Realistic LiDAR with Noise Model for Real-Time Testing of Automated Vehicles in a Virtual Environment

Juan P. Espineira, Jonathan Robinson, Jakobus Groenewald, Pak Hung Chan and Valentina Donzella

Abstract— **The global Connected and Autonomous Mobility industry is growing at a rapid pace. To ensure the successful adoption of connected automated mobility solutions, their safety, reliability and hence the public acceptance are paramount. It is widely known that in order to demonstrate that L3+ automated systems are safer with respect to human drivers, upwards of several millions of miles need to be driven. The only way to efficiently achieve this amount of tests in a timely manner is by using simulations and high fidelity virtual environments. Two key components of being able to test an automated system in a synthetic environment are validated sensor models and noise models for each sensor technology. In fact, the sensors are the element feeding information into the system in order to enable it to safely plan the trajectory and navigate. In this paper, we propose an innovative real-time LiDAR sensor model based on beam propagation and a probabilistic rain model, taking into account raindrop distribution and size. The model can seamlessly run in real-time, synchronised with the visual rendering, in immersive driving simulators, such as the WMG 3xD simulator. The models are developed using Unreal engine, therefore demonstrating that gaming technology can be merged with the Automated Vehicles (AVs) simulation toolchain for the creation and visualization of high fidelity scenarios and for AV accurate testing. This work can be extended to add more sensors and more noise factors or cyberattacks in real-time simulations.**

Index Terms—**Autonomous and automated vehicles, light detection and ranging (LiDAR), Noise, Perception Sensor, Rain, Real-Time Simulation, Sensor Models.**

I. Introduction

THE automotive industry is currently focused on Advanced Driver Assistance Systems (ADAS) applications and lower levels of automation. However, several Original Equipment Manufacturers (OEMs) and technology companies are also developing higher levels of autonomy with many promises, from improved road safety to economical and societal benefits [1]. These systems rely on an array of perception sensors (camera, LiDAR, RADAR, ultrasonic, etc.) to recognize the environment around the vehicle, plan and adapt its actions as it goes from point A to B [2]. The Society of Automotive Engineers (SAE) have defined 6 levels of autonomy which are widely used by many research groups and developers [3]. These levels range from level 0 (L0), in which the vehicle's system will only provide warnings and information to the driver, to levels 4 and 5 (L4-L5) in which the Autonomous Control Systems (ACS) can take full control of the vehicle within its designed parameters without the need for any human intervention. This control includes bringing autonomously the vehicle to a safe state in fault conditions, such as system failures or situations that the system does not recognise; on the contrary, L3 vehicles will hand back to the driver/supervisor under these conditions.

To demonstrate that Automated Vehicles (AVs, L3 and above) are actually safer with respect to the human drivers, upwards of several millions of miles have been quoted by different research groups as the number of miles that would need to be driven by these technologies [4], [5]. Furthermore, any real world incidents involving automated vehicles have an enormous negative impact on the developer/manufacturer and also on the public's perception of this technology [6], [7].

Traditionally, OEMs use the V-model or a variant of it to develop new vehicle systems [8]. Conversely, the rapid increase of computing power and software capabilities enables the viability of system virtual testing earlier in the development process. However, virtual testing and controlled testing are not sufficient to demonstrate the ACS safety, more testing environments need to complement the development process, as shown in Fig.1. The testing continuum shows the different testing options available for AV electronic systems and sensors, and the benefits of bringing the testing towards the simulation side at any stage of the V-model.

There are several facilities available to test at different stages of the testing continuum. For example, in the UK, real world

This work was submitted on 16 October 2020. This work was supported by HVM Catapult UK. Valentina Donzella acknowledges that this work was supported by the Royal Academy of Engineering under the Industrial Fellowships scheme.

Juan P. Espineira, Jonathan Robinson, Dr Jakobus Groenewald, Pak Hung Chan and Dr Valentina Donzella are all with WMG, The University of Warwick, 6 Lord Bhattacharyya Way, Coventry, CV4 7AL, United Kingdom (emails: juan.espineira@warwick.ac.uk, jonathan.robinson@warwick.ac.uk, j.groenewald@warwick.ac.uk, pak.chan.1@warwick.ac.uk, v.donzella@warwick.ac.uk)

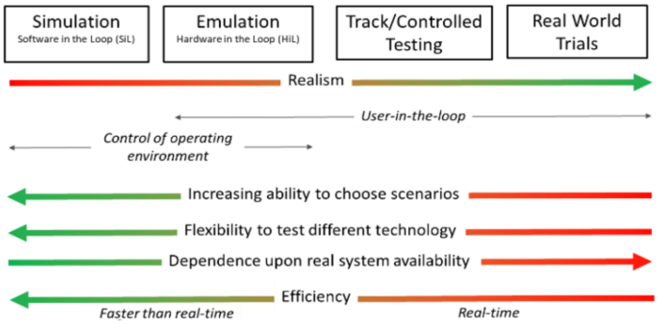
Simulation
Software in the Loop (SiL)
Emulation
Hardware in the Loop (HiL)
Track/Controlled Testing
Real World Trials
Realism
User-in-the-loop
Control of operating environment
Increasing ability to choose scenarios
Flexibility to test different technology
Dependence upon real system availability
Efficiency
Faster than real-time
Real-time

Fig. 1. Schematic view of WMG's proposal of the Testing Continuum for Automated/Intelligent Vehicles, where Simulation performs tests entirely within a simulated environment; Emulation tests hardware within a simulated environment; Track/Controlled Testing tests in a real restricted environment that was created for testing; Real World Trials tests in the real environment such as on a public road.

trials can be performed through Midlands Future Mobility which contains a 300-mile network of roads, such as highways and rural roads [9]. In track/controlled testing, e.g. the Horiba Mira Proving Ground or Millbrook Proving Ground, there is a higher degree of controllability in which actors can be controlled as well as some environmental parameters [10]. Testing virtually can bring many benefits such as earlier proving, higher degree of control and lower costs for prototyping. From the left side of the testing continuum, several commercial software such as IPG CarMaker and rFpro provide a fully simulated environment to carry out testing [11], [12].

Moreover, there are several research and industrial groups developing immersive simulators and test beds. These simulators allow for testing virtually different aspects of the autonomous systems and can entail the emulation of hardware, environmental condition, noise factors and road users. Examples of these facilities are WMG 3xD simulator and AVL's Driving-Cube^{TM} [13]–[18].

Three aspects are fundamental when implementing virtual testing (simulation or emulation based):

1) *Fidelity*
It is imperative for the simulator to provide sufficient fidelity models: of the environment, to have realistic rendered and natural actors; of the sensor models that will capture the environment. Simulations can be used by OEMs to train and validate the machine learning algorithms with fringe or dangerous scenarios [19]. If the ACS under test in the simulation does not receive the same sensor input as in the real world, the vehicle behavior cannot be truly predicted and validated in the simulation.

2) *Real-Time Operations*
During testing of a mixture of real and emulated/simulated components, it is difficult to accelerate or slow down the simulation due to the physical constraint of the rate at which data can be gathered and processed by the hardware. Altering the simulation time can generate artifacts in the results or can hide timing issues and race conditions that might arise in real-time.

3) *Possibility to Inject Noise Factors*
Simulations with hardware in the loop have an advantage over proving grounds in that there is a higher control of the environment. This capability allows for repeatability of tests, and also the testing of the same scenario with subtle changes, such as adding different noise factors, to observe changes in the ACS reaction. Noise can cause data loss and distortion, having a profound impact on the perception algorithm; a framework to analyse noise factors affecting automotive sensors have been recently proposed [20].

To the best of our knowledge, this paper presents, for the first time, a real-time LiDAR simulation model which includes beam propagation effects and a probabilistic rain model, evaluating real-time the effect impact of the rain droplets on each LiDAR beam. The proposed model can generate data comparable to current automotive sensors, in terms of data rates and format, and run simultaneously with the visual rendering for a 360° projection in a vehicle in the loop simulator.

II. Background

LiDAR was first used as a perception sensor for an autonomous road vehicle in 2004, when it was used on a car competing in the DARPA challenge [21]. Equipping LiDAR to vehicles will assist towards higher levels of autonomy alongside the other sensors, such as RADAR and vision [22]. LiDAR boast excellent angular and range resolution making it an attractive addition to the vehicle sensor suite [22], [23].

A. LiDAR working principles

The LiDAR unit emits numerous narrow beams of near-infrared light with circular/elliptical cross-sections, which reflect off of objects in their trajectories and returns to the detector of the LiDAR sensor [24]. There are three common techniques for measuring the distance and position of objects:

1) *Time-of-Flight (ToF)*
Most LiDAR sensors use ToF for measurement where the time between emission of the beam and receipt of the returned beam determine the distances and position of the object ($z = c \cdot \Delta t/2$) where z is the distance to a target, c is the speed of light and Δt is the time between emission and receipt of the beam. Dependent upon the specification of the LiDAR, a different number of these beams are emitted by different laser sources. Each detection from a beam can be used to create a point (i.e. *x,y,z* coordinates) where that reflection took place which, when combined, form a pointcloud [24], [25].

2) *Amplitude Modulated Continuous Wave (AMCW)*
For AMCW, the intensity of the emitted light is modulated at a particular frequency which undergoes a phase shift when reflecting off an object [24]. This phase shift can then be used to calculate the distance to the object.

3) *Frequency Modulated Continuous Wave (FMCW)*
With FMCW, the frequency of the emitted light is incrementally changed in a ramp or triangular manner. The received signal will consequently have a constant phase delay equivalent to the TOF. Multiplying this by the gradient of the slope gives the frequency of the light at the time of measurement and hence the distance to the object. This technique also enables relative velocity to be determined through the Doppler shift [24].

A recent in depth review on LiDAR technologies and its working principles can be found in [2].

B. Effect of adverse weather on LiDAR performance

In this subsection, recent works investigating effects of weather on LiDAR performance are presented. One of the main issues with LiDAR is degraded performance in the presence of rain [26]. If a LiDAR beam intersects with a raindrop at a short distance from the transmitter, the raindrop can reflect enough of the beam back to the receiver such that it is detected as an object [22], [26]. Alternatively, raindrops can absorb some of the emitted light [27]. This degrades the range performance of the LiDAR demonstrating the need to model the effect that rainfall will have on the sensor [26], [27].

Goodin *et al.* produced a simplified rain model [28]. They made assumptions that the rain was homogenous and uniformly scattered and neglected the spatial variation of the hard target. They combined the laser pulse energy, speed of light, effective receiver area, transmitter and receiver efficiencies into a single coefficient, C_s, where C_s is the product of the above parameters over 2. This reduced the LiDAR equation to:

$$P_r(z) = \frac{C_s\rho}{z^2}\exp(-2\alpha z) \quad (1)$$

where α is the scattering coefficient, ρ is the backscatter coefficient and P_r is the received power. Since Cs is constant for a particular sensor they further reduce the equation by introducing relative power $P_n = P_r/C_s$ and representing α via a power law $\alpha = 0.01R^{0.6}$ which takes into account the rainfall intensity, R, in mm/h. Hence the equation becomes

$$P_n(z) = \frac{\rho}{z^2}\exp(-0.02R^{0.6}z) \quad (2)$$

To implement the measured distance variation due to rain, Goodin *et al.* derived an equation using work from Filgueira *et al.* [29], and their new distance (z') measurement was

$$z' = z + N(0{,}0.02z(1 - e^{-R})^2) \quad (3)$$

However, the measurements taken by Filgueira *et al.* were only using rainfall rates up to 8 mm/h meaning the model is only valid up to 8 mm/h rainfall intensity [29]. Rain showers up to 2 mm/h is considered a light shower, 10 mm/h a moderate shower and above that is considered a heavy shower, with rainstorms potentially reaching 100 mm/h [27], [30]. For continuous rain less than 0.5 mm/h is considered light, up to 4 mm/h moderate and more than 4 mm/h heavy [30].

In their experiment, Filgueira *et al.* performed real world tests using a Velodyne VLP-16 to investigate range, intensity and number of points for the LiDAR with respect to different materials, with and without rain [29]. Their results suggest that rain has a more pronounced effect on the intensity compared to the range measurement at increasing rain intensities. Both intensity and number of points reduced with increasing rainfall as expected, but at differing rates for different materials. Also climate chambers have been used to investigate the effect of weather on LiDAR performance, e.g. as presented in [26]. The main focus was fog, however rain at 55 mm/h was also tested. As in [29], a decrease in both intensity and number of points was found in the presence of rain. Another paper complimented this by showing that rain reduces the maximum range of LiDAR sensors [31].

Other simulations models have been proposed through the years; e.g. a simulation model with a series of physical equations to predict and quantify the attenuation of LiDAR beams with rain was developed in [27]. Their model was found to be within 7.5% of the actual attenuation when compared to real data. A model using a "hit ratio" for each beam was implemented in [32], whereby the "hit ratio" determines the likelihood of the beam hitting a raindrop. Points within the pointcloud will only be modified if their hit ratio is greater than a selected threshold value, e.g. 10%. This model was then tested in a specialised test facility where it was demonstrated that some points appeared closer to the sensor. However, the authors noted that this method was computationally expensive [32].

Recent works have been increasingly focusing on a more realistic modelling of the interaction between the LiDAR beams and the rain droplets, for example Berk *et al*. takes into account a probabilistic representation of the number of drops in the volume surrounding the sensor and depending on the rainfall intensity [23]. The number of drops, $N(D)$, is calculated by modelling the drop size distribution using an exponential distribution:

$$N(D) = 8000 \cdot \exp(-4.1 \cdot R^{-0.21} \cdot D) \quad (4)$$

where D is drop size in mm. By integrating this with respect to drop size, the average number of drops in a unit volume can be determined and multiplying by the LiDAR beam volume gives the average number of drops in the beams path. This model enables the simulation of LiDAR detection performance by means of comparing probability of real object detection against probability of false alarms due to rain droplets [23].

Recent publications have also investigated the validation of noise models and the combination of noise effect on sensor models in virtual environments [33], [34]. The paper by Hasirlioglu and Riener in 2019 proposes a methodology to train and validate a virtual rain simulation on a regathered set of weather noise free dataset [33]. In the paper by Byeon and Yoon, an automotive simulation software, Prescan, was used with a rain noise model implemented. The proposed model can simulate rain precipitations in three different regions, and the impact on the received detected power is evaluated, via a Matlab LiDAR model [34].

TABLE I

STRENGTHS AND WEAKNESSES OF RECENT PUBLICATIONS IN LIDAR NOISE ANALYSIS AND MODELLING

Author	Strengths	Weaknesses
Berk *et al.* [23]	Simplifies LiDAR equation Provides functionality to calculate backscatter and extinction Used theoretical LiDAR designs to test effect on Probability of detection	Uses method to calculate probability of detection rather than pointcloud adjustment
Heinzler *et al.* [26]	Real conditions physically replicated	Mathematically heavy Only considers 55mm/h of rain
Guo *et al.* [27]	Simulation results compared to experimental results Detailed representation of beam in the presence of rain	No method to directly adjust pointcloud inferable from results Computationally heavy
Filgueria *et al.* [29]	Simplistic and easy to implement Considers introduced error on distance measurement	Only valid for rainfall measurements up to 8mm/h
Hasirlioglu and Riener [32]	Real conditions physically replicated	Computationally expensive Only considers 100mm/h of rain

Building on these works, and the works summarised in Table I, our paper presents a LiDAR model created using Unreal

Engine 4. Using this software, our simulation can generate real-time both visual images and LiDAR data with the possibility to add real-time noise factors to the virtual environment, such as rain. Furthermore, our LiDAR model generates data in the same format of commercial LiDAR pointclouds, therefore can be used by any hardware and/or software that can be interfaced with the real sensor. Table 1 compares some of the strengths and weaknesses of the papers reviewed in this section.

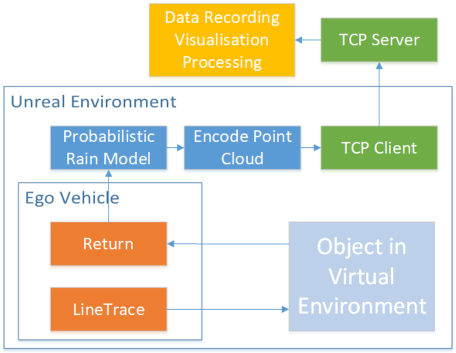

Fig. 2. Schematic view of the software architecture for obtaining a noisy 3D pointcloud from the virtual environment. Unreal uses the LineTrace Function to generate each one of the ideal 3D points associated with the received beams; then the noise model is applied to each point and the noisy pointcloud generated.

III. Methodology

With the importance of the perception system for the decision making of automated vehicles, it is imperative for simulated sensor data to be realistic and to take into account noise factors in order to be used in the development and testing stages. With this aspect in mind, we have reviewed over thirty different simulation software suites for automotive use based on accessibility to the software/hardware, usability and potential for building on additional capabilities, such as adding enhanced sensor models and noise factors. We shortlisted six software suites that were assessed in depth; the shortlist contains a range of open source software and commercial software [16]. Out of the six software suites, we have chosen to use Unreal Gaming Engine to create our environment and sensor model within the WMG 3xD simulator, and for this reason the simulations described in this paper are based on Unreal Engine. Several open source software such as Carla and Airsim uses Unreal Engine as a backend. Young *et al.* has also used Unreal Engine to for simulation of autonomous vehicles [35]. Furthermore, Unreal Engine offers the best flexibility and fidelity in comparison with other shortlisted software, offering the capability to develop new adds on in C++, such as our LiDAR model.

A. LiDAR Model

The LiDAR sensor model created is based on the available LineTrace Function which is a raycast method of the Unreal Engine. Raycasting is based on finding the first intersection between a 3D object and a line with given start and finish. In our model, each one of the LiDAR beams is represented by one line generated by the LineTrace Function, and the 3D objects will be the road users in the surrounding of the ego-vehicle, see Fig.2. The origin of the lines is placed on the sensor position on the ego car and the end point will depend of the angles of emission; in the case there is an object in the line trajectory the function then returns a 3D point (the intersection point) and the physical properties of the object like reflectivity and the normal vector (to the surface of the object) of the impact. The distance to the object and the properties returned from the raycast are then used in an attenuation equation to calculate a relative returned intensity; values of returned intensity above the detection threshold will add the specific point to the pointcloud. The relative returned intensity is an inverse exponential equation, modified from (2). The number of raindrops encountered along the beam trajectory increasingly reduces the intensity returned and may result in an object not detected and therefore a False Negative (FN) point. Where there is no object in the beam path, the LineTrace Return outputs the max range of the LiDAR sensor. The complete and noisy pointcloud is encoded and then transferred through TCP to its destination. In our model, this is transferred to a visualisation software, but can also be to an ACS for example.

B. Real-Time Rain Model

The rain model developed builds upon the work in [23]. The drop size distribution is calculated using the exponential distribution in (4). However, this can be replaced by any distribution, should this be found to be more accurate for the application of interest [23]. This is then integrated with respect to drop size with limits of 0.5 mm and 6 mm, as raindrop size is typically in this range [36]. This gives the average number of raindrops in a unit volume so multiplying by the beam volume gives the average number of raindrops, μ, in the beams path. For simplicity, no beam divergence is assumed, however the effect of beam divergence can be assessed by discretising the beam into a number of sections, k, and summing the individual beam volumes, as shown in (5-6).

$$V_{beam} = \sum_{i=1}^{k} V_{beam_i} \tag{5}$$

$$V_{beam_i} = \pi * r_{beam_i}^2 * Z/k \tag{6}$$

Where r_{beam_i} is the radius of the beam in section i and Z is the distance to the object. The average number of raindrops, μ, is then used as the mean for the Poisson distribution as in Berk *et al.* (7) [23].

$$p(n) = \exp(-\mu) * (\mu^n/n!) \tag{7}$$

If one or more raindrops are detected within the beam path based on (7), an additional partial return will be created due to water reflection. The partial return can be interpreted as a false positive, as explained below.

The aim of our model is to be able to generate in real-time an ideal pointcloud or a pointcloud affected by the rain in the same format and data-rate of the pointcloud generated by a real LiDAR sensor. In order to achieve that, our model needs to be efficient enough as to not slow down the simulation system's computational efficiency. We can generate up to 60 frames per second for the simulation visuals and hence the LiDAR model should be able to generate a pointcloud compatible with the simulation speed and with the number of points generated per second by commercial LiDAR sensors.

The proposed sensor model is able to create an output signal with the same format and framerate of a commercial 128 channel LiDAR, without compromising the real-time performance of our virtual environment, i.e. the WMG 3xD

Simulator. The LiDAR and noise models work as schematically shown in Fig. 3 that represents the decision logic behind the noisy pointcloud composition; this process entails: removing points due to high attenuation; returning a rain drop point; returning an actual point on one of the objects in the scene. In case of no real object in the beam trajectory, the code applies the rain model by comparing RIRD, the cumulative Returned Intensity of Rain Drops in the beam path (the statistical number of raindrops is p(n) from (7)), to the detection threshold, DT. A False Positive (FP) will be returned when calculated RIRD > DT. In the case of object detection instead, if there are no raindrops in the beam path (p(n) < 1) the point is returned regularly. If there are raindrops in the beam path (p(n) > 1) the returned raindrop intensity RIRD is compared to the Returned Intensity of the Object (RIO), and the higher of the two is further compared to the receiver DT. If rain intensity is higher than the object one and high enough to be detected we would have a false positive (in the position of the first rain droplet in the beam trajectory), otherwise no detection and therefore a false negative.

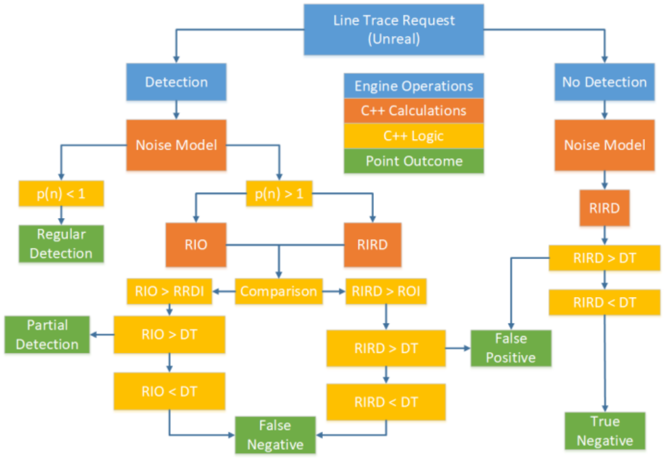

Fig. 3. The flow process for the logic of the rain model where p(n) is the number of drops returned by the probabilistic rain model, RIO is the Returned Intensity of the Object, RIRD is the Returned Intensity of rain drops, and DT is the Detection Threshold for intensity

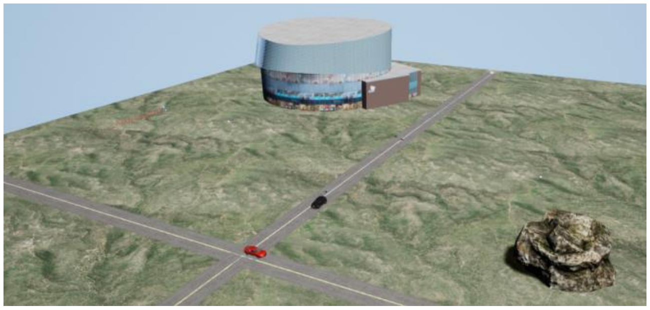

Fig. 4. A view of the simulation scene created in Unreal Engine. The ego vehicle is the black vehicle at the centre. Other objects in the scene includes a red vehicle, a boulder and a building at 23m, 47m and 78m respectively from the ego vehicle).

C. Experimental Setup

For the purposes of this paper, an experiment has been carried out to test real-time the deployment of the rain model in combination with the LiDAR based on the LineTrace function. The experiment consists of running the simulation with the modelled LiDAR installed on the virtual vehicle roof under no rain and then under two rainfall intensity conditions (10 mm/h and 50 mm/h which are the limit of moderate and heavy rain showers as defined by the MetOffice [30]). Furthermore, we have parameterised our model based on a Velodyne 32 channel LiDAR with a vertical aperture of 40° (evenly distributed between -25° and 15°), a horizontal resolution of 0.2°, and a refresh rate of 10Hz [37]. The relative detection threshold (between the emitted and received power) value of the LiDAR detector can also be tailored to match the specification of any real LiDAR units. In our simulations we used three different relative detection threshold values, namely $1.0\text{x}10^{-5}$, $0.5\text{x}10^{-5}$ and $0.1\text{x}10^{-5}$, representing different limits of detection of the LiDAR detector, similar to those used in [23]. Increasing the detection threshold would mean that the LiDAR unit can filter out more noise, but it might mask some distant object with a low intensity return. A LiDAR with a lower detection threshold is able to detect signals reflected by objects even if they have been remarkably attenuated, but would not be able to filter out all the noise in the environment, due to rain in our case. The values $1.0\text{x}10^{-5}$ and $0.1\text{x}10^{-5}$ represent these extreme cases in our simulation, whereas going either above or below them won't change the output pointcloud any further. An optimal detection threshold for our model should enable the model to mimic the pointcloud output by a real LiDAR sensor under the same environmental conditions. The value $0.5\text{x}10^{-5}$ represents this case, and it can be fine-tuned based on pointcloud measurement performed with a real LiDAR sensor. The output of the different simulations (varying rain rate and DT) will be the resulting pointclouds that are processed for analysis. The scene, built with Unreal Engine, used in our simulation consists of the ego vehicle in an empty environment with only

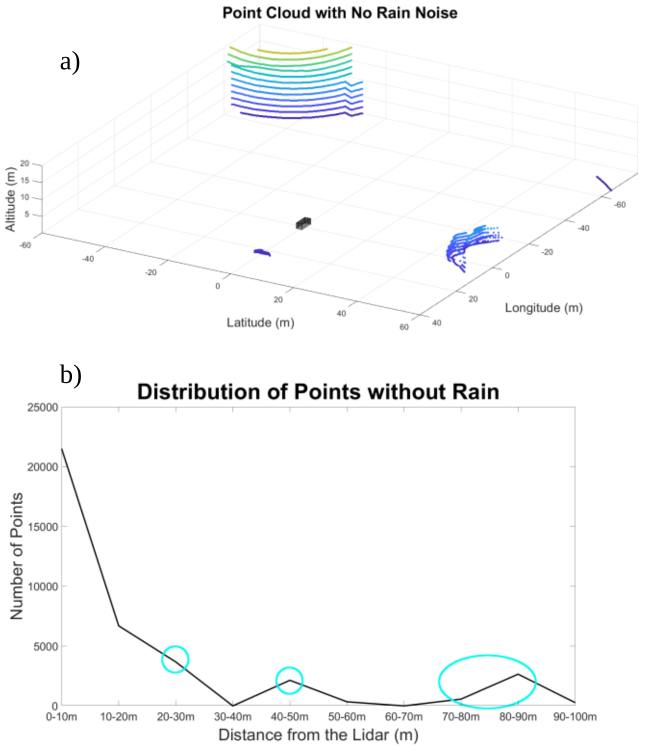

Fig. 5. (a) PointCloud generated by the sensor model (ground points have been filtered out) and (b) number of points detected verses distance in ideal weather conditions. In (b) the three circles represents positions of the three targets

three objects at specified distances illustrated in the Fig. 4. These objects are a red sedan, a rock and a building (at 23m, 47m and 78m respectively from the ego vehicle).

IV. Results

First of all, we performed a simulation of the LiDAR detection by the ego vehicle in the scene in Fig. 4 with ideal environmental conditions. In this case, the attenuation is dependent mainly on the optical path of the beam [23]. The ideal pointcloud generated is reported in Fig. 5a, where we filtered out the ground points. The figure clearly demonstrates that the three objects in the scene are accurately detected, with a neat representation of their size and shape. Our model generates 57600 line traces per frame. From these emissions, 36642 points are returned detections in the pointcloud, of which circa 2101, 738 and 134 belong to the three objects in the scene (the building, the boulder and the second vehicle respectively). Fig 5b represents the number of points in the pointcloud at different distances from the ego vehicle, including ground points. The points detected in vehicle proximity (between 0m-20m) are mainly ground points, conversely points in the regions 20m-30m, 40m-50m, and 70m-80m represent the objects in the scene. It is worth noting that the number of ground points decreases steeply when the distance increases, and also that due to the resolution and accuracy degradation with the distance [2].

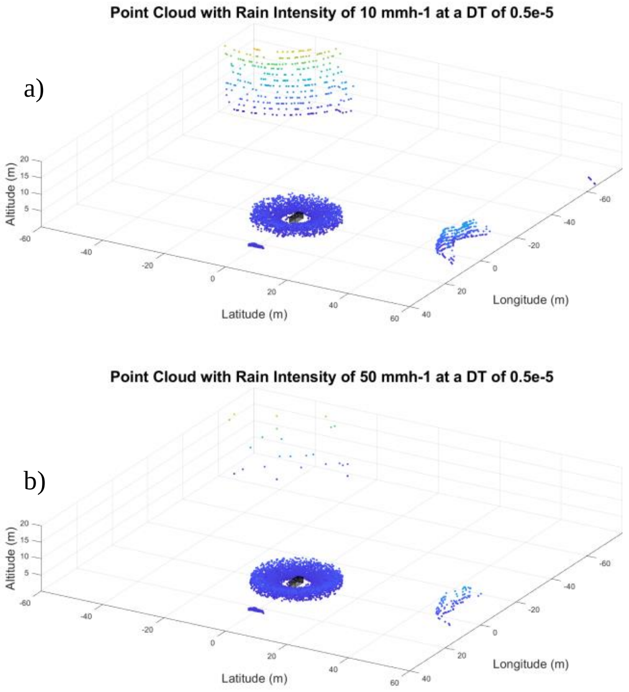

Fig. 6. (a) Simulated pointcloud with a rain intensity of 10 mm/h and a threshold of 0.5 e-5 and (b) Simulated pointcloud with a rain intensity of 50 mm/h and a threshold of 0.5 e-5

Then we generated the simulation for the two rain conditions and the three detection threshold values, as described in previous section. The pointclouds generated with the same threshold value (0.5 x 10^{-5}) and the 2 rainfall intensities are depicted in Fig. 6a-b. It can be observed that several points not associated to real objects are present in the short range (up to around 50 m of distance from the ego vehicle), and these points are not on the ground plane and therefore cannot be filtered out. Nonetheless the real objects in the scene have a decreased number of points representing them in the pointcloud, and this degradation is more evident for the highest rainfall.

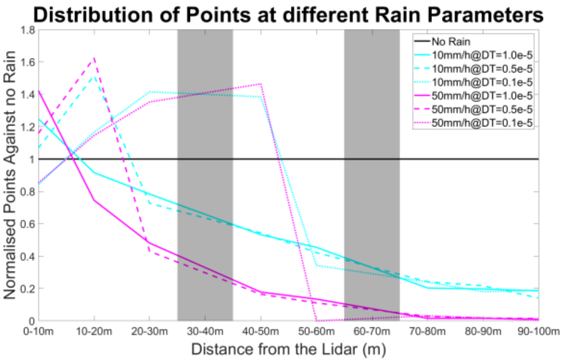

Fig. 7. Normalised distribution (using ideal distribution, Fig. 5b, as reference) of detected points for different testing parameters of rain and detection threshold. In the shadowed areas, points have been interpolated

In Fig. 7 we compared the detected points at given distance intervals with respect to the ideal detection presented in Fig. 5b. There is a trend of detecting more points in the short range (with a peak value at increasing distance as the detection threshold decreases). On the contrary, it is observed that there is a significant reduction in points at the farther distances from no rain to 10 mm/h to 50 mm/h, and this trend is similar for all the thresholds. There is also a significant drop of the detected points as the rainfall intensity increases. This roughly matches the range reduction observed by [38]. Similarly, the amount of points increases at closer range as observed by the same authors.

Fig. 8 shows the amount of False Positive (FPs) points (detected points do not corresponding to real objects in the scene) and False Negatives (FNs) points (points in scene belonging to real objects that are not detected due to the rain noise) as a function of the detection threshold used in the model and for the two rain conditions; the simulation with no rain has been used as the ground truth. It is observed that FPs and FNs are affected in an opposite way by the selected DT: FPs (continuous lines in Fig.8) increase at a lower intensity thresholds and FNs (dashed lines) increase at a higher intensity thresholds. However, as expected, these two types of misdetections increase when the rainfall increases (in our simulations from 10mm/h to 50mm/h).

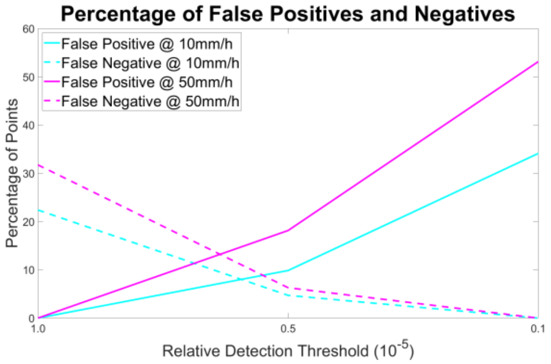

Fig. 8. Percentage of false positive (solid lines) or false negative (dotted lines) points in the pointcloud as a function of the LiDAR detector threshold; the higher rain rate is characterised by an increase in mis-detected points with respect to the lower rain rate

V. Discussion

An important aspect when using simulations for testing Automated Vehicle functions is to feed them with sensor signals that are of the same type of the signals generated by real sensors. Our Unreal LiDAR model can run real-time in our 3xD simulator, generate the same information stored in a real LiDAR pointcloud and can produce pointclouds with 128 LiDAR channels at 20 Hz. Comparing this model with real sensor performance, we are able to simulate commercial 360° LiDARs, such as the Velodyne's VLS-128 and the Ouster OS1-128.

In Fig. 4a, it is clearly shown that in ideal weather condition, the pointcloud generated by our model has a richness of details typical to LiDAR sensor. However, our interest is not only to generate a realistic LiDAR output, but also to have the possibility to study, in our immersive 3xD simulator, the effect of noise. In particular, in this study, we focused on rain noise, as rainy weather is a challenging condition for environmental perception sensors, and as mentioned before, particularly for LiDAR [26]. In our 3xD simulator we have the possibility to real-time change the weather conditions, to visually emulate a rainy day. Now, with the model that we developed and presented, we can combine this kind of scene with its effect on the LiDAR sensors mounted of the ego-vehicle. This achievement opens the possibility to study the combined real degradation of visual and LiDAR sensors in the case of rainfall of different intensities. Furthermore, our LiDAR model has the possibility to vary the detector threshold, therefore can be tailored to the performance of a specific commercial LiDAR.

The effect of increasing rainfall is clearly shown in Fig. 5. As the rain increases, the pointcloud becomes noisier and the objects in the scene are more difficult to detect, confirming the decrease in range reported in [29]. These visual inspections of the pointcloud are confirmed by the distribution of points versus distance from the sensor for different rainfall intensities, as reported in Fig. 6. All of the generated pointclouds in rainy weather show an increased number of points in the short range (up to 50 m) due to several rain drops erroneously detected by the LiDAR; conversely, the number of points detected in medium range (50-100m) is dramatically reduced, and the reduction becomes more relevant as the distance from the sensor increases. We observed about 40%-80% reduction in number of points for 10 mm/h rain intensity, and about 80%-100% for 50 mm/h rain intensity. The effect of heavy rain obviously hinders the detection of objects in medium range. We can also note that the threshold has an effect on filtering the noise in the short range, the higher the threshold the more evident is this effect; however, this reduction of noise has an impact also on detecting points with lower intensity reflected by distant objects. This trend is confirmed in Fig. 6; in fact, the highest threshold has clearly a lower number of false positives with respect to the other two lower values of detection threshold. This decrease means that a lower number of rain points are detected in the pointcloud. However, the highest threshold has also a significant increase in false negatives, as explained above. Overall, the medium threshold has a good balance between FPs and FNs, which accounts for under 25% of the number of points emitted in the frame in total. As expected, the lower threshold has a dramatic increase in FPs, as several rain droplets in the short range are detected with this setting.

These pointcloud roughly match what is described by Jokela *et al.* in their experiment, however, that was for light snow and few false positives are observe as opposed to this work [39].

VI. Conclusion

This work presents a real-time LiDAR model implemented using the Unreal gaming engine that is used in automotive testing for high fidelity visual rendering of the testing scenarios. The sensor model is combined with a noise model (in our case a probabilistic rain model) without affecting simulation real-time performance and paving the way to combining more noise sources using the framework presented in [20]. We tested LiDAR performance without and with different rainfall intensities, and studied the effect of the detection threshold on false positive and false negative detections. The detection threshold can be fine-tuned to produce with our model an output pointcloud with the same format and speed of a real LiDAR, and with realistic data degradation due to environmental noise. We are currently working on modelling more noise factors, including adversarial attacks and blinding of the LiDAR.

References

[1] "UK Connected and Automated Mobility Roadmap to 2030," Zenzic, London, U.K., 2019.

[2] C. Hsu *et al.*, "A Review and Perspective on Optical Phased Array for Automotive LiDAR," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 1, pp. 1-16, 2021.

[3] "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE, Warrendale, PA, USA, 2018.

[4] N. Kalra and S. M. Paddock, "How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?," 2016.

[5] X. Zhao, V. Robu, D. Flynn, K. Salako, and L. Strigini, "Assessing the Safety and Reliability of Autonomous Vehicles from Road Testing," in *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, 2019, pp. 13–23.

[6] National Transportation Safety Board, "Collision between vehicle controlled by developmental automated driving system and pedestrian, Tempe, Arizona," Washington, D.C., 2019.

[7] National Transport Safety Board, "Collision Between a Sport Utility Vehicle Operating With Partial Driving Automation and a Crash Attenuator," Washington, D.C., 2020.

[8] B. Liu, H. Zhang, and S. Zhu, "An Incremental V-Model Process for Automotive Development," in *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, 2016, pp. 225–232.

[9] Zenzic, "Who we are," 2020. [Online]. Available: https://midlandsfuturemobility.co.uk/about-us/. [Accessed: 16-Jun-2020].

[10] Horiba Mira Ltd, "Proving Ground," 2020. [Online]. Available: https://www.horiba-mira.com/Proving-Ground/. [Accessed: 16-Jun-2020].

[11] IPG Automotive GmbH, "CarMaker: Virtual testing of automobiles and light-duty vehicles," 2020. [Online]. Available: https://ipg-automotive.com/products-services/simulation-software/carmaker/. [Accessed: 16-Jun-2020].

[12] rFpro, "Driving Simulation," 2020. [Online]. Available: http://www.rfpro.com/driving-simulation/. [Accessed: 16-Jun-2020].

[13] J. Groenewald, "IV Facilities," 2020. [Online]. Available: https://warwick.ac.uk/fac/sci/wmg/research/cav/ivfac. [Accessed: 03-Jun-2020].

[14] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "Development of a Drive-in Driver-in-the-Loop Fully Immersive Driving Simulator for Virtual Validation of Automotive Systems," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–4.

[15] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "Identifying a gap in existing validation methodologies for intelligent automotive systems: Introducing the 3xD simulator," in *Intelligent Vehicle, IEEE Symposium*, 2015, pp. 648–653.

[16] J. Groenewald *et al.*, “Certifying a synthetic environment for CAV validation and verification,” in *Driving Simulation Conference Europe 2019 VR – 18th Driving Simulation & Virtual Reality Conference & Exhibition*, 2019.

[17] S. Khastgir, G. Dhadyalla, S. Birrell, S. Redmond, R. Addinall, and P. Jennings, “Test Scenario Generation for Driving Simulators Using Constrained Randomization Technique,” *WCX 17 SAE World Congr. Exp.*, 2017.

[18] A. Gruber *et al.*, “Highly Scalable Radar Target Simulator for Autonomous Driving Test Beds,” vol. 0, no. 1, pp. 147–150, 2017.

[19] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, “A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research,” *Sensors*, vol. 19, no. 3, pp. 1–29, 2019.

[20] P. H. Chan, G. Dhadyalla, and V. Donzella, “A Framework to Analyze Noise Factors of Automotive Perception Sensors,” *IEEE Sensors Lett.*, vol. 4, no. 6, pp. 1–4, 2020.

[21] R. Behringer *et al.*, “The DARPA grand challenge - development of an autonomous vehicle,” in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 226–231.

[22] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, “The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car,” *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 103–111, Jun. 2019.

[23] M. Berk *et al.*, “A Stochastic Physical Simulation Framework to Quantify the Effect of Rainfall on Automotive Lidar,” in *SAE Technical Papers*, 2019, vol. 2019-April, no. April.

[24] S. Royo and M. Ballesta-Garcia, “An overview of lidar imaging systems for autonomous vehicles,” *Appl. Sci.*, vol. 9, no. 19, Oct. 2019.

[25] M. E. Warren, “Automotive LIDAR Technology,” in *2019 Symposium on VLSI Circuits*, 2019, pp. C254–C255.

[26] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork, “Weather Influence and CLassification with Automotive Lidar Sensors,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1527–1534.

[27] J. Guo, H. Zhang, and X. J. Zhang, “Propagating characteristics of pulsed laser in rain,” *Int. J. Antennas Propag.*, vol. 2015, 2015.

[28] C. Goodin, D. Carruth, M. Doude, and C. Hudson, “Predicting the Influence of Rain on LIDAR in ADAS,” *Electronics*, vol. 8, no. 1, 2019.

[29] A. Filgueira, H. González-Jorge, S. Lagüela, L. Díaz-Vilariño, and P. Arias, “Quantifying the influence of rain in LiDAR performance,” *Meas. J. Int. Meas. Confed.*, vol. 95, pp. 143–148, Jan. 2017.

[30] National Meteorological Library, “National Meteorological Library and Archive Fact Sheet 3 -Water in the atmosphere,” 2012.

[31] R. H. Rasshofer, M. Spies, and H. Spies, “Influences of weather phenomena on automotive laser radar systems,” *Adv. Radio Sci.*, vol. 9, pp. 49–60, 2011.

[32] S. Hasirlioglu and A. Riener, “A Model-Based Approach to Simulate Rain Effects on Automotive Surround Sensor Data,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, p. 2609,2615.

[33] S. Hasirlioglu and A. Riener, “A General Approach for Simulating Rain Effects on Sensor Data in Real and Virtual Environments,” *IEEE Trans. Intell. Veh.*, vol. 8858, no. c, pp. 1–1, 2019.

[34] M. Byeon and S. W. Yoon, “Analysis of Automotive Lidar Sensor Model considering Scattering Effects in Regional Rain Environments,” vol. XX, pp. 1–11, 2020.

[35] P. Young, S. Kysar, and J. P. Bos, “Unreal as a simulation environment for off-road autonomy,” in *Proceedings Volume 11415, Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2020*, 2020, pp. 1–8.

[36] MetOffice, “Rain,” 2020. [Online]. Available: https://www.metoffice.gov.uk/weather/learn-about/weather/types-of-weather/rain. [Accessed: 15-Jun-2020].

[37] Velodyne LiDAR Incorporation, “VLP-32C User Manual Rev.B,” San Jose, 2018.

[38] M. Kutila, P. Pyykonen, H. Holzhuter, M. Colomb, and P. Duthon, “Automotive LiDAR performance verification in fog and rain,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2018-Novem, pp. 1695–1701, 2018.

[39] M. Jokela, M. Kutila, and P. Pyykönen, “Testing and validation of automotive point-cloud sensors in adverse weather conditions,” *Appl. Sci.*, vol. 9, no. 11, 2019.

Juan P. Espineira was born in Ciudad Autonoma de Buenos Aires, Argentina in 1991. He received the MEng degree in mechanical engineering from Coventry University, Coventry, in 2018

From 2017 to 2018, he was a Teaching Assistant with the Control Engineering Department of Coventry University. Since 2019, he has been a Project Engineer with WMG, University of Warwick. Juan is a Project Engineer He is developing simulation systems for the 3xD Simulator using Unreal Engine and interfacing to multiple systems on Autonomous Vehicles. He has background in manufacturing, structural analysis, electronics and control systems. He has worked on projects involving CAV Safety Standards and the use of Simulation to develop reliable and robust test cases for CAV technology.

Jonathan Robinson was born in Norwich, England in 1996. He graduated with a BSc in Mathematics from the University of Warwick in 2018. In 2019, he was awarded an MSc in Smart, Connected and Autonomous Vehicles from the University of Warwick, WMG, which he then graduated with in early 2020.

Since 2019, he has been working as a Graduated Trainee Engineer at WMG, University of Warwick. During this time he has worked on projects within the 3xD Simulator at WMG, projects involving the design and development of Human Machine Interfaces for the mitigation of motion sickness, and the simulation of components for optical phased array LiDAR chips.

Jakobus Groenewald Dr Jakobus Groenewald graduated with a First-Class BEng Honours degree in Electronic Engineering from the University of Bangor in 2010. He subsequently completed his Ph.D. in optical communications systems at the University of Bangor in 2014.

In November 2014, Jakobus was appointed as a Research Fellow on the ABACUS project that forms part of the WMG High Value Manufacturing team to conduct research into innovative business models and design approaches for extending the in-service battery life of future low carbon vehicles. In 2016 he was appointed as Test Facilities Engineer for the 3xD simulator at WMG. In 2017 Jakobus was promoted to Lead engineer in WMG's Connected and Autonomous Vehicles Research Group. His research interest includes Smart, Connected and Intelligent Vehicles focusing on ADAS systems, V2x communications, automotive sensor spoofing, modelling, simulation and emulation.

Pak Hung Chan was born in Hong Kong China, in 1996. He has completed an integrated master’s degree with Honours in Mechanical Engineering at the School of Engineering, The University of Warwick, UK, in 2018. He is currently working part-time towards a Postgraduate Certificate in Automotive Technology at WMG, The University of Warwick.

Since 2018, he has been working as an Engineer at WMG, focusing on research around autonomous vehicles and perception sensors. He has been involved in projects to create noise model for LiDAR sensor and also the cyber resilience of GNSS sensors for automotive use.

Valentina Donzella received her BSc (2003) and MSc (2005) in Electronics Engineering from University of Pisa and Sant’Anna School of Advanced Studies (Pisa, Italy), and her PhD (2010) in Innovative Technologies for Information, Communication and Perception Engineering from Sant’Anna School of Advanced Studies. In 2009, she was a visiting graduate student at McMaster University (Hamilton, ON, Canada) in the Engineering Physics department.

She is currently Associated Professor, in the Intelligent Vehicles group at WMG, University of Warwick, UK, and she has been awarded a Royal Academy of Engineering Industrial Fellowship on camera sensors; before this position, she was a MITACS and SiEPIC postdoctoral fellow at the University of British Columbia (Vancouver, BC, Canada), in the Silicon Photonics group. She is first author and co-author of several journal papers on top tier optics journals. Her research interests are: LiDAR, Intelligent Vehicles, integrated optical sensors, sensor fusion, and silicon photonics.

Dr Donzella is Full College member of EPSRC and Senior Fellow of Higher Education Academy.