the driving simulators have to guarantee

1https://www.carsim.com/products/ds/index.php
2https://www.tassinternational.com/testimonials
3http://delta3dengine.org/
that the results obtained can be replicated (i.e., at least in a high percentage) in a real field-test. However, it is not always possible to compare the results of a driving simulator with the results of a real field-test (i.e., generally, for economic and logistic limitations), therefore it is necessary to find other alternatives. Engen proposes to consult a larger array of sources in order to find such alternatives and evidence of validity to be included in research models. Evidence of validity can be confirmed according to behaviors or specific events obtained from the tests results of the driving simulators.

III. DRIVING SIMULATOR PLATFORM ARCHITECTURE

The previous section had as objective to consider some relevant aspects of the driving simulators and their applications. For example, the relation between the realism and the control of the simulated situations in this driving simulation platform could be located at the test tracks circle’s center point, as shown in Figure 1. The main reason for why this particular point was selected in the test track is due to the lack of a higher level of realism in the models used in the current platform version (i.e., exact vehicle dynamics and environmental conditions of the simulation scenario, among others). Taking into account this lack of realism in the models of the current version, it is recommended the use of this simulation platform only in the first stage of the system development. The obtained result using the presented driving simulator platform could help the platform users to consider a test of one’s system in real field-tests, as it was made in the Subsection III-B.

The driving simulator platform proposed, is divided in three main parts: i) a generic simulator for academic robotics, ii) a middleware for robotic applications, and iii) a communications simulator. The generic simulator for academic robotics used is the Modular Open Robots Simulation Engine MORSE (i.e., the version used in this study was morse 1.2.2). MORSE is the main tool for the driving simulator platform. As described by Echeverría et al. in [24], MORSE is a free and open-sourced software tool, which can be adapted to the requirements of a particular system. MORSE has a large number of sensors, which can be used in the development and evaluation of the safety and emergency systems. Furthermore, MORSE proposes not only some methods to alter the input and output data (i.e., through modifiers) but also allows the creation of new sensors. For example, in this proposal it was created the Network Interface Communication NIC sensor to emulate a communication vehicular card.

The simulations in MORSE are based in Blender, which is a tool with the main purpose of creating images and 3D computer animations through a high level of graphic detailing (i.e., the version used was blender-2.73a). This characteristic of MORSE allows the creation of different 3D vehicle models (see Figure 3) with some dynamic characteristics (see Subsection III-A). With Blender it is also possible to create virtual scenarios for the simulations. In this proposal, were considered realistic scenarios to create virtual scenarios, using GPS coordinates, which were plotted in the Blender version used.

In the driving simulator platform, it is also used the Robot Operating System ROS [26], a middleware used in robotic, which has, as the main feature, libraries and tools to help software developers create robot applications (i.e., the version

---

Fig. 1: Selection of the best test track for an application, in function of the realism and the control. Taken from [7]

Fig. 2: Hardware used in the driving simulator platform

---

4https://www.openrobots.org/morse/doc/stable/contributing.html
used was indigo). Given that the main tool of the driving simulator platform, **MORSE**, lets an easy integration with **ROS**, it is possible to get data of each one of the sensors in every vehicle during a simulation. Through **ROS**, it is possible to implement models (i.e., see Subsections IV-A and V-A).

Finally, the communications simulator used in the proposed driving simulator platform architecture is **NS−3** (i.e., the version used was ns-3.19). With **NS−3** it is possible to develop codes using the programming languages C++ or python. Besides being open-source, **NS−3** also contains some tools for analysis of networks (e.g., pcap). In **NS−3** is possible to also find different models for computer networks (e.g., propagation and wifi among others). For example, in this study it was used the model Wireless Access in Vehicular Environments **WAVE**, to simulate the vehicle to vehicle communication **V2V**, used in sections IV and V. In the simulation using **NS−3**, the vehicular communication **V2V** has a node (i.e., in **NS−3** a node is a computer device that connects with a network) which represents a vehicle of the simulation executed in **MORSE**. Figures 5 and 11 present two different examples of the architecture’s configuration proposed in this section. In Figures 5 e 11 is possible to observe that the three main parts of the driving simulator platform architecture communicate with each other by using two sockets (i.e., A socket is a software endpoint that establishes bidirectional communication between a server program and one or more client programs). In the proposed driving simulator platform architecture, **NS−3** requests information to **MORSE** through the Socket − SS. When **MORSE** receives the request, **MORSE** responds to **NS−3** with a determined kinematic package data of the vehicle that has requested it. The answer of **MORSE** is sent using the Socket − SS. Then, **NS−3** receives the kinematic package data sent by **MORSE**, and directs the information to the correspondent node. Finally, the information received in **NS−3** is sent to **ROS** using the Socket − MS, with the objective to be used in a model. Data received by **NS−3** are considered package in network simulation. This process is used by each node during all simulations.

### A. Vehicular dynamic model

The actual Vehicular Dynamic Model (VDM) used in this driving simulation platform is based on the Bullet physics engine adopted for **Blender**. **Bullet Physics** is an open-sourced collision library, written in portable C++, used for professional collision detection and for rigid and soft body dynamics. This library is primarily designed for the use in games, visual effects, and robotic simulation [27].

**Bullets** approach to a vehicle controller is called a Raycast Vehicle. A raycast vehicle works by casting a ray for each wheel. By using the rays intersection point, we can calculate the suspension length and, consequently, the suspension force that is then applied to the chassis, keeping it from hitting the ground. The friction force is calculated for each wheel where the ray contacts the ground. This is applied as a sideways and forwards force [28]. A detailed example of the entire implementation of the VDM used in the vehicles of this driving simulator platform can be found at http://www.tutorialsforblender3d.com/Game_Engine/Vehicle/Vehicle_2.html.

### B. Validation of the driving simulator platform

In this proposal, the driving simulator platform validation is based on a study, presented by Filho et al in [29], about the simplification of the amount of control system parameters for the navigation of an Autonomous Vehicle (AV). Regarding validation, in this study, the assessment of the lateral control law in the driving simulator platform (i.e., it was used the **CaRINA 2 Blender** model, as seen in figure 3) is compared to the assessment of the same lateral control law used in the real testbed **CaRINA 2** [25] in two field-tests.

The driving simulator platform was used during the design and prototyping of the lateral control law. In the **CaRINA 2 3D** model, were not considered the longitudinal forces applied in the vehicle. In order to obtain results as close to reality as possible, Filho et al considered some errors (e.g., steering encoder errors, IMU orientation errors, localization errors and steering speed limitation) in the **CaRINA 2 Blender** model. The objective was to experiment with different controller setups.

Regarding the lateral control law assessment in the driving simulator platform, Filho et al considered a complex track in order to acquire the best result. For this reason, the simulated track, showed in Figure 4a, is 400 meters long, with closed and opened turns, which allowed the reproduction of different behaviors of the vehicle. Although this track does not exist in reality it has some characteristics of the real fields selected for the tests (e.g., Field of Test 1 (FOT1) and Field of Test 2 (FOT2)).

---

5The Java™ Tutorials
The \textit{FOT1} is a long loop track of 600 meters in length. In this trajectory, there are one-way asphalt roads with two lanes at all times, an accentuated left turn and a roundabout (see Figure 4b). Additionally, the \textit{FOT2} is a test track with a loop 1.6 kilometers long, going through 4 roundabouts, with asphalt roads of one single two-way lane and two one-way lanes (see Figure 4c).

Regarding the lateral control evaluation, the autonomous vehicle traveled around (i.e., in the driving simulator platform, as on both \textit{FOT}) the track selected for each case (i.e., \textit{DSP}, \textit{FOT1} and \textit{FOT2}). For several times, it was considered an average speed to get the cross track error and the orientation error in the control proposed.

Table I shows the average results of the tests done on different tracks (i.e., \textit{DSP}, \textit{FOT1} and \textit{FOT2}) with their values of cross track error and orientation error (i.e., \textit{C. error} and \textit{O. error}). It is possible to see that the cross track error in every test is practically the same. However, in the orientation error, the difference among the driving simulation platform and the other two field-tests is considerable. The main reason of this error is presented in the beginning of this section. Therefore, it is considered that this driving simulator platform has a medium fidelity.

| Tracks | Laps | Speed (Km/h) | C. error (m) | O. error (°) |
|--------|------|--------------|--------------|--------------|
| DSP    | 5    | 31.3         | 0            | 0.0808       |
|        |      | 90           | 0.6327       | 2.0173       |
| FOT1   | 3    | 20.4         | 28.8         | 0.0971       |
|        |      | 0            | -1.0397      | 1.0397       |
| FOT2   | 3    | 23.9         | -            | 0.088        |
|        |      | -            | 0.9225       | 1.0544       |

IV. COOPERATIVE VEHICLE SYSTEMS

The variety of applications of the cooperative vehicle systems and their developments are also increasing rapidly [30]. For instance, the Cooperative Adaptive Cruise Control system (\textit{CACC}) is an enhancement of the Adaptive Cruise Control system (\textit{ACC}) that uses inter-vehicle communication to perform safe cruising at shorter inter-vehicle distance [31]. In order to validate the proposed driving simulator platform architecture, a \textit{CACC} was used since it requires both V2V communication and the simulation of multiple agents. A state-of-the-art controller based on [32] was used.

A. Control

Given that $[\Delta x_i, \Delta v_i, \Delta a_i]^T$ are the distance, relative speed and relative acceleration between the $i$-th and the $(i-1)$-th vehicle of a platoon, respectively, $\Delta x_i^{safe}$ is the safety distance between vehicles given by

$$\Delta x_i^{safe} = hv_i + d$$

where $h$ is the headway time and $d$ is the desired distance for a stopped vehicle. From the desired safe distance, it is possible to define the control error $e_i$:

$$e_i = \Delta x_i - \Delta x_i^{safe}$$

and its derivative $\dot{e}_i$:

$$\dot{e}_i = v_{i-1} - v_i - ha_i$$

Assuming that the acceleration is directly controllable and the jerk is unbounded, a \textit{CACC} can be defined as a PD controller

$$\Delta a_i = K_p(\Delta x_i - hv_i - d) + K_d(\Delta v_i - ha_i)$$

However, implementing it as a pure PD controller term would cause instability depending on the operation frequency
since the current acceleration would be based on the previous one. In order to avoid this scenario, Equation 4 can be rewritten as:

\[ a_i = \frac{a_{i-1} + K_p(\Delta x_i - h v_i - d) + K_d(\Delta v_i)}{1 + K_d h} \]  

(5)

**B. Driving Simulator Platform Configuration**

![Fig. 5: Driving simulator platform configuration used in the cooperative vehicle system.](image)

Figure 5 presents the configuration of the driving simulator platform, used for the implementation and evaluation of the controllers model, present in the Subsection IV-A. In the configuration of its architecture (i.e., which was first presented in [33]) it is possible to observe the three main parts of the driving simulator platform, described in Section III. In Figure 5, the rectangle Robotic Simulator represents the MORSE simulator. In that rectangle is possible to observe the vehicles (i.e., robots Leader, Follower1, Follower2, Follower3) used in the validation of the CACC control (see Section IV-C2). In each vehicle simulated in MORSE, sensors (i.e., GPS, velocity, pose) were added to obtain data about position, orientation, and acceleration of the vehicles. Each information was used in the control model.

In Figure 5 is also possible to observe that each vehicle in Robotic Simulator uses a sensor Network Interface Communication NIC. The sensor NIC creates a message with different data (i.e., ID vehicle, position (x,y,z), yaw, velocity, and acceleration) of each simulated vehicle in MORSE. This message is shared with the rectangle Communication Network simulator that represents the NS – 3 simulator in Figure 5. The sending of messages between the Robotic Simulator and Communication Network simulator rectangles is done through the use of the SOCKET-SS. The sockets’ function shown in Figure 5 is detailed in the last paragraph of the driving simulator platform architecture description, before Subsection III-A.

The communications model of vehicle to vehicle V2V, implemented in the communications simulator NS – 3, consists in a precedent communications topology proposed by Y. Chen in [34]. The V2V communication is developed using four nodes, which are represented by each one of the vehicles used in MORSE (i.e., Leader, Follower1, Follower2, Follower3).

Each node in NS – 3 has a physical layer 802.11p Wi-Fi, a MAC layer and a definition of the hardwares physical properties. Moreover, for each node in NS – 3 was assigned an IP address, with the purpose of sending data packages between nodes. When the communications simulator NS – 3 obtains a message shared by the sensor NIC in MORSE, the V2V communications model, sends the obtained message to the corresponding vehicular node. This process is repeated throughout the system validation CACC.

After a node receives a message, through the communications model in NS – 3, it resends the message to the rectangle MIDDLEWARE, which represents ROS, using the SOCKET-MS. In this moment, it begins the execution of the control model presented in Subsection IV-A. As it is described in the Section III, ROS is a middleware that allows the implementation of models. It is divided into a set of nodes, and inside of this set there is one main node called Master. The Master node initializes ROS and allows the others nodes (i.e., these nodes are pieces of software) to communicate with each other. The ROS nodes are different from the nodes used in NS – 3.

In the validation of the controls model with the driving simulator platform, seven nodes were implemented in ROS: i) one node to get the sent messages through the SOCKET-MS (i.e., represented by Network simulator Output inside of rectangle Middleware); ii) and a pair of nodes CACC controller (see Figure 5) for each of the three Follower vehicles, with the purpose of determining the behavior of each Follower vehicle, based on the obtained messages from the node Network simulator Output. Finally, the control’s model effect can be observed in each Follower vehicle throughout the execution of the simulation in MORSE.

**C. Results**

As a validation for the driving simulator platform proposed in this study, two experiments were performed with the CACC control described in Subsection IV-A. In every experiment we assumed that the vehicles had access to their own position and velocity information. A video with the execution of a test is available in the website https://www.youtube.com/watch?v=WQsRR9ajiPc.

1) Delay effect on the CACC system: In the first experiment, two vehicles were used. The follower vehicle must keep a safe distance from the leader vehicle using the longitudinal control law described in Section IV-A. The purpose of this experiment was to evaluate the effect of the communication delays on the control of the follower vehicle. The scenario used for the experiments was an open space with no obstacles.

The tests have been performed using different wireless communication delays of 0.01 s, 0.1 s, 0.5 s, and 1.0 s. Figures 6 and 7 show the velocity of both vehicles obtained from the network simulator and the histogram of the difference in the velocity of the two vehicles, directly from the output of the NS – 3. The total time of the experiment was 33 s. From 0 s to 10 s, the leader vehicle accelerated from 0 to 37 m/s; from 10 s to 15 s, the velocity of the leader decreased to 8 m/s; from 15 s to 25 s, it increased again to
TABLE II: Statistics features of L-F velocity variations due to delay times in communication.

| Delay (s) |  |  |  |  |
|-----------|---|---|---|---|
| 0.01      | 0.0630 | 1.4620 | 2.1374 | 3.276 |
| 0.1       | 0.0540 | 1.2684 | 1.6089 | 3.582 |
| 0.5       | 0.0263 | 1.1926 | 1.4224 | 6.091 |
| 1.0       | -0.0121 | 1.7828 | 3.1786 | 8.007 |

Fig. 6: Leader vehicle Velocity (L, black), Follower vehicle velocity (F, red), velocities variation (L-F, blue), communication delay 0.01 s.

Fig. 7: Leader vehicle Velocity (L, black), Follower vehicle velocity (F, red), velocities variation (L-F, blue), communication delay 1.0 s.

37 m/s; and from 25 to 33 s, the leader decelerated until it achieved 3 m/s. Figures 6a and 6b show the velocity profile and histogram, respectively, of vehicle’s velocity obtained from the network simulator with a 0.01 s communication delay. We can notice that the follower’s velocity follows very close the leader’s velocity. Meanwhile, Figure 7a and Figure 7b present the velocity profile and histogram, respectively, from the simulation with a 1.0 s communication delay. The higher communication delay results on a non-continuous velocity profile. Also the histogram presented in Figure 7b shows a larger variance in the velocity difference when compared to the data presented in Figure 6b.

Table II shows the average results of the velocity difference of both vehicles obtained from the network simulator with communication delay of 0.01 s, 0.1 s, 0.5 s, and 1.0 s. It is possible to notice that the maximum velocity variation increases with higher delay values.

2) CACC system validation: In the second experiment four vehicles were used (one leader L and three followers F1, F2, and F3) to evaluate the CACC performance. Each follower vehicle only communicated to the vehicle immediately ahead of it.

Figures 8 and 9 present the simulation results for communication delays of 0.01 s and 0.333 s. Figures 8a and 9a present the velocity of each vehicle over time for the two delay settings. In Figure 8a, it is possible to notice that when the leader is accelerating its velocity is higher than the velocity of the followers. During the leader deceleration we have the opposite behavior. It happens due to the small delay in the control response, which is caused by the inertia of the vehicle, communication delay, and other physical aspects of simulation. In Figure 9a, it is possible to observe how the behavior of the three followers is affected by the increment of the communication delay among vehicles. Consequently, an irregular behavior in each acceleration/deceleration pattern can be noticed.

Figure 8b and Figure 9b show the distance between the vehicles over time with communication delay of 0.01 s and 0.333 s, respectively. As stated in the CACC control law, the distance varies according to the velocity of the vehicles. Nevertheless, it is possible to see that the distances between vehicles L–F1, F1–F2, and F2–F3 have the same profile. When the communication delay of 0.333 s was used, an increase in the difference of distance among vehicles of CACC system was observed.

Studies consisted of experiments done in a real field-test, with similar characteristics to the experiments done with the
proposed driving simulator platform, were used as a reference in order to contrast the results related to the effect of the time delay in the CACC control. This decision is justified in Subsection II-F.

Among the different studies researched, the most valuable was the one by Ploeg et al. in [35], where the authors proposed a design and validation of a Cooperative Adaptive Cruise Control CACC in a real field-test. In its experiment a fleet of six Toyota Prius III were used, since this model presents favorable technical characteristics for a real field-test. These vehicles were also equipped with other components in order to configure the CACC system appropriately (see Section IV in [35]).

Ploeg et al. considered the inclusion of time delay when developing the control implemented in the field-test (see section III-B in [35]). According to the authors, the time delay may produce disturbances in the control of vehicles (i.e., string stability), which can also be observed in Figure 9 of this study. For this reason, Ploeg et al. in [35] considered a time delay of 150 ms in the communication system. The magnitude of such time delay was considered to be the best suited by the authors for their experiment. This value of time delay was also favorable to our results. Table II in Subsection IV-C of this study, shows that a short time delay presents lower variations in the CACC system. The obtained results in the CACC validation, proposed by Ploeg et al. can be found in Section V-B of [35].

V. INFLUENCE EVALUATION OF A DRIVING ASSISTANCE SYSTEM IN A DRIVER

In order to evaluate the Driving Assistance System (DAS) influence on a driver, a test drive with twenty people was performed, considering two collision situations. In this evaluation, it was considered a driving simulator platform configuration that uses a DAS' model with vehicular communication (V2V), sensors on the vehicles (e.g., GPS, IMU), a highway scenario, and the possibility to add an external hardware (i.e., Joystick G27).

A. Driving Assistance System model

For this study, it was implemented the first case (i.e., following) of the study proposed by Sebastian et al. in [36]. In the test drive, the case following occurs when two or more vehicles have the same orientation and direction on the same lane. Algorithm 1 illustrates the situation.

In the case following, we have a vehicle A following vehicle B or vice versa. Figure 10 shows an example of case following. The algorithm line 2 checks if the vehicles (i.e., A and B) are in the same orientation. This could be done by comparing the direction angles between the vehicles, in which \( \theta_A \approx \theta_B \) (see Figure 10). In the algorithm 1, this comparison is determined as \( |\theta_A - \theta_B| < \beta \), in which \( \beta \) is a constant to be determined.

Next in algorithm line 3, the lateral distance \( d_p \) and longitudinal distance \( d_a \), between the A and B vehicles have to be computed. These distances are represented in Figure 10, and may be calculated using equations 10 and 11. However,
Algorithm 1 Case following

1: for each pair of vehicles \((A, B)\) do
2:     if \(|\theta_A - \theta_B| < \beta\) then
3:         compute the distances \(d_p, d_s\) and \(d_a\)
4:     find the follower \(f\) and the leader \(l\)
5:     calculate safe distance \(d_{sf}\)
6:     if \(d_a < d_{sf}\) then
7:         Collision warning
8:     end if
9:     end if
10: end for

Fig. 10: Vehicles with the same direction. Adapted from [36]

to get equation 10 is necessary to consider the value of the next variables:

\[
x_{A'} = \sin \theta_A + x_A \quad , \quad y_{A'} = \cos \theta_A + y_A
\]  
(6)

\[
u = \frac{(x_B - x_A) (x_{A'} - x_A) + (y_B - y_A) (y_{A'} - y_A)}{\sqrt{(x_{A'} - x_A)^2 + (y_{A'} - y_A)^2}}
\]  
(7)

\[x_P = x_A + u(x_{A'} - x_A)
\]  
(8)

\[y_P = y_A + u(y_{A'} - y_A)
\]  
(9)

\[d_p = \sqrt{(x_P - x_B)^2 + (y_P - y_B)^2}
\]  
(10)

The longitudinal distance \(d_p\) is also calculated based on the Euclidean Distance between the \(A\) vehicle’s position and the found coordinate \(P\) (see Figure 10) using equations 6,7,8 and 9:

\[d_a = \sqrt{(x_P - x_A)^2 + (y_P - y_A)^2}
\]  
(11)

On the other hand, it is also necessary to compute the lateral safety distance \(d_{ls}\) with the equation 12. The distance \(d_{ls}\) allows to decide if the two vehicles are in the same lane, considering the width of \(A\) vehicle \((\omega_A)\), the width of \(B\) vehicle \((\omega_B)\), and a minimal lateral safety distance \(d_{mls}\) that have to be defined.

\[d_{ls} = \frac{w_A}{2} + \frac{w_B}{2} + d_{mls}
\]  
(12)

In order to find out if a vehicle is a leader or a follower (i.e., algorithm’s line 4), the original algorithm had some changes.

In this new approach, it is defined a vector \(\overrightarrow{AP}\) between the points \(A\) and \(P\) (see Figure 10), after it is found the angle \(\phi\) between the vectors \(\overrightarrow{AP}\) and \(\overrightarrow{VA}\) (i.e., \(\overrightarrow{VA}\) is the \(A\) velocity vector), using equation 13:

\[
\phi = \arccos \frac{\overrightarrow{AP} \cdot \overrightarrow{VA}}{|\overrightarrow{AP}| |\overrightarrow{VA}|}
\]  
(13)

If \(\phi < \pi/2\) the \(A\) vehicle is a follower otherwise it is a leader. Finally, it is calculated the safety distance \(d_{sf}\) using Equation 14 (i.e., algorithm’s line 5).

\[d_{sf} = d_{min} + v_f t_r + \frac{1}{2} \left( \frac{v_f^2}{a_f} - \frac{v_l^2}{a_l} \right)
\]  
(14)

In which \(d_{min}\) is the minimal distance between \(A\) and \(B\) vehicles to avoid a collision; \(v_f\) is the following vehicle speed; \(t_r\) is a parameter of driver’s reaction time; \(a_f\) is the following vehicle maximum deceleration; \(v_l\) is the leading vehicle speed and \(a_l\) is the leading vehicle maximum deceleration. The value of the distances \(d_a\) and \(d_{sf}\) are necessary to determine a Collision Warning condition (i.e., algorithm line 6).

B. Driving Simulator Platform Configuration

Fig. 11: Driving simulator platform configuration used to evaluate the DAS\(^{\prime}\) influence in a driver.

Figure 11 presents the configuration of the driving simulator platform’s architecture. Its new configuration was the result of carefully planned experiments to validate the selected DAS model influence on a driver. These experiments (see Subsection V-C) required an interaction among vehicles, a highway scenario, a new V2V communication network topology and the use of a joystick that would allow the interaction of a user with the driving simulator platform.

Inside of the rectangle Robotic Simulator, which represents the MORSE simulator in Figure 11, it is possible to observe the representation of four different vehicles. For the 3D model of the vehicles in MORSE, it was considered the use of sensors (i.e., GPS, velocity, pose and NIC) as it is presented in Subsection IV-B.

For this new driving simulation platform configuration, in the message of the sensor Network Interface Communication NIC, it was considered a new value. This new value is the
distance that exists between the Ego-vehicle and each vehicle present inside the highway scenario of the simulation. The distance is calculated in the sensor NIC using the vehicles location data. This information is important because this value determines the communication range of the vehicles in the new communications model implemented in \( NS - 3 \).

The rectangle Communication Network simulator, that represents the \( NS - 3 \) simulator in Figure 11, obtains the sent messages by the sensor NIC, again through SOCKET-SS (see Section III). The communications model implemented in \( NS - 3 \) represents, through nodes, each one of the vehicles used in MORSE. The structure of the V2V communication model allows the node Ego-vehicle to receive messages from the other vehicles nodes present in the simulation scenario. If the distance value among the vehicles (i.e., obtained in the sent message by NIC) is greater than a range of 300 m, the communications model determines that the node Ego-vehicle cannot receive messages. When other vehicles enter the communication range of the Ego-vehicle, the communication begins. The V2V communications model do not allow the node Ego-vehicle to send messages to the others vehicles nodes, as it can be observed in the rectangle Communication Network simulator in Figure 11.

When the Ego-vehicle node receives a message, from the node of another vehicle, the Ego-vehicle's node sends this information for the rectangle MIDDLEWARE, through SOCKET-MS. In Figure 11, the rectangle MIDDLEWARE, which represents ROS is composed by the nodes: i) Network Simulator output; ii) Driving Assistance System (DAS); and iii) Joystick (i.e., A ROS system is comprised of a number of independent nodes, each of which communicates with the other nodes using a publish/subscribe messaging model. The ROS' nodes are different of nodes used in \( NS - 3 \).

The node Network Simulator output receives a message obtained from SOCKET-MS and makes it available for the ROS' system. The DAS node is the piece of software in the ROS' system where it is implemented the model presented in the Subsection V-A. For the model execution, the DAS node needs the data obtained from the Network Simulator output node and sensor data from Ego-vehicle, which are obtained from MORSE (see Figure 11). MORSE allows a direct communication with ROS. Finally, the joystick node allows the interaction between the users and the driving simulator platform. In this node are implemented the configuration and the execution of the joystick functions, needed in the experiments. As it can be observed in Figure 11, this node also communicates directly with the Ego-vehicle in MORSE.

C. Experiments

With the objective to validate how the implemented DAS' model influences in a driver, two of thirty-seven potential collision situations were chosen (i.e., a lead vehicle stops and a vehicle changing lanes in the same direction). These collision situations are proposed by the National Highway Traffic Safety Administration (NHTSA) in [37]. Those collision situations are the result of several crash analysis, based on the accidents data collected in the United States in 2009, which could have been avoided using a system with V2V communications technology. From this result, the risk typology in this section was developed. In Subsection III-A of [37] it can be observed the other thirty-five collision situations.

For the evaluation of the DAS' model influence on a driver, postgraduate students from the science institute of mathematics and computers (ICMC-USP) were invited to participate in the experiment. It was not disclosed to the participants the main purpose of the experiment, and each participant was required to have a valid drivers license and driving experience of a manual gear shift. Only twenty postgraduate students voluntarily accepted the invitation to participate. From the 20 participants 7 were female and 13 were male, and the participants' age averaged 27.2 years (i.e., \( \text{sd} = 3.65 \) years). They were divided into two groups: i) without DAS (i.e., 4 women, and 7 men); and ii) with DAS (i.e., 3 women, 6 men), in order to verify the DAS model influence in the two collision situations the drivers from both groups were exposed to.

For the implementation of the two collision situations, it was developed a virtual highway based on a specific segment of the highway Washington Lúiz, in the São Paulo state (i.e., from the bridge at the main access way to São Carlos Street until the turn at exit 240). In this virtual highway, the GPS coordinates of the real highway segment were used in order to obtain the real shape of the highway structure. Moreover, Blenders tools were also used (see Section III) to add some characteristics in the virtual highway, such as: i) asphalt under ideal conditions, ii) the grass around the highway, iii) the use of objects and infrastructure present in the real highway. For the virtual highway scenario, it was considered a sunny environment without climatological changes and one-way lanes.

Four vehicles were used in the virtual highway scenario. Only one of these vehicles was driven by the participants (i.e., Ego-vehicle) using the joystick G27, while the other three vehicles were moving autonomously. For the experiment, each participant received instructions on how to operate the joystick G27, which consisted of a steering wheel, accelerator, brake and clutch for the gear shift (See Figure 2). Before the experiment began and data were collected, the participants had time to adapt to the driving simulator platform.

The three vehicles that were moving autonomously were controlled by an implemented algorithm, which defined their paths within the virtual scenario. Two of these vehicles were used on the collision situations (i.e., a conventional red vehicle and a truck), and the third vehicle was used to distract the drivers (i.e., a conventional orange vehicle). The sequence of events of the simulation consisted of the first collision situation, the encounter of the drivers with the orange vehicle and the second collision situation. As follows, is a detailed description of the characteristics considered for each collision situation:

1) The lead vehicle stopped: to configure the danger in this situation, it was developed a routine in the algorithm for the red vehicle moving autonomously. In this routine, the red vehicle detects the presence of the Ego-vehicle in a range of 30 m from its rear. It then activates the brakes in the vehicle controller, used by Blender in the red vehicle model (see
Section III-A), instantaneously decelerating the red vehicle until it is completely stopped.

When a participant starts the experiment, the Ego-vehicle is parked on the virtual highway. After the participant drives for a while in the virtual scenario path, the user encounters the other three vehicles. Two of these vehicles, the red and orange, are moving in each lane of the highway (i.e., highway with just two lanes), at a velocity of 70 km/h, which prevents an overtaking maneuver between them. This is the moment the red vehicle detects the Ego-vehicle and the first collision situation initiates.

In this collision situation, the participant from the group using the DAS model receives a warning sound, which serves to alert the participant of the collision risk that the red vehicle ahead could represent. The sound signal is enabled when the condition in line six of Algorithm 1 is accomplished. In Figure 11 it is possible to observe the relationship between the node DAS in ROS with the sound signal (i.e., AL).

2) A vehicle changing lanes in the same direction: In this second collision situation, it was used a 3D model of a truck as one of the autonomous vehicle in the simulation. Here, the algorithm implemented for the truck first percepts (i.e., in a range of 15 m) the presence of the Ego-vehicle approaching behind of it in a different lane. Then, the routine in the algorithm activates the steering control of the vehicle controller, used by Blender in the truck model, causing the truck to move towards the Ego-Vehicle’s lane. The sudden change of lane by the truck, moving at 70 km/h on the highway, initiates the second collision situation of the experiment. Like in the first situation, a warning sound alerts the participant from the group using DAS of the trucks presence.

The orange vehicles sole purpose is to distract the attention of the driver between the two collision situations, but without causing any further risks to the driver, who overtakes the orange vehicle without difficulties. A video of these experiment is available on https://www.youtube.com/watch?v=H4lXVlq_8II.

D. Results

To obtain an experience feedback, when the platform users completed the experiment, each participant completed a survey containing seven questions. With the survey results was possible to observe a worrying situation from two of the questions. In Did you break the speed limit? 65% of the users answered affirmatively. In Did you make some dangerous maneuvers during the trial route? 70% of the users also answered affirmatively. This result confirms that drivers between 21 and 30 years of age have more than 50% chance to cause an accident, as also seen in Stiller et al. [38], where it analyzed the driving assistance systems impact in the last thirty years.

Figure 12 presents the results of two other questions from the survey. Figure 12a shows that the most dangerous situation for the driving simulator platform users was the first one (i.e., the lead vehicle stopped). Figure 12b, in relation to the question Did you think that the Driver Assistance System helped in the test?, in the group with DAS, only 56% of the users considered the DAS’ model useful in the test.

To evaluate the DAS influence in a driver, the driving simulator platform made a test log, with the objective to obtain some data of each user during the experiment, described in Subsection V-C. The data obtained in log are Ego-vehicle’s velocity, the Ego-vehicle’s brake application rate, and the initial time of the collision situation.

The initial time’s value of the collision situation helps to find the magnitude of the Ego-vehicle’s velocity and the Ego-vehicle’s brake application rate in the exact moment, when beginning the collision situation inside of each users log (see vehicle situation and the truck situation in Figures 13 and 14). The Ego-vehicle’s velocity, and the Ego-vehicle’s brake application rate are the only data considered to analyze the DAS influence in drivers.

In order to analyze the results, were considered the mean, median, standard deviation, and the maximum and minimum values from the number of samples (i.e., participants) in the experiment by each group (see the second paragraph in Subsection V-C). In Tables III and IV can be observed this statistical parameters. In addition to these parameters, it was used a bootstrap method to determine a 95% confidence intervals (i.e., Bmin and Bmax in Tables III and IV), with the objective to compare the data set of Ego-vehicle’s velocity and the Ego-vehicle’s brake application rate for each group. The bootstrap method is used for the results analysis, with small sample drawn from a distribution that differs from a normal distribution. In this approach, no assumptions were made about the original population [39].
TABLE III: Results group without DAS

| Parameter | VCS1 (Km/h) | BCS1(%) | VCS2 (Km/h) | BCS2(%) |
|-----------|-------------|---------|-------------|---------|
| Mean      | 68.98       | 11.32   | 95.07       | 0.35    |
| Median    | 66.43       | 0.0     | 92.71       | 0.0     |
| Std       | 17.02       | 24.49   | 28.86       | 1.18    |
| Min       | 42.95       | 0.0     | 54.59       | 0.0     |
| Max       | 101.72      | 80.0    | 165.33      | 3.94    |
| BD min    | 60.28       | 1.75    | 54.59       | 0.0     |
| BD max    | 79.61       | 32.74   | 116.41      | 1.07    |

TABLE IV: Results group with DAS

| Parameter | VCS1 (Km/h) | BCS1(%) | VCS2 (Km/h) | BCS2(%) |
|-----------|-------------|---------|-------------|---------|
| Mean      | 66.03       | 7.57    | 73.32       | 0.0     |
| Median    | 62.11       | 0.0     | 77.91       | 0.0     |
| Std       | 10.01       | 19.96   | 27.74       | 0.0     |
| Min       | 57.44       | 0.0     | 34.57       | 0.0     |
| Max       | 87.91       | 60.62   | 113.8       | 0.0     |
| BD min    | 61.30       | 0.52    | 55.98       | None    |
| BD max    | 74.59       | 28.61   | 89.99       | None    |

By comparing the statistical parameters values shown in Tables III and IV, for the Ego-vehicle’s velocity data (i.e., VCS1, Ego-vehicle’s velocity in the first collision situation), it is possible to observe that the mean, the median, and the bootstraps confidence intervals (i.e., Bmax and Bmin) in the two groups do not have significant differences.

Moreover, for the Ego-vehicle’s brake application rate data (i.e., BCS1, Ego-vehicle’s brake application in the first collision situation), the values of the median, in Tables III and IV, show that the brake practically was not activated in both groups since the most probable value in the median was zero. This fact shows that, during the first collision situation, a considerable majority of the experiment participants, in both groups, did not react in the beginning of the collision situation.

Another aspect observed in Tables III and IV for BCS1, was that the bootstraps confidence intervals (i.e., Bmax and Bmin) do not present a considerable difference. Therefore, these values can determine that the use of the DAS model, during the first collision situation did not have an impact on the driver. In Figure 12a it is possible to observe this fact, given that it was the red vehicle (i.e., used in the experiment to generate the first collision situation) that, according to the survey completed by the experiment participants, caused the majority of the crashes.

On the other hand, in the second collision situation, the differences between the statistical parameters’ values, in Tables III and IV are considerable, for the Ego-vehicle’s velocity data (i.e., VCS2, Ego-vehicle’s velocity in the second collision situation). Only the standard deviation between the groups has a slight difference. The most considerable difference between the two groups can be found between the bootstraps confidence intervals for VCS2.

Therefore, with a 95% of confidence it is possible to evidence that the participants group using the DAS model was moving at approximately 20 km/h slower than the other participants group which did not use the DAS model, in the
beginning of the second collision situation. Figures 13 and 14 represent this fact, considering the relation between the labels, velocity (i.e., Ego-vehicle velocity) and truck situation.

Comparing the statistical parameters obtained for the Ego-vehicle’s brake application rate data (i.e., BCS2, Ego-vehicle’s Brake application in the second collision situation), it was possible to verify that all the participants of the group that used the DAS model, did not use the Ego-vehicles brake during the beginning of the second dangerous situation.

Consequently, after analyzing the experiments results, it is possible to determine that the DAS model, in the second collision situation, could have an influence on the driver. Figure 14b is an example of a participants log from the group that used the DAS model. In this figure is possible to observe that Ego-vehicle’s brake was activated seconds after that the DAS model adverted the driver.

Finally, considering that the two collision situations were consecutive, it could be determined that the experiments participants, who used the DAS model, understood the models effect only after the first collision situation. Therefore, although the DAS model works, it could be thought that the DAS model does not have the level of influence necessary for a driver to avoid an accident.

VI. CONCLUSION

For the driving systems development, the use of a driving simulator platform can facilitate the work for the design time as well as the economic, temporal and logistical aspects, among others. However, with the obtained results, the use of the driving simulator platform cannot be considered as definitive. The simulation tools should only be used in a starting level.

A simulation tool could have best results if the models used in the simulation platforms (e.g., sensors, vehicles, communications and so on) have a feedback from obtained results in the field-test. Nevertheless, this feedback requires flexible driving simulation platforms (i.e., mainly in research) that allow the researcher to modify the characteristics in the setup of the driving simulation platform.

Taking into account these question, in this study, it is presented an adaptable driving simulator platform that allows the platform users to adjust its characteristics (e.g., hardware and software), for an experimental implementation, regarding technical and economic requirements. The two applications presented in Sections IV and V allow us to prThis decision is justified by Engen inove the flexibility of the driving simulator platform.

The control system development, for the autonomous vehicle navigation, is an example of the simulation platform validation. With the use of the proposed driving simulator platform was possible to develop and evaluate the control system, in starting level, and to optimize the available resources.

The use of the driving simulator platform has allowed to analyze the influence of the DAS on a driver. Considering the results, it is clear that the use of an informative approach (i.e., a warning sound) to alert the driver about the danger is not enough. Therefore, although the DAS model works, This model does not guarantee that the driver can avoid an accident.

For future work, the next version of the Vehicular Dynamic Model (VDM) in the driving simulator platform has to consider some aspects, such as: i) to edit and easily replace, the different components of the VDM, ii) to use the main parameters of real vehicles, in the vehicles’ model design, and finally, iii) the model must be developed to be stiff and efficient from a numerical point of view. It is also important to mention that the simulation platform will be available to other research groups, with the objective to enable a comparison between this proposal and other approaches.

ACKNOWLEDGMENT

The authors acknowledge the support granted by CNPq and CAPES.

REFERENCES

[1] Global plan for the decade of action for road safety 2011-2020. Technical report, United Nations, 2010.
[2] Z. Sitavancov, M. Hjek, Intelligent transport systems. Technical report, European Commission DG Mobility and Transport, 2010.
[3] D. D. E. R. Ingenieurwissenschaften, B. Hassan, E. Dakahilia, A Design Framework for Developing a Reconfigurable Driving Simulator. Prof, G., and Tr, A. 2014.
[4] E. Blana, A survey of driving research simulators around the world (Working paper no. 481). Leeds: University of Leeds, 1996.
[5] H. Jamson, Cross-Platform Validation Issues. In: Fisher, D.; Caïd, J.; Rizzo M.; Lee, J. (Eds.): Handbook of Driving Simulation for Engineering, Medicine, and Psychology. CRC Press Taylor and Francis Group, USA, pp. 12.1-12.13, 2011.
[6] B. Lang, A. M. Parkes, S. Cotter, R. Robbins, C. Diels, P. Vanhulle, G. Turi, E. Bekiaris, Benchmarking and classification of CBT tools for driver training, 2007.
[7] M. R. H. Smith, G. J. Witt, D. L. Bakowski, A Final Report of SAFety VEhicles using adaptive Interface Technology (Task 15): SAVE-IT Summary and Benefits Estimation, (May), 150, 2008.
[8] C. Castro, Human factors of visual and cognitive performance in driving. CRC Press, 2008.
[9] B. Wymann, E. Esri, C. Guionneau, C. Dimitrakakis, R. Coulom, A. Sommer, TORCS: The Open Racing Car Simulator, v1.3.6, 2014.
[10] C. Radin-Brown, A. Williamson, and M. Lenn, Can driving simulation be used to predict changes in real-world crash risk, Rapport technique, Monash university, Accident Research Centre, 2009.
[11] D. Stavrinos, J. L. Jones, A. A. Garner, R. Griffin, C. A. Franklin, D. Ball and P. R. Fine, Impact of distracted driving on safety and traffic flow, Accident Analysis & Prevention, 61, 63-70, 2013.
[12] A. Knapp et al., Code of Practice for the Design and Evaluation of ADAS, Preventive and Active Safety Applications, eSafety for road and air transport, European Commission Project, Brssel, 2009.
[13] O. Gietelink et al, Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations, Vehicle System Dynamics, 569-590, 2006.
[14] J. S. Goncalves, R. Rossetti, J. Jacob, J. Goncalves, C. Olaverri-Monreal, A. Coelho, R. Rodrigues, Testing Advanced Driver Assistance Systems with a serious-game-based human factors analysis suite, Intelligent Vehicles Symposium Proceedings, pp.13,18, 2014.
[15] C. Maag, D. Muhlbacher, H. Kruger, Studying Effects of Advanced Driver Assistance Systems (ADAS) on Individual and Group Level Using Multi-Driver Simulation, Intelligent Transportation Systems Magazine, IEEE, v. 4, n. 3, p. 4554, 2012.
[16] B. Bifulco, L. Pariota, F. Galante, A. Fiorentino, Coupling instrumented vehicles and driving simulators: Opportunities from the DRIVE IN2 project, Intelligent Transportation Systems (ITSC), pp.1815,1820, 2012.
[17] S. Noth, J. Edelbrunner, I. Iossifidis, An integrated architecture for the development and evaluation of ADAS, Intelligent Transportation Systems, pp.347,354, 2012.
[18] K. Yoshimoto and T. Sueotomi, The history of research and development of driving simulators in Japan, Journal of mechanical systems for transportation and logistics, 1(2), 159-169, 2008.
