Comparative Performance Analysis of LiDAR-Based SLAM Algorithms: A Case Study

Krithiga Ramesh*†, Maxim Friesen*, Tullio Facchinetti†, and Lukasz Wisniewski*
*TH-OWL - inIT, Germany †University of Pavia, Italy
krithiga.ramesh@th-owl.de, maxim.friesen@th-owl.de,
tullio.facchinetti@unipv.it, lukasz.wisniewski@th-owl.de

2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA) | 979-8-3503-6123-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ETFA61755.2024.10710946

***Abstract*—Mobile robots are essential in various industries, with Simultaneous Localization and Mapping (SLAM) technology playing a crucial role in their autonomy. This work-in-progress paper lays the foundation for evaluating 2D LiDAR-based SLAM algorithms for implementation on a Clearpath Jackal robot in a smart factory environment. The study focuses on three SLAM algorithms: GMapping, Cartographer, and Hector SLAM. A real-world smart factory, serving as a case-study location, is modelled in the Gazebo simulator to evaluate the selected algorithms according to mapping quality, location accuracy, and performance consistency. The simulation uses a hardware-in-the-loop approach, where LiDAR data is processed by the physical Jackal robot, ensuring realistic testing conditions. The findings from the simulation, including the key factors influencing the performance metrics, are validated through real-world testing. This paper outlines the methodology for both simulation and real-world deployment, setting the stage for determining the most suitable SLAM algorithm for efficient and accurate mapping and localization within the operational constraints and requirements of a smart factory environment. Additionally, preliminary insights into factors affecting SLAM performance in the real-world and the relative strengths and weaknesses of each framework are discussed.**

***Index Terms*—SLAM, LiDAR, Smart Factory**

I. Introduction

Mobile robots have become indispensable in automation-based industries, seamlessly integrating with various production processes. Their development heavily relies on Simultaneous Localization and Mapping (SLAM) technology [1]. Given the diversity of applications and the highly dynamic nature of modern factory environments, SLAM algorithms may perform differently, requiring their comparison and evaluation to identify the most suitable one for specific tasks. Factors such as localization accuracy, mapping quality, computational efficiency, and robustness to environmental changes are crucial performance metrics that vary across algorithms [2].

Therefore, in this work-in-progress paper, we aim to analyze different LiDAR-based SLAM algorithms suitable for implementation on a Clearpath Jackal robot within the SmartFactoryOWL [3], a living lab with 2000 square meters shop floor. Specifically, we focus on evaluating the GMapping [4], Cartographer [5], and Hector SLAM [6] algorithms. The evaluation is conducted in a hardware-in-the-loop (HiL) simulation environment modelled in the Gazebo simulator, followed by real-world deployment to assess specific use cases based on selected key performance indicators (KPIs).

This leads to the following research objectives:

1) Identify the key differences between GMapping, Cartographer, and Hector SLAM algorithms, highlighting their unique characteristics.
2) Evaluate the performance of these algorithms within a HiL simulation model based on selected KPIs to assess localization accuracy, mapping quality, and computational efficiency, and determine which factors most significantly impact performance.
3) Validate the simulation results through real-world integration and testing in the SmartFactoryOWL environment.

The remaining paper is organized as follows: Section II reviews the related work pertinent to this research. Section III provides an overview of the problem, detailing the specific challenges and requirements addressed in this study. Subsequent sections cover the evaluation setup, results, and discussion, followed by the conclusions and future work.

II. Related work

SLAM technology can be broadly classified into LiDAR SLAM and Visual SLAM, based on the sensors utilized for environmental assessment [7]. Visual SLAM employs various cameras to construct a map and determine the camera's position within it [2]. Despite significant advancements in image processing and AI enhancing its functionality [8], LiDAR SLAM is preferred for its reliability in diverse environments, particularly in low-light or no-light conditions where Visual SLAM struggles [9]. LiDAR-based systems are resilient to environmental changes like varying light conditions and shadows, making them ideal for dynamic applications [10]. State-of-the-art algorithms such as LeGO-LOAM and LOAM demonstrate remarkable accuracy and efficiency in real-time tasks [11].

For this study, we focus on LiDAR-SLAM algorithms that do not rely on odometry, selecting GMapping, Cartographer, and Hector SLAM as viable options. GMapping is effective for indoor environments but may face challenges in highly dynamic surroundings, requiring frequent map updates [4],

This work received funding from the Federal Ministry for Economic Affairs and Climate Protection (BMWK) and the Association of Industrial Research Associations (AiF eV) under grant agreement No 01IF23171N.

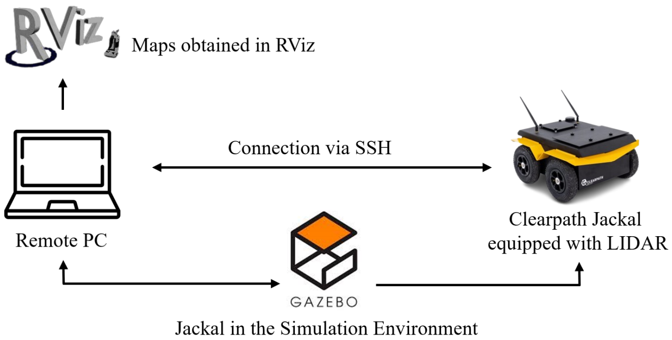

Fig. 1. Hardware-in-the-loop evaluation setup.

[12]. Hector SLAM is suitable for high-speed mapping applications and operates effectively without odometry but may underperform in large-scale environments [6]. Cartographer excels in large-scale environments and complex indoor navigation, reducing drift and enhancing map consistency, though it demands powerful hardware for real-time processing [5], [13]. Several comparative studies have evaluated SLAM algorithms on criteria like accuracy, computational efficiency, and scalability. Huang et al. [14] compared GMapping, Cartographer, and Hector SLAM using metrics like mapping accuracy, localization error, and computational time. Zou et al. [15] assessed five 2D LiDAR-SLAM algorithms based on orientation accuracy and localization error, but not under dynamic conditions. De Carvalho et al. [16] compared processing time, simulation time, and the number of published messages for 3D-LiDAR-based SLAM algorithms in Gazebo. Filipenko and Afanasyev [17] focused on map quality and computational requirements for 2D SLAM algorithms.

Despite these evaluations, there is a need for a comprehensive analysis tailored to the specific requirements of smart factory environments, which involve dynamic and complex operational scenarios. This study aims to fill this gap by providing a detailed comparison of SLAM algorithms in both simulated and real-world smart factory contexts.

III. SLAM in a Smart Factory

Accurate positioning is crucial for various smart factory applications, including automated material transport, inventory management, equipment maintenance, and quality assurance processes. Mobile robots in smart factories must navigate dynamic layouts, adapt to varying lighting conditions, and interact seamlessly with other machinery. These tasks require high levels of positional accuracy to ensure operational efficiency and safety.

A. Case Study Example

In our case study, we aim to implement a mobile autonomous network diagnostic platform that measures 5G network parameters across the smart factory shop floor. The setup, wirelessly connected to a 5G access point, is mounted to a Clearpath Jackal robot that drives to various locations within the factory to perform end-to-end 5G network measurements. These measurements are mapped to their respective locations, creating a dynamic heat map of the selected network evaluation metrics. The Jackal either moves to predefined spots or executes measurements autonomously by selecting random locations within a predefined measurement area. The successful implementation of this project relies heavily on accurate maps of the factory environment, which the robot uses to navigate from point to point. Understanding the dynamic and unpredictable environment is crucial, and this study seeks to identify the most suitable SLAM algorithm for this context.

B. SLAM Algorithms under Evaluation

The SLAM algorithms considered for this research are briefly elaborated as follows:

- **GMapping**, short for Grid-based FastSLAM, is a probabilistic SLAM algorithm that constructs grid maps of the environment while estimating the robot's trajectory [4]. It uses the FastSLAM framework, which combines particle filters with Rao-Blackwellized particle filters (RBPF) to handle uncertainties.
- **Cartographer**, developed by Google, uses the Extended Kalman Filter (EKF) for creating 2D and 3D maps using LiDAR and IMU sensor data [5].
- **Hector SLAM**, employs an occupancy grid mapping approach and scan-matching techniques to generate accurate maps while estimating the robot's real-time position. It uses the Extended Kalman Filter (EKF) to integrate sensor measurements with a motion model [6].

IV. Evaluation Setup

The evaluation setup is designed as a hardware-in-the-loop approach as depicted in Fig. 1. Relevant sections of the smart factory under study are modeled in the Gazebo simulator running on a remote PC. The simulation environment interfaces with the physical Jackal via an SSH tunnel and emulates the smart factory sections, including obstacles and corresponding LiDAR data. The selected SLAM algorithms are executed on the Jackal's Robot Operating System (ROS). The Jackal publishes the resulting positioning data to its respective ROS topics, to which the remote PC is subscribed. The captured location and LiDAR telemetry from the Jackal are then visualized in RViz, a 3D visualization tool that displays sensor data and state information based on the current position of the robot.

1) Hardware Setup: The hardware setup involves utilizing the Clearpath Jackal robot, equipped with a SICK LMS111-101000 2D LiDAR sensor. The remote PC, hosting Gazebo, RViz and providing SSH connectivity to the Jackal is configured on Ubuntu 20.04 LTS.

2) Software Setup: The software setup primarily revolves around the ROS2 framework, specifically the Foxy Fitzroy release. ROS is an open-source middleware platform widely used in robotics research and development [18], with various versions available, each offering distinct features and improvements over its predecessors. While ROS1 offers stability and extensive documentation, making it a reliable choice for traditional robotics applications, ROS2 provides several significant advantages. ROS2 uses the Data Distribution Service for real-time communication, offering more reliable and scalable performance. It includes built-in security features, such as authentication and encryption, which are critical for industrial applications.

Moreover, the Nav2 stack, included in ROS2 represents a significant advancement in robotic navigation [19]. It comprises state-of-the-art path planning algorithms, improved dynamic obstacle avoidance capabilities, and a highly modular framework for customization and extension. These advantages make ROS2 and Nav2 the preferred choice for this study.

A. Definition of the Environment

The SmartFactoryOWL in Lemgo is a complex environment featuring various dynamic modular workstations and a primary transportation zone. It also serves as a collaborative space for employees and students. Considering this, we simplified the environment and divided the area of the smart factory into several sections, as shown in Fig. 2:

1) Section A - Modular working area (in Blue)
2) Section B - Office space for employees (in Yellow)
3) Section C - Fixed huge machinery (in Red)
4) Section D - Primary transportation area (in Green)

The shaded area in black represents a space where the robot cannot move, mostly occupied by huge machinery. Dividing the area allows us to obtain a high-resolution map of each section.

Only Section A has been modelled in the Gazebo simulation environment for initial testing. This controlled setting allows us to generate a map in simulation and compare it with the real-world counterpart to identify factors affecting performance. Key elements of the factory, such as walls and machinery, have been represented as simplified blocks within Gazebo. While detailed 3D modelling and replication are planned for future work, the current testing phase focuses on accurately representing the dimensions of the modular workstations as simple blocks in the Gazebo environment.

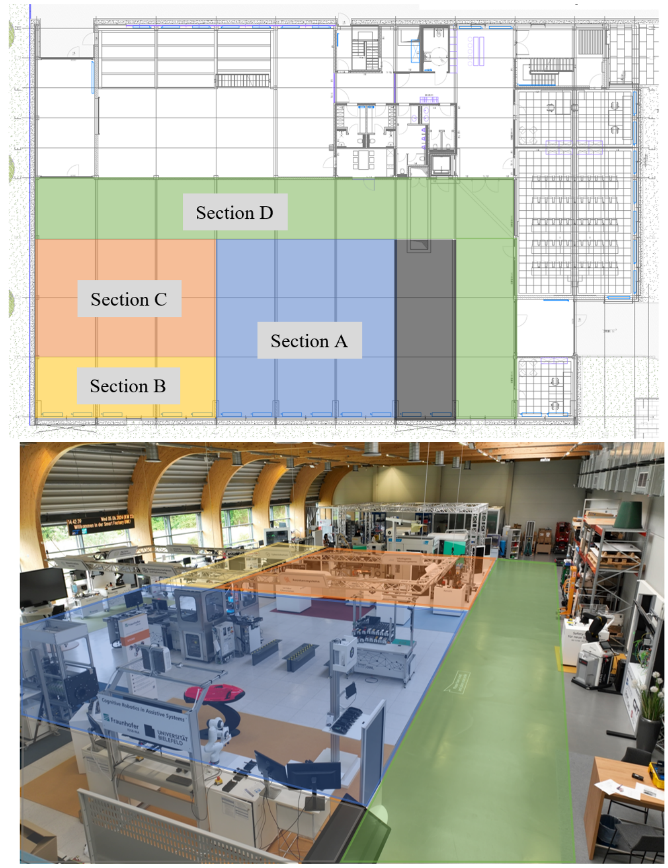

Fig. 2. SmartFactoryOWL and its selected sections under study

B. Metrics for comparison

To effectively compare the three algorithms in the defined environment, we have identified three primary metrics: mapping quality, location accuracy, and consistency.

- **Mapping Quality** refers to the completeness of the generated maps, crucial for representing the environment effectively. One technique involves the 'map resolution' or grid cell size of the occupancy grid map generated by the algorithm. A smaller grid cell size indicates a higher resolution and potentially higher mapping quality [12].
- **Location Accuracy** measures the precision of the positioning data provided by the algorithms. High location accuracy ensures that the estimated positions closely match the actual positions within the smart factory. One approach to estimate location accuracy is to compare the estimated positions with ground truth positions and compute the absolute error [12].
- **Consistency** assesses the reliability of the algorithms over time. An algorithm with higher consistency will produce similar results under the same conditions, indicating robustness and reliability. The goal is to conduct repeated trials of the mapping process under identical conditions and compare the generated maps. Statistical measures like the correlation coefficient can be calculated to quantify consistency [20].

V. Initial Work

For the initial steps, Section A was modeled in Gazebo, as depicted in Fig. 3, with the GMapping algorithm implemented. Commands were given to the robot to move through the environment, aiming to cover most of the area. Simultaneously, the LiDAR data from the robot was used to generate a map in RViz. However, as seen in Fig. 4, the GMapping algorithm produced an unsatisfactory map with unclear obstacles.

The poor map quality generated by GMapping can be attributed to several factors. Sensor noise and inaccuracies from the 2D LiDAR degrade map quality. Additionally, poor data associations and incorrect loop closures can result in inaccurate estimation of the robot's position. To address these issues, parameter tuning within the GMapping algorithm is necessary.

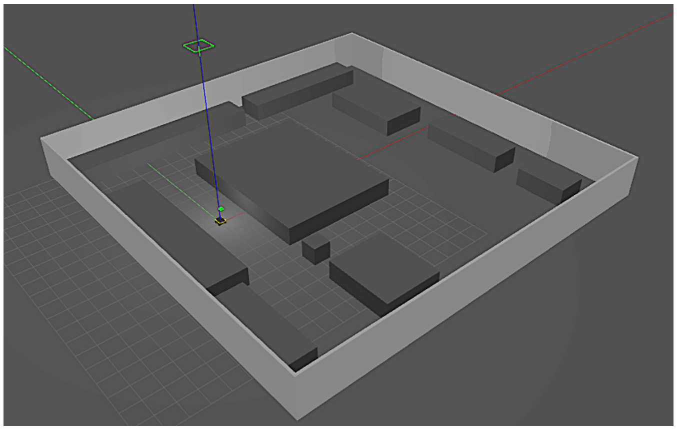

Fig. 3. Gazebo environment of Section A

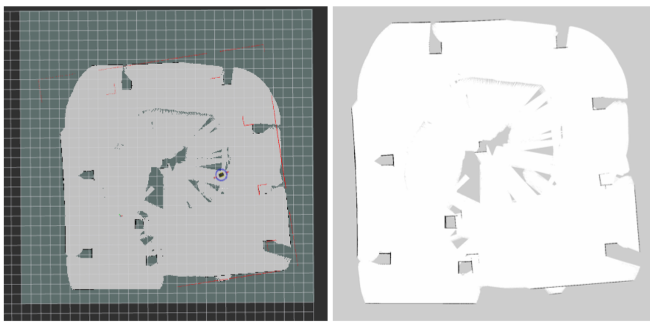

Fig. 4. Maps obtained from Gazebo for GMapping

Adjusting particle filter parameters and map resolution can help balance computational load and map accuracy, ensuring higher quality mapping. Further analysis will be conducted to identify the exact factors affecting mapping performance, with consideration of alternative algorithms for future work.

VI. Conclusion & Outlook

This study has focused on the comparative performance analysis of the LiDAR-based SLAM algorithms GMapping, Cartographer, and Hector SLAM, within the context of a smart factory environment. Utilizing a hardware-in-the-loop approach, preliminary work in the Gazebo simulation environment provided valuable insights into the required evaluation setup for conducting the extended analysis, including modeling the smart factory environment and interfacing with the Jackal robot.

The next steps involve several key phases to ensure comprehensive evaluation and validation of the SLAM algorithms. First, we will accurately model the remaining sections of the smart factory in the Gazebo simulator, ensuring that all relevant details are captured. We will then quantify and concretely specify the three chosen KPIs: mapping quality, location accuracy, and consistency. Following this, we will integrate the three selected SLAM algorithms within the Clearpath Jackal robot's ROS2. Each algorithm will be executed within the different modeled sections in Gazebo. We will compare their performance according to the specified KPIs, identifying performance differences and their root causes. Subsequently, to validate our findings, we will repeat the tests in one of the sections in the real-world smart factory environment. This phase will include incorporating multiple dynamic obstacles to evaluate algorithm performance under realistic conditions. Data collected during these evaluations will help us understand how environmental dynamics influence map generation and identify key factors that affect SLAM performance. By comparing metrics obtained from both simulation and real-world testing, we aim to provide valuable insights and practical solutions for optimizing related industrial automation use cases within smart factories.

References

[1] A. A. B. Pritsker, *Introduction to simulation and SLAM II (2nd ed.)*. USA: Halsted Press, Aug. 1984.

[2] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[3] SmartFactoryOWL, "Smartfactoryowl: The industry 4.0 real lab in ostwestfalen-lippe," 2024, accessed: 2024-06-08. [Online]. Available: https://smartfactory-owl.de/?lang=en

[4] G. Grisetti *et al.*, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Rob.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[5] W. Hess *et al.*, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2016, pp. 1271–1278.

[6] S. Kohlbrecher *et al.*, "A flexible and scalable SLAM system with full 3D motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, Nov. 2011, pp. 155–160.

[7] S. Thrun *et al.*, "Simultaneous localization and mapping," *Springer Handbook of Robotics*, pp. 871–892, 2008.

[8] X. Hou *et al.*, "Visual slam: why filter?" *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 421–437, 2013.

[9] D. Fox *et al.*, "A probabilistic approach to collaborative multi-robot localization," *Auton. Robots*, vol. 8, no. 3, pp. 325–344, 2000.

[10] K. Konolige *et al.*, "Large-Scale visual odometry for rough terrain," in *Robotics Research*. Springer Berlin Heidelberg, 2011, pp. 201–212.

[11] T. Zhang *et al.*, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2092–2099, 2019.

[12] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, Mar. 2002.

[13] M. Quigley, "ROS: an open-source robot operating system," *IEEE Int. Conf. Robot. Autom.*, 2009.

[14] S. Huang *et al.*, "Evaluation of robotic mapping algorithms in ros-based simulation environment," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 132–139.

[15] Q. Zou *et al.*, "A comparative analysis of LiDAR SLAM-Based indoor navigation for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6907–6921, Jul. 2022.

[16] M. E. C. de Carvalho *et al.*, "Comparative analysis of LiDAR SLAM techniques in simulated environments in ROS gazebo," in *Synergetic Cooperation between Robots and Humans*. Springer Nature Switzerland, 2024, pp. 275–285.

[17] M. Filipenko *et al.*, "Comparison of various SLAM systems for mobile robot in an indoor environment," *2018 International Conference on Intelligent Systems (IS)*, pp. 400–407, Sep. 2018.

[18] Open Source Robotics Foundation, "Ros (robot operating system)," 2023, accessed: 2024-06-07. [Online]. Available: https://www.ros.org/

[19] S. Macenski *et al.*, "Robot operating system 2: Design, architecture, and uses in the wild," *Sci Robot*, vol. 7, no. 66, p. eabm6074, May 2022.

[20] M. Gong *et al.*, "Comparing image similarity measures for land-cover classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 283–287, 2015.