Quantitative Evaluation of SLAM Systems via Monte Carlo Simulation Framework

Alonso Llorente[1], Iulian Filip[1], Ricardo Vázquez Martín[2], Daniel Medina[1]

***Abstract*— This paper addresses the challenge of evaluating Simultaneous Localization and Mapping (SLAM) algorithms, highlighting the limitations of existing assessment tools that often lead to over-tuned and less generalized solutions. We introduce a novel Monte Carlo evaluation tool developed in the Gazebo simulation environment using the ROS framework. This tool enables empirical performance assessment of SLAM algorithms with statistical guarantees, simulating diverse scenarios, including sensor failures and model mismatches. The system, application and sensor agnostic, evaluates SLAM performance using noisy data from a 3D vehicle model equipped with a diverse sensor suite in a replicated real-world environment. For this study, we used the proposed methods to evaluate two LiDAR-Inertial SLAM algorithms, Fast-LIO 2 and LIO-SAM, proving its versatility and robustness in assessing SLAM performance.**

I. INTRODUCTION

As demand for efficiency rises, human involvement in tasks is decreasing, especially in transport, with growing interest in autonomous systems such as robots and ADAS. These systems require reliable navigation and environment-awareness solutions, making SLAM a key problem. SLAM estimates a robot's environment and its motion, but new challenges persist, including the integration of emerging sensors like LiDAR and event-based cameras, global localization via GNSS, and the demand for more stringent navigation in complex environments. Despite extensive SLAM research, no tool exists for thoroughly evaluating SLAM performance. Traditional methods compare SLAM outputs with post-processed ground truth from datasets such as KITTI or NuScenes, but these methods face limitations like ground truth accuracy issues, lack of quantitative map evaluation, high dataset costs, and overfitting to specific datasets.

To address this, we propose a Monte Carlo evaluation tool to assess SLAM solutions with statistical guarantees. Developed in Gazebo using ROS, the tool simulates diverse sensors and environments, testing SLAM algorithms under various conditions, including sensor failures and model mismatches.

We compare conventional methods with this new approach by evaluating two prominent SLAM algorithms, Fast-LIO 2 and LIO-SAM, highlighting strengths, weaknesses, and improvements in accuracy and robustness.

The contributions of this work include:

- A sensor-agnostic Monte Carlo simulation tool for evaluating SLAM performance with statistical rigor.

[1]Authors are with the Institute of Communications and Navigation of the German Aerospace Center (DLR), `name.surname@dlr.de`

[2]Assistant Professor of Systems Engineering and Automation at University of Malaga (UMA), `rvmartin@uma.es`

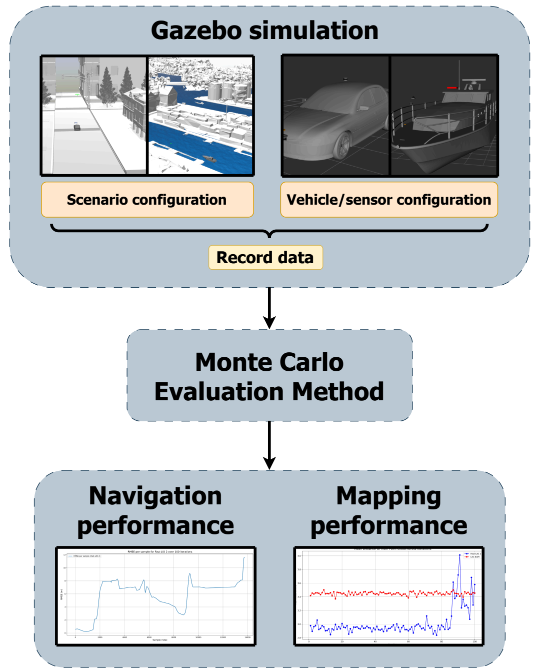

Fig. 1. General view of the Monte Carlo Simulation tool designed to analyze the performance of SLAM algorithms

- A comparative evaluation of SLAM algorithms, demonstrating the new method's advantages over conventional approaches.

II. RELATED WORK

SLAM (Simultaneous Localization and Mapping) is a fundamental problem in robotics, crucial for autonomous systems to understand and navigate their surroundings. Initially, LiDAR-based SLAM approaches focused on offline mapping using algorithms like Iterative Closest Point (ICP) [1] for aligning point clouds. Research then shifted towards real-time SLAM, exemplified by the development of the LOAM framework [2], which efficiently extracts edge and planar features from LiDAR data. Subsequent algorithms, such as LIO-SAM [3] and LeGO-LOAM [4], integrate additional sensors like IMUs to enhance accuracy and robustness. Direct methods, including FAST-LIO2 [5] and DLIO [6], operate directly on raw LiDAR data, offering higher accuracy at the cost of computational efficiency. These advancements have expanded the applicability of SLAM in various environments.

SLAM evaluation involves assessing trajectory accuracy, mapping performance, and computational power required.

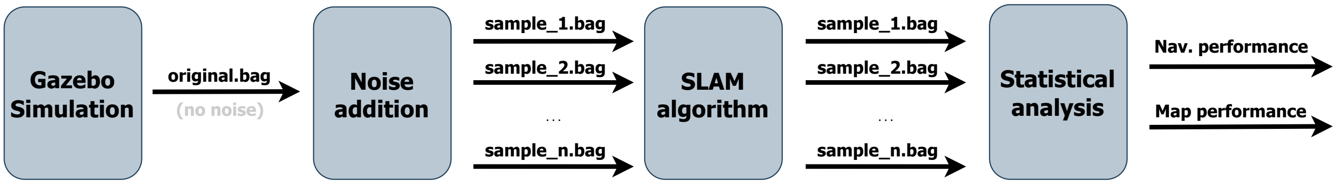

Fig. 2. Scheme for the Monte Carlo based evaluation.

Traditional metrics such as Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) measure how closely the estimated trajectory matches the ground truth, highlighting drift and local consistency. Mapping performance analysis, however, focuses on the quality and level of detail of the generated maps. Typically, this evaluation is performed using the ICP [1] algorithm or its variants. Computational time is usually assessed by recording the average time taken for the algorithm to perform specific tasks.

Benchmark datasets like KITTI, UrbanLoco [7], KAIST [8], and EuRoC [9] provide trajectory ground truth for evaluating SLAM algorithms' performance. This "ground truth" is generally the best trajectory estimate obtained through GPS-Inertial sensor fusion in offline post-processing. In practice, we cannot obtain the true trajectory when dealing with real sensor data. Analyzing mapping performance is even more challenging due to the lack of ground truth or reliance on post-processed SLAM-mapping solutions. Simulation environments can help address this by creating scenarios similar to real life, where there is complete knowledge and control over the position, orientation, and 3D mapping of the environment.

Simulation platforms like Gazebo, CARLA [10], AirSim [11], and Unity provide controlled settings for testing SLAM algorithms under various conditions, including changes in lighting, texture, and obstacle density, without the risks and costs of real-world testing. Gazebo, for instance, offers detailed physics-based simulations and supports a range of sensors, including LiDAR, making it a valuable tool for testing LiDAR SLAM in different scenarios. CARLA and AirSim, originally designed for autonomous driving research, have been adapted for SLAM evaluation, providing realistic urban and rural environments for testing and development. Additionally, new simulation frameworks are being developed specifically to address the unique challenges of SLAM. For example, Sim4CV [12] provides synthetic environments with ground truth data for benchmarking visual SLAM.

III. METHODOLOGY

To quantitatively evaluate SLAM systems, we propose a four-step process (see Figure 2). First, non-noisy data is recorded using a simulator. This data is then processed through a noise addition block, generating multiple noisy versions based on a specified noise model. These noisy samples are fed into the SLAM algorithm for testing. Finally, the SLAM results are statistically analyzed by comparing them to ground truth and point cloud data from the simulator.

A. Simulation environment

The simulation environment is created using Gazebo 11 with ROS Noetic and includes pre-configured environments, vehicles, and sensor setups. It could be adaptable to any transportation field, enabling testing across diverse scenarios such as automotive, inland waterways, and others, with vehicle and sensor configurations like LiDAR, stereo cameras, and GNSS.

B. Noise addition block

After recording non-noisy data, noise is iteratively added to simulate real-world conditions, assessing system robustness [13]. A Monte Carlo approach is used, with noise applied to LiDAR and IMU data, simulating conditions such as sensor inaccuracies.

1) IMU noise model: The model includes two types of noise: one that affects measurements instantly and another that simulates gradual drifts over time. This approach provides a realistic representation of IMU sensor noise, making it useful for testing and calibrating algorithms that process such data under realistic conditions.

2) LiDAR noise model: To simulate real-world factors like temperature fluctuations, surface reflectivity, and environmental conditions on LiDAR measurements, we use a simplified noise model. This model introduces variations to the LiDAR data, capturing key characteristics of measurement noise. It provides a practical way to test SLAM algorithms under realistic conditions without the complexity of detailed models.

C. SLAM algorithm block

After generating the noisy data, the SLAM algorithm is applied to each dataset to obtain navigation and mapping estimates. Proper tuning of the algorithms is essential to match the noise levels. Once tuned, the solutions are computed and stored for analysis, ensuring a comprehensive test of the SLAM system under various conditions and providing insights into its performance and robustness.

D. Statistical analysis block

After computing the SLAM algorithm results, a detailed statistical analysis is conducted using exact ground truth data from the simulation. This enables precise comparison of the SLAM outputs, focusing on metrics like positional accuracy, mapping precision, and result consistency across test scenarios. Leveraging this reference data ensures robust and reliable performance evaluation under various conditions.

1) Navigation analysis: The navigation analysis involves comparing the estimated trajectory from the SLAM algorithms with the ground truth trajectory. For each estimated point along the trajectory, the Root Mean Square Error (RMSE) is calculated to evaluate the accuracy of the estimated trajectory relative to the ground truth.

The RMSE is computed for each point i along the SLAM output from sample n: (x_i^n, y_i^n, z_i^n). These estimated points are then compared to the ground truth coordinates $(x_i^{gt}, y_i^{gt}, z_i^{gt})$ across all iterations.

2) Mapping analysis: Map evaluation is performed using the 3D environment point cloud from the Gazebo simulation. The process involves comparing the estimated point cloud against the map truth to assess drift. For each iteration, the mean distance between points in the estimated map and the map truth is calculated, using a k-nearest neighbor search-based technique [14]. This distance reflects the drift between the estimated and actual maps, and how this drift varies with different noise levels is analyzed.

IV. Experimental Set-up

This section provides an overview of the experimental setup used for data collection, which recordings will be further analyzed in the results section.

The simulation uses a model of the AURORA vessel, a real pleasure craft operated by the Multi Sensor Systems Group at DLR [15]. The environment represents Berlin's Westhafen Island, where AURORA's real measurements are taken. The simulated vessel is equipped with the same sensors as the actual one (see Figure 3).

Recordings were made in the simulator to replicate real-world data collection, with the environment designed to closely match the actual area. The ground truth path from the simulator is also shown in Figure 3.

Fig. 3. Comparison of Sensor Distribution and Trajectory: The upper-left image shows the sensor distribution on the real-world AURORA vessel, while the upper-right image illustrates the Westhafen area where is usually driven. The bottom-left image depicts the sensor configuration of the simulated AURORA vessel, and the bottom-right image shows the simulated trajectory, reflecting the real-world setup.

V. Results and Discussion

In this section, we evaluate the proposed method using the dataset from the experimental setup to assess two SLAM algorithms: Fast-LIO 2 and LIO-SAM. Both use a LiDAR-Inertial setup but differ in approach—Fast-LIO 2 uses filtering, while LIO-SAM employs smoothing. Fast-LIO 2 processes point cloud data directly, whereas LIO-SAM uses a feature-based approach. Comparing these algorithms on the same dataset will provide insights into their performance.

For the evaluation, we generated 100 noisy sample files from the original dataset to test the algorithms under controlled noise conditions.

A. Navigation performance

To illustrate the performance of the estimation across all samples, we plot the average of the 100 navigation solutions from both Fast-LIO 2 and LIO-SAM algorithms. These plots are overlaid with the RMSE for each sample at every estimation step. This is shown in Figure 4.

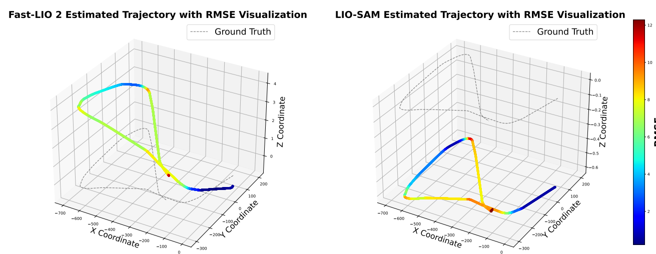

Fig. 4. Comparison of 3D average estimated paths overlapped with RMSE per sample in a colored scatter format using Fast-LIO 2 and LIO-SAM.

B. Mapping performance

The result of the mean distance between estimated maps along the iterations, can be seen in Figure 5.

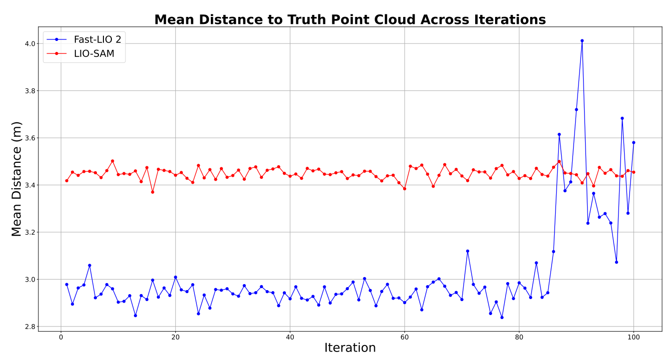

Fig. 5. Mean distances plot for each map estimate over 100 iterations for Fast-LIO 2 and LIO-SAM

The mean error for both algorithms is approximately 3 meters across all iterations. However, Fast-LIO 2 shows a higher standard deviation, likely due to more outliers in the later iterations. For a qualitative comparison, Figure 6 shows the drift in estimated maps from both algorithms compared to the map truth..

VI. Conclusion and Future Lines

In this manuscript, we introduced a Monte Carlo-based method for statistically evaluating SLAM solutions and as-

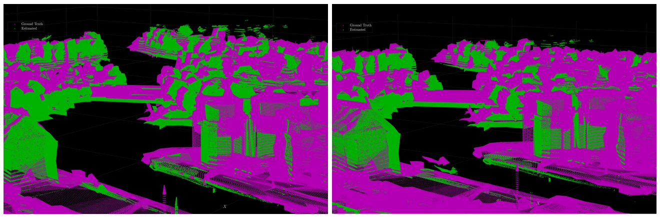

Fig. 6. Estimated map from sample 50 (green) using Fast-LIO 2 (left) and LIO-SAM (right) overlapped with the map truth (purple) from simulation, allowing to see this around 3 meters drift.

sessed Fast-LIO 2 and LIO-SAM. Our approach offers improved generalizability and accuracy over traditional methods, aiming to lower evaluation time and costs.

Future work will enhance the simulation by incorporating various weather conditions and sensor constraints, such as fog and rain, to introduce more complex noise models. We will also expand the simulation to include diverse transportation scenarios, increasing its versatility and depth.

References

[1] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[2] J. Zhang and S. Singh, "Loam : Lidar odometry and mapping in real-time," *Robotics: Science and Systems Conference (RSS)*, pp. 109–111, 01 2014.

[3] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[4] T. Shan and B. 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)*. IEEE, 2018, pp. 4758–4765.

[5] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: fast direct lidar-inertial odometry," *CoRR*, vol. abs/2107.06829, 2021.

[6] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3983–3989, 2023.

[7] W. Wen, Y. Zhou, G. Zhang, S. Fahandezh-Saadi, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, "Urbanloco: A full sensor suite dataset for mapping and localization in urban scenes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2310–2316.

[8] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[9] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[11] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.

[12] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, pp. 902–919, 2018.

[13] A. Llorente, "Monte carlo simulation tool to assess slam performance," Ph.D. dissertation, 06 2024.

[14] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[15] L. Hösch, A. Llorente, X. An, J. P. Llerena, and D. Medina, "High definition mapping for inland waterways: Techniques, challenges and prospects," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 6034–6041.