Human-Machine Interface for Real-Time Interaction Focused on LiDAR SLAM Feature Extraction

Natalia Prieto-Fernández
Department of Electric, Systems and Automatics Engineering
Universidad de León
León, Spain
email: nprif@unileon.es

Sergio Fernández-Blanco
Universidad de León
León, Spain
email: sfernb03@estudiantes.unileon.es

Álvaro Fernández-Blanco
Universidad Carlos III de Madrid
Madrid, Spain
email: 100505928@alumnos.uc3m.es

José Alberto Benítez-Andrades
Department of Electric, Systems and Automatics Engineering
Universidad de León
León, Spain
email: jbena@unileon.es

Francisco Carro-De-Lorenzo
TRESCA Ingeniería S.A.
León, Spain
email: fcarro@tresca.es

Carmen Benavides
Department of Electric, Systems and Automatics Engineering
Universidad de León
León, Spain
email: carmen.benavides@unileon.es

***Abstract*—The development of Human-Machine Interfaces (HMIs) capable of visualising real-time processed data from robotic systems is of essential importance today. These graphical user interfaces play a crucial role in the smooth and efficient interaction between humans and robots, resulting in a significant improvement in the productivity and safety of various industrial and scientific applications. In this context, the present work addresses the design and development of an innovative human-machine interface for real-time 2D Light Detection and Ranging (LiDAR) based Simultaneous Localization and Mapping (SLAM). Such interface has been implemented in MATLAB and establishes an effective communication with the Robot Operating System (ROS) environment. Its straightforward design, planned for real-time teleoperation, facilitates the acquisition and analysis of data from sensors and actuators of a robotic system. The real-time processing of the data supports computationally expensive tasks, such as those associated with SLAM processing. Indeed, feature extraction for environment mapping from 2D LiDAR data is one of its applications.**

Keywords-Human-Machine Interface (HMI); Light Detection and Ranging (LiDAR); Simultaneous Localization and Mapping (SLAM); teleoperation.

I. INTRODUCTION

The development of Human-Machine Interfaces (HMIs) capable of visualising real-time processed data from robotic systems represents a major milestone in the evolution of automation and robotics. Advances in these fields have led to the creation of autonomous robots capable of navigating complex environments, making their own sensor-based decisions and completing specific missions without the need for constant human supervision. The level of autonomy of a robot is intrinsically related to the level of human intervention [1]. Even robots with a high degree of autonomy may require human guidance in dangerous or unknown situations, where human criterion and expertise cannot yet be replaced. It is in the face of critical events [2][3][4], such as remote medical care, disaster response and inspection of hazardous infrastructure, where the development of graphical user interfaces for teleoperation becomes essential.

HMI allows human operators to interact with the robot in a seamless way, as well as to monitor and control the robotic system effectively [5]. Moreover, these interfaces can contribute to the early detection and resolution of problems or challenges that may arise during robotic operation, resulting in increased efficiency and safety levels [6]. Furthermore, they are particularly valuable in environments where rapid and accurate decision-making requires fast processing of large volumes of data. This environmental information stems from the sensors available to each robotic system. Most robotic platforms can be programmed in Robot Operating System (ROS) environment, which may be complex for users without prior experience in robotics. In this sense, the implementation of robotic systems that use technologies such as Light Detection and Ranging (LiDAR) based Simultaneous Localization and Mapping (SLAM) to capture data from the environment, or communication through ROS, has generated a growing demand for graphical user interfaces capable of managing and visualising this data in real time.

Our work focuses on the design and development of an HMI capable of controlling a LiDAR SLAM robotic system in real time while ensuring seamless communication between ROS and MATLAB. The graphical user interface we propose is simple and intuitive for the user compacting the visualisation of the raw data from sensors, the data processed in ROS as well as the data processed in MATLAB. It is accessible from any PC and, most importantly, enables the visualisation of signals in real time. This paper is organized as follows: Section II provides a review of related work. In Section III, the robotic system is described. SLAM methodology is defined in Section IV. Section V contains the details of our HMI. In Section VI, HMI experiments are presented. Finally, conclusions are to be found in Section VII.

 ISBN: 978-1-68558-163-3

II. Related Work

HMIs are more complex than mere control panels. They provide a virtual window through which operators can view and interact with the environment. These interfaces must be intuitive and efficient, as any delay or confusion in communication could result in major consequences.

The healthcare sector serves as an example of a highly demanding environment. Surgeons can teleoperate surgical robots as long as they have access to acceptable real-time image and data. Advances in HMI technology significantly improve the accuracy and safety of medical procedures [3][7][8].

The exploration of hostile environments is another area where the teleoperation of robotic systems simplifies human tasks. Many robots are used in search and rescue (SAR) missions [9][10]. In such situations, robots can perform tasks that are dangerous or inaccessible to human operators.

In the industrial domain [11][12], LiDAR or camera-equipped robot teleoperation for tasks such as mapping and inspection benefits greatly from data processing and analysis capabilities. This facilitates decision-making and monitoring of operations in real time, improving efficiency and safety in production environments.

The ability to process and visualise data in real time is key to the success of teleoperation. In this context, the integration between ROS and different programming environments, such as MATLAB, plays a crucial role. ROS provides a solid platform for robot control and perception, while MATLAB offers powerful signal processing and data analysis capabilities. The effectiveness of the combination of both systems has been demonstrated in several case studies [13][14][15]. Indeed, our HMI proposal integrates ROS and MATLAB. The objective of this work is the development of bidirectional communication between the robotic system and the human operator, featuring real-time processing of sensor data and movement commands. Unlike other graphical interfaces, the proposed HMI focuses on the visualisation of SLAM data, in particular the key points extracted from the 2D LiDAR sensor of the mapped environment. In addition, our HMI displays the data processed in different environments, here ROS and MATLAB, in real time.

III. Robotic System

The robotic system used in this work is responsible for performing SLAM tasks. It creates a two-dimensional map of the environment in which it operates, while simultaneously determining its precise position.

All experiments have been carried out with 4WD Rover Zero 3, developed by *Rover Robotics*. The robot is shown in Figure 1, along with the sensors used for the characterisation of the surrounding environment. For visualisation purposes, the camera *Intel® RealSense™ D435i* has been employed. An omnidirectional 2D LiDAR sensor has been used for mapping and positioning tasks in real time, namely *RPLiDAR S2* from *Slamtec*. The signals from the sensors are processed by the *Intel NUC10i* mini PC located on board. Obtained data is analysed in real time using the ROS framework. The robot is remotely controlled from a virtual joystick implemented in the designed HMI.

Figure 1. Teleoperated robotic system used in SLAM experiments.

Both the robotic platform and the external PC running the HMI are connected to the same network via WiFi. Information exchange between both systems, represented by dashed arrows in Figure 2, is thus allowed.

IV. SLAM Methodology

Our robot is designed to perform real-time mapping and positioning tasks using SLAM technology. This involves collecting data from sensors, camera and LiDAR, to build a map of the environment. At the same time, it uses this data to estimate the position and trajectory of the robot within the map. Since these tasks require large processing power, part of the computational load is delegated to MATLAB. This leads the HMI to play a key role in the acquisition and visualisation of sensor data, ROS data as well as MATLAB data allowing the robot to make informed decisions and navigate in a guided manner in real-time within its environment. The system architecture and the flow of information between the different system components is shown in Figure 2.

A. LiDAR SLAM

The map and the trajectory followed by the robot are obtained in real time by means of Google's *Cartographer* [16]. It runs on the ROS framework of the mini PC that is integrated in the robotic system. This algorithm works with submaps to facilitate information processing. As the robot moves, *Cartographer* gradually adds the LiDAR sensor readings to the last generated submap [17].

B. Feature extraction

Mapping any environment generates a large volume of information. The current trend in SLAM processing is to reduce this raw data to a set of characteristics that define the mapped profile [18]. In structured environments [19], it is very common to use extraction techniques that focus on identifying key features such as straight sections, curved sections and corners. These features are essential to understand and properly represent the geometry of the environment.

 ISBN: 978-1-68558-163-3

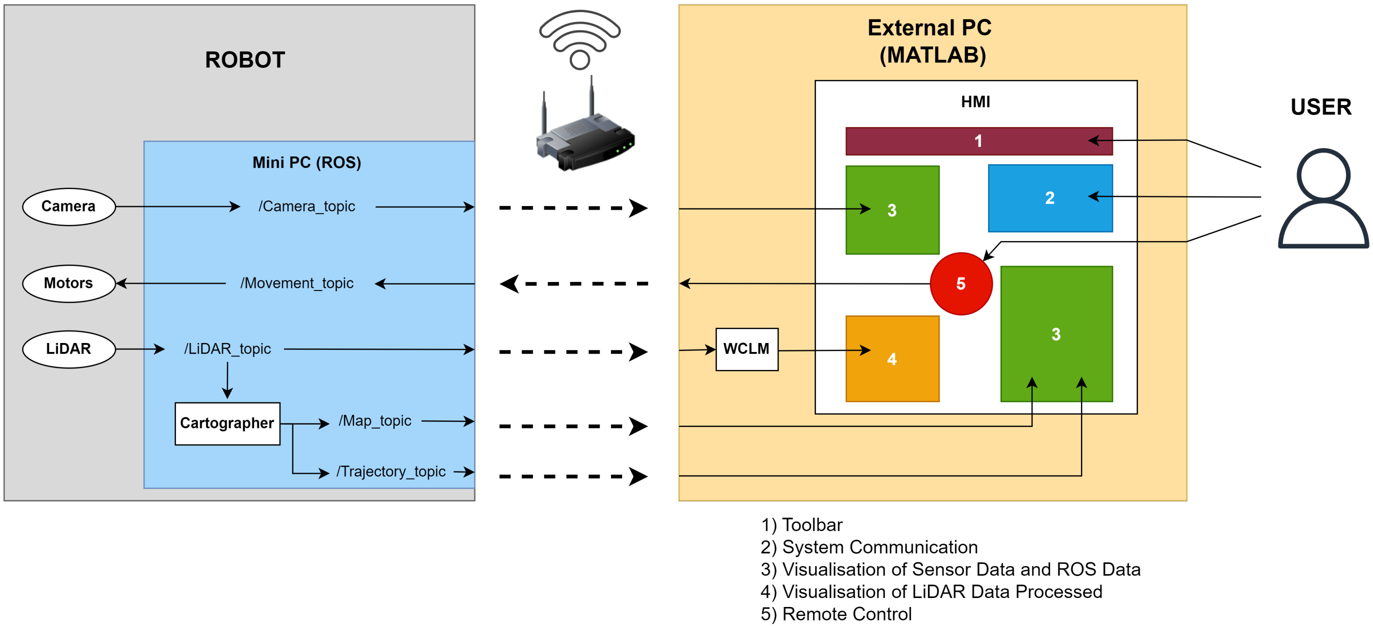

Figure 2. System Architecture Diagram.

The profile under study is structured. Therefore, we will extract its corners in real time from the current LiDAR scan. This part of the process is performed on MATLAB. We use Weighted Conformal LiDAR-Mapping (WCLM) [20] for this purpose. This methodology allows for the extraction of corners while determining their uncertainties, using the data provided by a 2D LiDAR sensor (distance ρ and angle θ).

From the readings x_i and y_i corresponding to each point of the mapped profile, the coordinates x_Q and y_Q defining each straight line in the inverse domain (1) can be extracted [20].

$$x_Q \cdot \mathbf{x_i} - y_Q \cdot \mathbf{y_i} = 1 \tag{1}$$

The expressions for x_Q and y_Q are (2) and (3), respectively.

$$\widehat{x_Q} = \frac{\left(\sum_i w_i x_i\right)\left(\sum_i w_i y_i^2\right) - \left(\sum_i w_i y_i\right)\left(\sum_i w_i x_i y_i\right)}{\left(\sum_i w_i x_i^2\right)\left(\sum_i w_i y_i^2\right) - \left(\sum_i w_i x_i y_i\right)^2} \tag{2}$$

$$\widehat{y_Q} = \frac{\left(\sum_i w_i x_i\right)\left(\sum_i w_i x_i y_i\right) - \left(\sum_i w_i y_i\right)\left(\sum_i w_i x_i^2\right)}{\left(\sum_i w_i x_i^2\right)\left(\sum_i w_i y_i^2\right) - \left(\sum_i w_i x_i y_i\right)^2} \tag{3}$$

Where the weighting factor is $\omega_i = \frac{1}{\sigma_\rho^2 (\rho \sigma_\theta)^2}$, while σ_ρ and σ_θ are the uncertainties in distance and angle of the laser rangefinder, respectively.

The covariance matrix C_Q can be obtained as follows [20].

$$C_Q = \begin{bmatrix} V_{\widehat{x_Q}} & C_{\widehat{x_Q}\widehat{y_Q}} \\ C_{\widehat{x_Q}\widehat{y_Q}} & V_{\widehat{y_Q}} \end{bmatrix} \tag{4}$$

Once the parameters x_Q and y_Q of the straight lines and their uncertainties are known, the corner coordinates of the mapped profile x_c and y_c can be extracted following equations (6) and (7), respectively.

$$x_{Q_i} \cdot x_c - y_{Q_i} \cdot y_c = 1 \tag{5}$$

$$x_c = \frac{y_{Q_{i+1}} - y_{Q_i}}{x_{Q_i} \cdot y_{Q_{i+1}} - y_{Q_i} \cdot x_{Q_{i+1}}} \tag{6}$$

$$y_c = \frac{x_{Q_{i+1}} - x_{Q_i}}{x_{Q_i} \cdot y_{Q_{i+1}} - y_{Q_i} \cdot x_{Q_{i+1}}} \tag{7}$$

The covariance matrix of the corners is defined by (8).

$$C_c = J_c \cdot \begin{bmatrix} V_{x_{Q_i}} & C_{xy_{Q_i}} & 0 & 0 \\ C_{xy_{Q_i}} & V_{y_{Q_i}} & 0 & 0 \\ 0 & 0 & V_{x_{Q_{i+1}}} & C_{xy_{Q_{i+1}}} \\ 0 & 0 & C_{xy_{Q_{i+1}}} & V_{y_{Q_{i+1}}} \end{bmatrix} \cdot J_c^T \tag{8}$$

Where J_c is the Jacobian matrix of corner parameters with respect to line parameters x_Q and y_Q.

V. HMI MODEL

The aim in the design and development of the HMI is to facilitate the control of a SLAM robotic system using 2D LiDAR technology. This interface is designed to perform this function in different indoor environments, whether in industrial, domestic or similar contexts. Its design should be intuitive and accessible to a wide variety of users, even those without previous experience in handling ROS based systems.

Real-time bidirectional communication between the human operator and the robotic system must be facilitated by this interface. This communication has to be smooth and efficient facilitating users' decision making. As all data from the sensors is processed in ROS, information must be transferred to the interface. In this way, it is available to the user without having to directly access ROS. For this purpose, we have chosen MATLAB R2023a as the programming environment for the design and development of both the interface and the processing of the signals from ROS.

A. HMI Design

The interface we propose has been entirely designed using MATLAB's *App Designer*. One of the main advantages of using this environment is the ability to generate standalone applications that can be run outside MATLAB. This allows the interface, once designed, to be exported as an independent application using *MATLAB Compiler*. The generated executable can be installed on any device and is compatible with a wide variety of operating systems, the main ones being Windows, Linux and macOS. This makes our HMI an accessible interface for any user.

The resulting graphical user interface is presented in Figure 3. This interface has been carefully designed to provide an intuitive and efficient user experience. Other designs such as [15] were tested but the proposal presented below extends the functionalities of the interface, provides more information by displaying features extracted from 2D LiDAR data in real time with a more compact visualisation. It is divided into five sections as each one defines a specific functionality of the application, see Figures 2 and 3.

1) Toolbar: This section provides quick access to the key functions of the application. These functions include the display of previously stored data, the possibility to perform delayed analysis and the ability to save processed data.

2) System Communication: In this section, users can configure the communication between ROS and MATLAB providing the robot's IP address.

3) Visualisation of Sensor Data and ROS Data: A significant part of the interface is dedicated to the visualisation of raw data as well as the data processed in ROS. This includes the real-time transmission of the camera information, as well as the Geographic Information System (GIS) representation of the map and the path followed by the robot, which are generated by the mapping algorithm *Cartographer*. These visual elements provide the user with a clear and complete representation of the robot's environment.

4) Visualisation of LiDAR Data Processed: Another essential function of the application is the LiDAR data analysis area where the WCLM algorithm detects the corners of the mapped environment. Here, the interface allows visualisation of the LiDAR data processed directly in MATLAB. This involves the extraction of the corners that make up the environment and the subsequent reconstruction of the profile using this information. This real-time LiDAR data analysis is a valuable feature that allows users to obtain detailed information about the surrounding environment.

5) Remote Control: A virtual joystick has been incorporated in the centre of the interface. This interactive component allows users to send movement commands to the robot in an intuitive and precise way.

Real-time updating of the user interface has been one of the challenges faced both in the design of the interface and in the work in general.

B. Signal Processing

One of the main milestones of this work is the ability to process data from the sensors and send instructions to the robot in real time. This is possible through MATLAB's *ROS Toolbox*. This tool allows communication and interaction between MATLAB and ROS. In this case, it allows for the implementation of the WCLM algorithm [20] in MATLAB with the LiDAR data from ROS.

The main objective is being able access to the available *topics* in ROS from the HMI. In this way, messages can be sent and received between both systems. Among the wide variety of options offered by the *ROS Toolbox*, we will focus on initiating communication with ROS, subscribing to the *topics* of the sensors and *Cartographer*, publishing in the *topics* related to the control of the actuators and closing the communication.

The first step is knowing the IP address of the robot in order to establish communication from the interface. Once the connection to the *ROS Master* has been initiated, the information from the camera, the LiDAR sensor and the map and trajectory data can be accessed by subscribing to the *topics* that host these messages. Movement commands to the robot are published in the *topic* designated for this purpose. Finally, when the exploration is completed or the HMI is closed, both systems are disconnected.

VI. Experimental Results

The 4WD Rover Zero 3 robot was employed in the experiments, which involved mapping the corridors on the ground floor of the Technological Building of the School of Industrial, Computer and Aerospace Engineering of the University of León. This research addressed several aspects. First and foremost, we sought the effective visualisation of the resulting map on GIS. In second place, we evaluated the effectiveness of the bidirectional communication between the robot and the Human Machine Interface used in the study. Third, a calculation process of the uncertainties associated to the measurements was launched.

The first step was to analyse the data processed in ROS from *Cartographer*. In Figure 4, the map can be seen in yellow, and the trajectory followed by the robot, in blue. This information is represented by MATLAB on a satellite image provided by Google. This *Cartographer*-generated map is built as the robot moves through the environment. Please note that the robot is teleoperated from the HMI, which sends instructions in real time. A human operator makes the movement decisions based on the data stream from the camera.

The fluent real-time communication between the two systems, i.e., the robot and the HMI, results in an efficient interaction between the operator and the robot. This constant flow

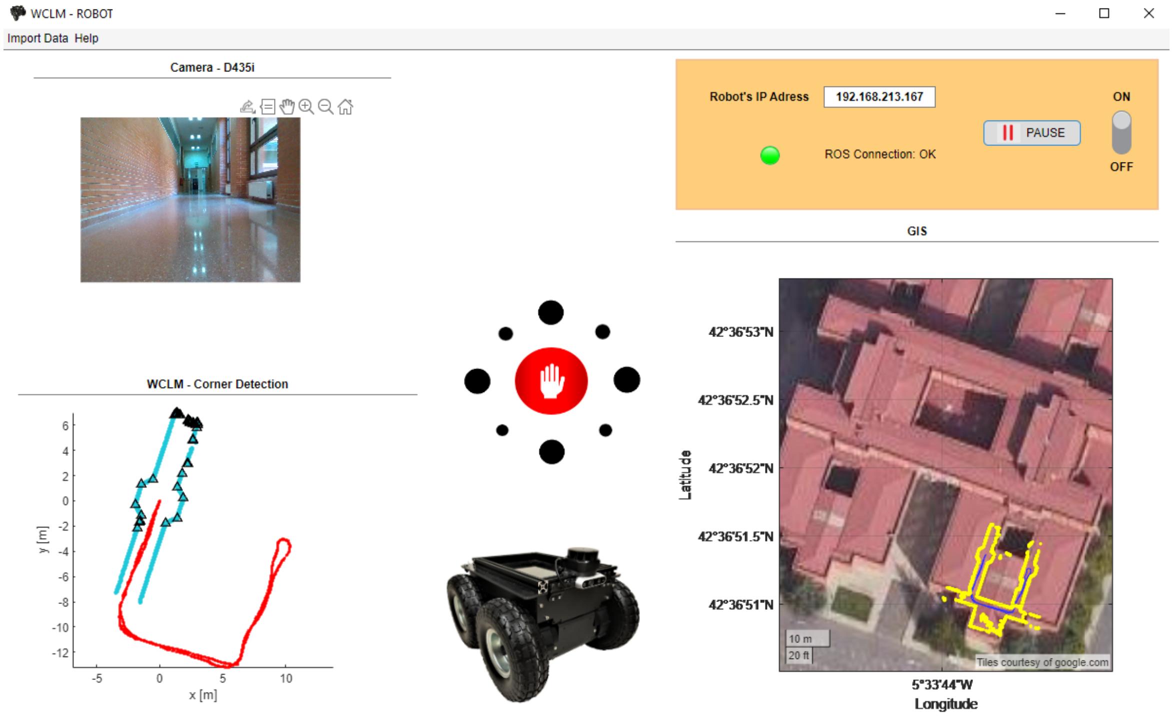

Figure 3. Human-Machine Interface for industrial robot teleoperation.

Figure 4. Map (yellow) and trajectory (blue) followed by the robot during the mapping of the Technological Building of the School of Industrial, Computer and Aerospace Engineering of the University of León.

of information benchmarks one of the fundamental aspects of this work: the efficiency of bidirectional communication. This has been noted by several users, who also indicated an intuitive interface.

Since *Cartographer* does not provide us with the uncertainty levels for each point on the map, we have extracted the features of the environment mapped by the LiDAR sensor. For this purpose, the WCLM method proposed by Prieto-Fernández et al. [20] is used. Following this methodology, it is enough to have the LiDAR measurements and to know the uncertainty of the sensor used to determine the coordinates of the corners and their uncertainties. According to the information provided by the manufacturer, its angular resolution is ($\sigma_\theta = 0.12/2^o$), and its depth accuracy ($\sigma_\rho = \pm 5\ cm$).

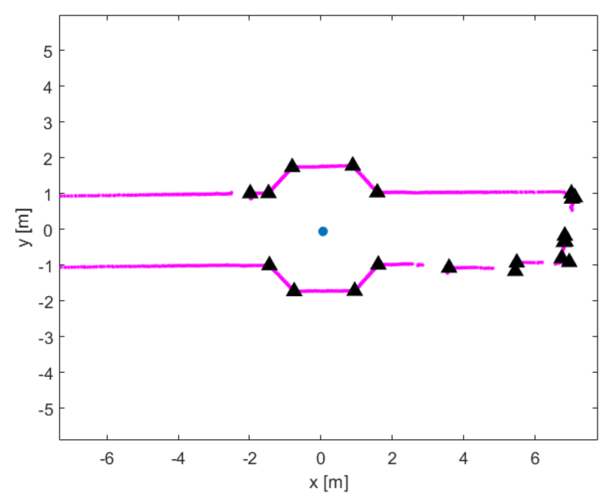

Figure 5. WCLM for corner extraction for position A. The LiDAR data is presented in pink, corners are highlighted in black and sensor position in blue.

Figures 5 and 6 show the same corridor of the Technology Building mapped from two different positions, which allows to observe how the uncertainty of the same corners varies

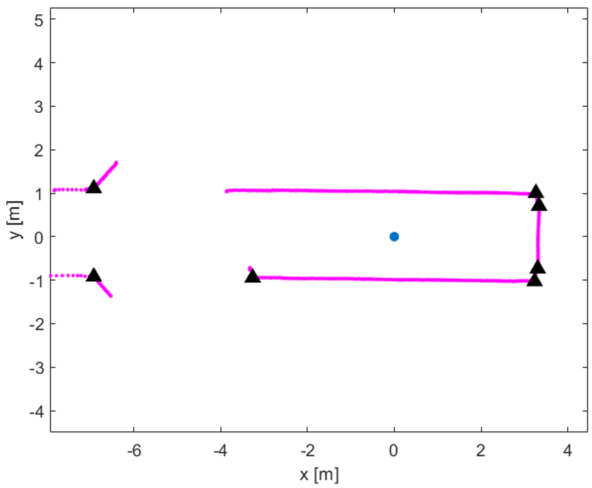

Figure 6. WCLM for corner extraction for position B. The LiDAR data is presented in pink, corners are highlighted in black and sensor position in blue.

according to the acquisition point. The measurements were taken from the point marked in blue, the mapped profile is represented in pink and the corners obtained by the WCLM method are highlighted in black.

TABLE I
WCLM CORNER DETECTION AT POSITION A

Corner	x_c [m]	y_c [m]	σ_{x_c} [mm]	σ_{y_c} [mm]
1	-1.500	-0.948	18.040	3.816
2	-0.808	-1.675	19.345	11.250
3	0.888	-1.664	19.455	11.350
4	1.549	-0.933	19.579	8.236
5	3.530	-1.018	28.563	4.473
6	5.384	-1.109	28.535	19.950
7	5.427	-0.874	29.045	5.182
8	6.896	-0.860	27.371	7.634
9	6.685	-0.752	21.567	16.372
10	6.745	-0.299	24.151	12.443
11	6.788	-0.286	30.484	15.737
12	6.771	-0.122	30.117	23.151
13	6.971	0.913	22.910	18.326
14	7.052	0.951	30.833	17.280
15	6.956	1.059	45.764	14.571
16	1.520	1.087	16.943	3.901
17	0.834	1.836	18.108	10.170
18	-0.861	1.795	20.021	11.556
19	-1.525	1.066	21.029	10.480
20	-2.038	1.063	27.201	9.364

TABLE II
WCLM CORNER DETECTION AT POSITION B

Corner	x_c [m]	y_c [m]	σ_{x_c} [mm]	σ_{y_c} [mm]
1	-6.921	-0.919	29.243	13.652
2	-3.261	-0.943	19.219	2.340
3	3.233	-1.027	23.780	2.168
4	3.304	-0.735	13.786	143.877
5	3.335	0.707	11.264	112.222
6	3.263	1.002	35.472	2.791
7	-6.921	1.109	26.825	12.249

The data presented in tables I and II includes the Cartesian coordinates of these corners as well as their uncertainties. Uncertainty levels are found to be intrinsically related to the number of points that delimit each corner. This relationship is explained by the fact that the corners are generated from the intersection of straight sections, and the number of points involved in the calculation of these straight sections significantly influences the uncertainty of the resulting corner. Matching corners on both maps are highlighted with the same colour in both tables. Please note that the same point may have different uncertainty levels in each map. This is due to the relative distance between corner and sensor in each scenario. In the case of red and green labelled corners, position A is characterised by the sensor being closer to these corners, which leads to higher data availability for position calculations, thus resulting in a lower uncertainty compared to position B. The case of the yellow corner is inverse: at position B, a larger amount of information is available compared to position A, resulting in a lower uncertainty.

These observations show the influence of the relative location of the sensor with respect to the objects of interest on the accuracy and uncertainty of the measurements. The reliability of the results is found to strongly depend on careful consideration of the position and layout of sensors.

VII. CONCLUSION

This work addresses one of the key challenges in signal processing within robotic systems focused on 2D LiDAR SLAM. The developed graphical user interface enables remote analysis and bidirectional communication in real time with the robot, thereby enhancing control and decision making capabilities and broadening application range into a variety of industrial and scientific contexts.

The main achievement of the HMI is its ability to control and monitor the robotic system both in real time and deferred scenarios while visualising the data processed in different environments, ROS and MATLAB. The latter endows the interface with multidisciplinary versatility, allowing users to generate their own data analysis code outside of the ROS environment. It facilitates the collection, visualisation and storage of such information for further processing, which is particularly valuable in research and development environments.

We can conclude that the resulting HMI meets the initial objectives. It is an intuitive and efficient interface, aimed at a wide range of users. Versatility is coupled with accesibility, since multiple platforms are supported and MATLAB is not required on the end device, turning the resulting product into a plug-and-play commodity for any user.

ACKNOWLEDGMENT

This research has been carried out thanks to TRESCA Ingeniería S.A. within the framework of the Research Project on *Survey Systems for Hostile Structured Environments using Simultaneous Localization And Mapping (SLAM) Systems with Lidar Technology.* The project has been co-financed by the ERDF, Thematic Objective 1, which seeks to promote technological development, innovation and quality research, and the

Regional Government of Castilla y León, through the institute for the Business Competitiveness of Castilla y León (ICE).

References

[1] J. J. Choi, Y. Kim, and S. S. Kwak, "The autonomy levels and the human intervention levels of robots: The impact of robot types in human-robot interaction," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2014, pp. 1069–1074.

[2] J. P. Queralta *et al.*, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[3] B. S. Peters, P. R. Armijo, C. Krause, S. A. Choudhury, and D. Oleynikov, "Review of emerging surgical robotic technology," *Surgical endoscopy*, vol. 32, pp. 1636–1655, 2018.

[4] A. Wichmann, B. D. Okkalioglu, and T. Korkmaz, "The integration of mobile (tele) robotics and wireless sensor networks: A survey," *Computer Communications*, vol. 51, pp. 21–35, 2014.

[5] A. Steinfeld *et al.*, "Common metrics for human-robot interaction," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, 2006, pp. 33–40.

[6] T. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Autonomous robots*, vol. 11, pp. 9–18, 2001.

[7] H. Su *et al.*, "State of the art and future opportunities in mri-guided robot-assisted surgery and interventions," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 968–992, 2022.

[8] H. Lv *et al.*, "Gulim: A hybrid motion mapping technique for teleoperation of medical assistive robot in combating the covid-19 pandemic," *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 1, pp. 106–117, 2022.

[9] T. Klamt *et al.*, "Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.

[10] A. F. Winfield *et al.*, "eurathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots," in *Towards Autonomous Robotic Systems: 17th Annual Conference, TAROS 2016, Sheffield, UK, June 26–July 1, 2016, Proceedings 17*. Springer, 2016, pp. 351–363.

[11] D. Aschenbrenner, M. Fritscher, F. Sittner, M. Krauß, and K. Schilling, "Teleoperation of an industrial robot in an active production line," *IFAC-PapersOnLine*, vol. 48, no. 10, pp. 159–164, 2015.

[12] M. M. Dalvand and S. Nahavandi, "Improvements in teleoperation of industrial robots without low-level access," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 2170–2175.

[13] I. Cerulo, F. Ficuciello, V. Lippiello, and B. Siciliano, "Teleoperation of the schunk s5fh under-actuated anthropomorphic hand using human hand motion tracking," *Robotics and Autonomous Systems*, vol. 89, pp. 75–84, 2017.

[14] R. Su, K. Xu, L. Zhao, and P. Yu, "A heterogeneous master-slave teleoperation method for 7-dof manipulator," in *2021 China Automation Congress (CAC)*. IEEE, 2021, pp. 1740–1744.

[15] N. Prieto Fernández, M. Bayón Gutiérrez, S. Fernández-Blanco, Á. Fernández-Blanco, F. Carro de Lorenzo, and J. A. Benítez-Andrades, "Desarrollo de un hmi en ros-matlab para la teleoperación de robots industriales," in *XLIV Jornadas de Automática*. Universidade da Coruña. Servizo de Publicacións, 2023, pp. 720–725.

[16] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[17] R. Yagfarov, M. Ivanou, and I. Afanasyev, "Map comparison of lidar-based 2d slam algorithms using precise ground truth," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 1979–1983.

[18] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, pp. 97–111, 2007.

[19] P. Núñez, R. Vázquez-Martín, J. Del Toro, A. Bandera, and F. Sandoval, "Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation," *Robotics and Autonomous Systems*, vol. 56, no. 3, pp. 247–264, 2008.

[20] N. Prieto-Fernández, S. Fernández-Blanco, Á. Fernández-Blanco, J. A. Benítez-Andrades, F. Carro-De-Lorenzo, and C. Benavides, "Weighted conformal lidar-mapping for structured slam," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.