Introducing Noise for AirSim's 3D Lidar Sensor to Reduce the Sim2real Gap

David Tejero[1], Marco A. Montes-Grova[1], José A. Álvarez[1], Francisco J. Pérez-Grau[1] and Antidio Viguria[1]

***Abstract*— In robotics, it is important to model sensor noise because it can affect the accuracy and reliability of the robot's perception of its environment. Modeling sensor noise also allows for more accurate simulations of robotic systems, which can help improve their performance in real-world scenarios. Given the rise in the use of simulation tools for rapid prototyping and iteration of aerial robotic systems, we propose the introduction of a noise model for the 3D lidar sensor that is supported in AirSim, in order to help the community build more accurate, reliable, and cost-effective solutions.**

I. INTRODUCTION

Autonomous robots need to sense the world around them. Sensor noise can cause measurement errors, which could lead to incorrect decisions and actions by the robotic systems [1]. Modeling sensor noise is important in robotics because it helps improve robot sensors' accuracy and ability to perceive the environment. Sensor noise can be caused by various factors, such as environmental conditions, manufacturing imperfections, hardware limitations, and signal processing errors. Some sensors that rely on measuring distances, such as sonar, infrared, and lidar sensors, are known to be particularly susceptible to noise [2].

Modern robotic systems are complex and must be tested in simulations with detailed sensor noise models to verify robotic behaviour effectively. Ignoring sensor noise in simulations can lead to unrealistic performance expectations and poor design choices. Using realistic noise models enables the development of more accurate simulations, which can improve the performance of robotic systems in real-world scenarios. The pitfalls of naive robot simulations have been recognized in areas such as evolutionary robotics [3], suggesting that carefully validated simulations can provide a useful tool for testing hypotheses about the behaviour of robots in complex environments. Implementing sensor noise in robotics simulations poses several challenges; some of the most important aspects are accurately simulating the physical world, which involves a composition of various models. To address these challenges, researchers continue to develop new methods and models to improve the accuracy and reliability of robotics simulations [4].

Simulation is a critical tool for aerial robotics research. It allows researchers to test ideas safely, predict system behaviour, revisit concepts, and tune algorithms to sufficient levels, given that good models have been derived. Moreover, the complexity of aerial robotic systems compared to ground robots and the safety implications of current regulations demand simulation tools in order to test developments as early as possible. There are several options available for robotics research [5], such as Gazebo within the Robot Operating System (ROS) [6], Carla which is more focused on autonomous driving [7], or AirSim developed by Microsoft [8]. Nevertheless, using simulation in aerial robotics research has important limitations [9] since real-world dynamics are very difficult to model accurately.

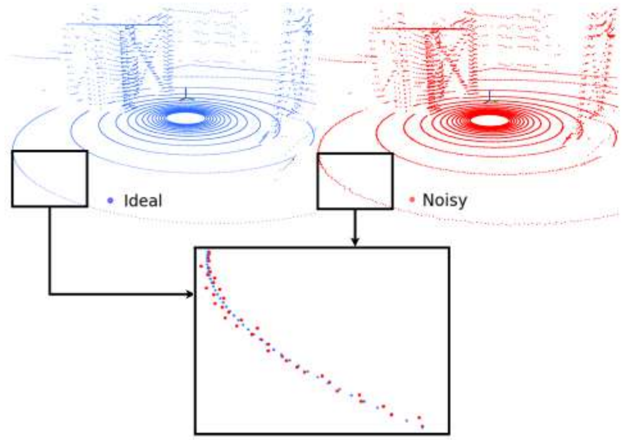

Fig. 1. Simulated lidar point clouds with our proposed noise model

In order to improve the performance of the algorithms when they are transferred from simulation to real robotic systems, we have identified a potential improvement for the sensors supported in AirSim. Currently, the only ones providing a noise model are the barometer and magnetometer. Lidar technology is increasingly used in aerial robotics research, given the reduction in cost, size and weight of available commercial models, which has allowed their integration into a wider range of aerial platforms. This has brought more interest in research based on this technology, which implies a higher importance in how these systems are used in simulation environments. The 3D lidar sensor supported in AirSim allows the configuration of several parameters, but none related to a noise model for the provided measurements. We propose introducing a noise model for this sensor corresponding to the specifications of current commercial products, as depicted in Fig. 1, showing results of experiments where such a model brings the simulator closer to a real-world scenario.

The rest of the manuscript is structured as follows. Section II details our AirSim framework and the scenario we have worked with. Section III explains the noise model developed

[1]CATEC (Advanced Center for Aerospace Technologies), Calle Wilbur y Orville Wright, 19, La Rinconada, 41309 Sevilla, Spain; [dtejero, mmontes, jaalvarez, fjperez, aviguria]@catec.aero

in this context, while Section IV shows some results of its impact in a practical application. Finally, Section V summarises our approach's outcomes and future potential.

II. SIMULATION ENVIRONMENT

AirSim is an open-source, cross-platform simulator designed for autonomous systems research. It is built on Unreal Engine [10], a 3D computer graphics game engine developed by Epic Games. The game engine does all the graphical rendering, collision, and vehicle movement simulation. AirSim supports software-in-the-loop simulation with popular flight controllers such as PX4 or ArduPilot, which is very convenient for testing autonomous missions before the deployment on the actual hardware platform. Moreover, it is easily integrated with ROS through a wrapper, allowing external nodes to work with the simulated data.

In order to simplify the configuration, installation of dependencies and deployment of this environment on any computer, our framework is based on Docker images. This allows for automating the deployment of applications within software containers, providing an additional layer of abstraction and automation of application virtualization in multiple operating systems. We have also developed scripts to deploy and automatically configure different parameters concerning the simulation and the involved onboard sensors.

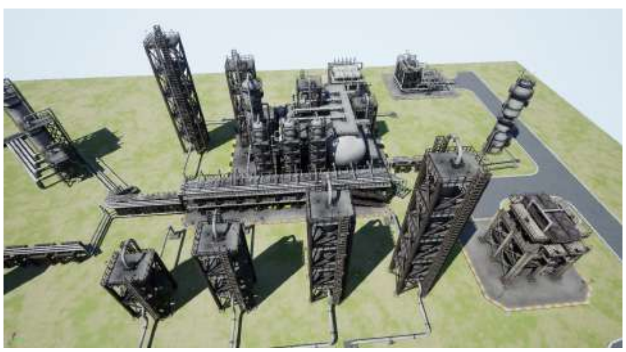

Fig. 2. Refinery environment used in the experiments

Furthermore, to demonstrate the validity of our contribution, we have created a realistic scenario for inspection and maintenance purposes, in this case a refinery environment. Oil and gas production plants frequently experience component deterioration due to environmental exposure, or materials used in production. If pipe corrosion is left unchecked, it can result in accidents, such as devastating explosions and the release of hazardous materials. Consequently, this can affect the safety, environment, and operability of the plant. Aerial robots are a very useful tool for inspection purposes in these plants. To ensure their safe operation, it is essential to have a robust localization system independent of the Global Navigation Satellite System (GNSS) combined with onboard inertial sensors, which can be unreliable in such a cluttered environment full of metallic structures. The proposed virtual world allows evaluation of the performance of algorithms in a complex scenario, where the robot localization needs to be as good as possible. Fig 2 shows an aerial view of the recreated refinery scenario used in this work.

III. LIDAR NOISE MODEL

The implemented noise model is closely related to how a 3D lidar sensor internally works. Most commercial lidars are formed by several vertically arranged laser beams rotating at high speed. The horizontal and vertical angle resolution can be known with high precision, so the directions of the laser beams can be measured with low error. Nevertheless, the range measurements depend on the beams' time of flight (ToF), which is more susceptible to errors due to environmental conditions or the internal clock resolution. Moreover, these errors increase in more distant points, causing a worse performance as the beams hit more distant objects. Following this idea, instead of adding a random noise for each point, our model only affects the range and not the direction of the beam it belongs to. The range noise is modelled as a zero-mean Gaussian distribution with a standard deviation which increases linearly with the range [11].

Since AirSim is open-source, the code for simulating a lidar within the Unreal Engine is available, so this noise model has been added to each point in the ray-tracing process. The main parameters which can be modified externally are the standard deviation at zero distance and at the maximum range. A linear noise model has been implemented by adapting the AirSim plugin, generating the desired point clouds when the lidar sensors are parameterized, providing such standard deviation values. In contrast to other open-source simulators [7] where the deviation is constant with distance, the proposed noise model increases linearly.

Two commercial lidar models have been studied, the Ouster OS0 and OS1, with 32 horizontal scans, and each of the scans consists of 512 points. The main differences between these two sensors are the vertical Field Of View (FOV) and the maximum detection range. According to the datasheets [12][13], the selected OS0 sensor has been parameterized in our simulation with a 90º vertical FOV (± 45º) and a maximum range of 45m. On the other hand, the OS1 sensor was characterized by a 45º vertical FOV (± 22.5º) and a maximum range of 100m. The accuracy of both sensor models is 3cm for lambertian targets, while their precision is defined by fixed values for the mean and standard deviation for different range intervals, according to Table I.

TABLE I
PRECISION FOR OUSTER SENSORS (10% LAMBERTIAN REFLECTIVITY)

OS0		OS1	
Range [m]	StdDev. [cm]	Range [m]	StdDev. [cm]
0.3 - 1	2	0.3 - 1	0.7
1 - 10	1	1 - 20	1
10 - 15	1.5	20 - 50	2
15 - 45	5	50 - 100	5

Based on this data, the values for the standard deviation at minimum and maximum ranges were chosen as follows: 1-10cm for OS0 and 1-15cm for OS1. While the proposed

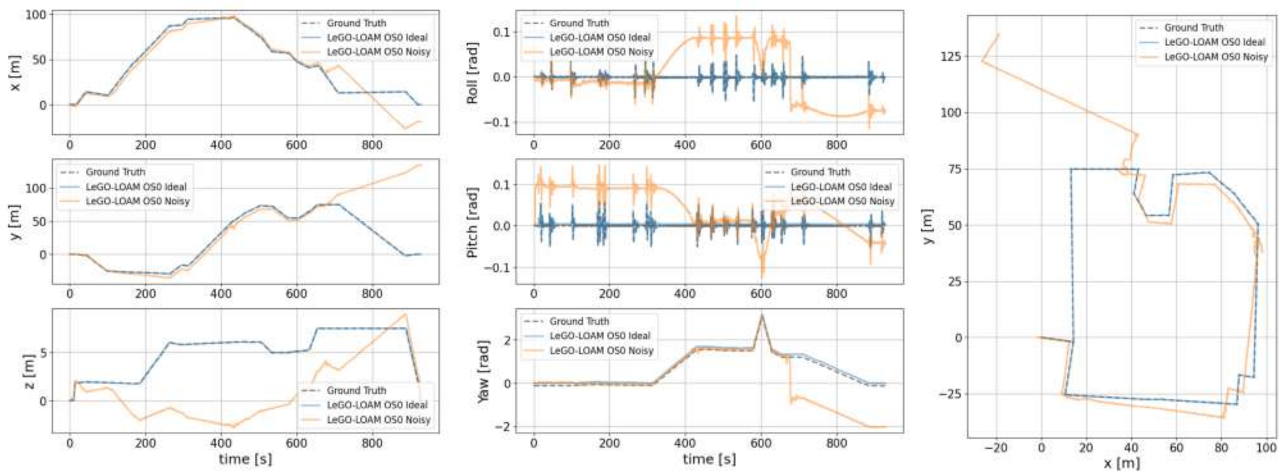

Fig. 3. Trajectories comparison between ideal and noisy OS0 ouster sensor

noise model appears to be basic, it reflects more accurately the noise data provided by the manufacturer, and better corresponds with our experience using such sensors in real experiments.

IV. EXPERIMENTAL RESULTS

An open-source localization algorithm was chosen to evaluate the influence of the introduced lidar noise. LeGO-LOAM [14] is a lightweight lidar odometry and mapping method that provides real-time six-degree-of-freedom pose estimations. It is specifically optimized for a horizontally placed 3D lidar sensor mounted on a ground vehicle, assuming there is always a ground plane in the scan. While these evaluations are based on an aerial vehicle, this assumption holds for our scenario with flat ground, even though LeGO-LOAM could properly handle variable terrain by definition. Only the lidar point clouds are included in the odometry computation, i.e. no inertial data are used. In this way, the odometry quality will strongly depend on the geometry of the point clouds, and the effect of the introduced noise can be clearly assessed.

A benchmark trajectory was defined to evaluate the localization of the aerial robot. A closed trajectory around the refinery environment was designed to emulate a real mission for the general inspection of the plant during a single flight, as depicted in Fig. 4. The odometry comparison (with and without sensor noise) is carried out by calculating the absolute pose error (APE), which evaluates the global consistency of the estimated trajectory by comparing the absolute distances between the estimations and the ground truth. The results have been obtained using *evo* [15].

First, the effect of adding noise to the Ouster OS0 sensor model is shown. As it can be seen in Fig. 5, the effect of the noise is considerable (note the difference in scale) which significantly increases the error in the position estimation.

While the localization using the ideal lidar resembles the real trajectory with high accuracy, the algorithm faces great difficulties if the proposed noise is introduced. The localization accumulates some drift as the aerial robot moves, especially in yaw and altitude changes. By the last quarter of the benchmark trajectory, there was a turn that caused high drift, completely deviating the estimation. From this moment, the APE error increases, as can be seen in Fig. 5.

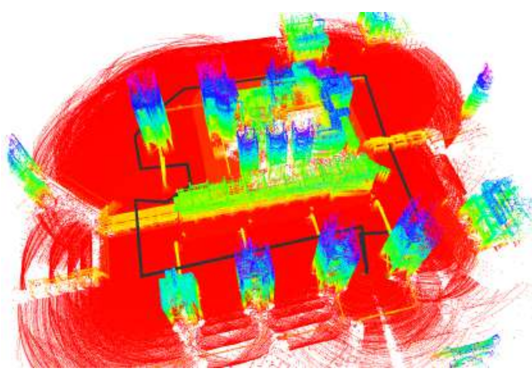

Fig. 4. Reconstructed 3D coloured map of the environment; the estimated trajectory is shown in black

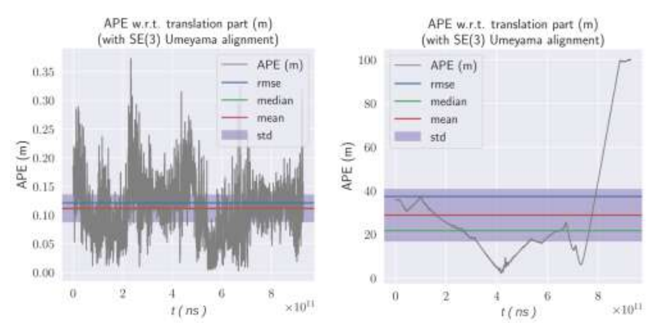

Fig. 5. APE obtained with ideal (left) and noisy (right) OS0 lidar

Using the OS1 sensor, as shown in Fig. 6, errors are relatively small with and without sensor noise over the entire trajectory. This is probably due to the fact that the OS1 sensor has twice the range of the OS0, and the modeled noise is spread over the entire range. Having a higher range is a differentiating factor, which improves the accuracy of odometry thanks to a more global perception of the scene. The OS1 lidar with the noise model causes much lower drifts compared with the previous noisy OS0.

In this case, both ideal and noisy models of OS1 resemble the benchmark trajectory with decent accuracy. Nevertheless, the effect of noise is not unnoticeable. APE errors for the odometry corresponding to the noisy OS1 are shakier and have higher values than those produced with the ideal sensor.

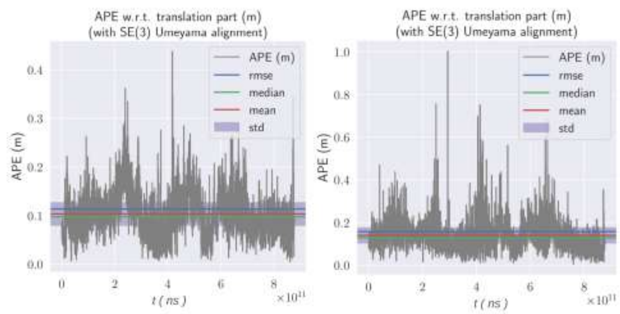

Fig. 6. APE obtained with ideal (left) and noisy (right) OS1 lidar

Table II compiles the errors obtained for all the simulated experiments. The ideal sensors have similar low values, causing the lidar-based odometry to perform accurately despite other conditions. However, when the noise model is introduced, both sensors perform worse. This proves the importance of modeling the noise to help reduce the gap between simulation and reality, since the specifications of the real sensor must be handled in the developed algorithms.

TABLE II

APE IN SQUARED TRAJECTORY FOR ALL CONFIGURATIONS

	OS0 Ideal	OS0 Noisy	OS1 Ideal	OS1 Noisy
RMSE	0.1213	37.3830	0.1137	0.1576
Mean	0.1118	28.8392	0.1034	0.1398
Median	0.1111	21.7313	0.0977	0.1289
Std	0.0471	23.7863	0.0472	0.0727
Min	0.0028	2.1846	0.0053	0.0042
Max	0.3728	100.1327	0.4382	1.0041

V. CONCLUSIONS

Modeling sensor noise in simulation environments helps reduce the cost and time required for the testing and development of aerial robotic systems. By introducing a lidar noise model using real specifications of currently available commercial products, the algorithms developed considering these measurements will be more robust and reliable, reducing the risk of failure in real-world scenarios. New improved commercial models can be easily integrated in order to update our contribution. Future work will consider exploring other noise models using, for example, a continuous piecewise linear function to adjust better to the datasheet values provided for the different range intervals. Nevertheless, assessing how the simulated noise adjusts to the real sensor behaviour remains an open question. Moreover, a loss function could also be implemented to mimic how some points are not correctly processed due to reflections, environmental conditions and sensor limitations. Furthermore, an identifier of the hit object could be retrieved for each point thanks to AirSim, so an even more realistic behaviour can be modelled by adapting the parameters to the nature of the objects materials, as well as simulating the vibrations of the aircraft or the effect of quick rotations on the sensor's echo.

ACKNOWLEDGMENTS

This work has been supported by the European Union's research and innovation programme within the context of PILOTING project (Horizon 2020 grant agreement No 871542) and BEEYONDERS project (Horizon programme grant agreement No 101058548).

REFERENCES

[1] Josh Bongard. Probabilistic robotics. sebastian thrun, wolfram burgard, and dieter fox. (2005, mit press.) 647 pages. *Artificial Life*, 14(2):227–229, 2008.

[2] M. Hebert. Active and passive range sensing for robotics. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 102–110 vol.1, 2000.

[3] Nick Jakobi, Phil Husbands, and Inman Harvey. "noise and the reality gap: The use of simulation in evolutionary robotics,". volume 929, pages 704–720, 01 1995.

[4] Kshitij Jerath, Sean Brennan, and Constantino Lagoa. Bridging the gap between sensor noise modeling and sensor characterization. *Measurements*, 116:350–366, February 2018.

[5] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

[6] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. volume 3, 01 2009.

[7] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[8] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *CoRR*, abs/1705.05065, 2017.

[9] HeeSun Choi, Cindy Crump, Christian Duriez, Asher Elmquist, Gregory Hager, David Han, Frank Hearl, Jessica Hodgins, Abhinandan Jain, Frederick Leve, Chen Li, Franziska Meier, Dan Negrut, Ludovic Righetti, Alberto Rodriguez, Jie Tan, and Jeff Trinkle. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences*, 118:e1907856118, 01 2021.

[10] Epic Games. Unreal engine 4.27. `https://www.unrealengine.com`, 2021.

[11] Johann Laconte, Simon-Pierre Deschênes, Mathieu Labussière, and François Pomerleau. Lidar measurement bias estimation via return waveform modelling in a context of 3d mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8100–8106, 2019.

[12] Ouster. Ouster os0 rev6 datasheet. `https://data.ouster.io/downloads/datasheets/datasheet-rev06-v2p2-os0.pdf`, 2021.

[13] Ouster. Ouster os1 rev6 datasheet. `https://data.ouster.io/downloads/datasheets/datasheet-rev06-v2p2-os1.pdf`, 2021.

[14] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

[15] Michael Grupp. evo: Python package for the evaluation of odometry and slam. `https://github.com/MichaelGrupp/evo`, 2017.