Received 29 October 2024; revised 12 March 2025; accepted 1 July 2025.
Date of publication 4 July 2025; date of current version 18 July 2025.
This article was recommended by Executive Editor David Wettergreen.

Digital Object Identifier 10.1109/TFR.2025.3586209

Multirobot Decentralized Collaborative SLAM in Planetary Analogue Environments: Dataset, Challenges, and Lessons Learned

PIERRE-YVES LAJOIE (Member, IEEE), KARTHIK SOMA, HAECHAN MARK BONG (Graduate Student Member, IEEE), ALICE LEMIEUX-BOURQUE, RONGGE ZHANG, VIVEK SHANKAR VARADHARAJAN, AND GIOVANNI BELTRAME (Senior Member, IEEE)

Department of Computer and Software Engineering, Polytechnique Montréal, Montreal, QC H3T 0A3, Canada

CORRESPONDING AUTHOR: PIERRE-YVES LAJOIE (pierre-yves.lajoie@polymtl.ca)

This work was supported in part by the Vanier Canada Graduate Scholarships Award, in part by the Fonds de recherche du Québec—Nature et technologies, and in part by the Canadian Space Agency.

(Field Report)

ABSTRACT Decentralized collaborative simultaneous localization and mapping (C-SLAM) is essential to enable multirobot missions in unknown environments without relying on preexisting localization and communication infrastructure. This technology is anticipated to play a key role in the exploration of the Moon, Mars, and other planets. In this article, we share insights and lessons learned from C-SLAM experiments involving three robots operating on a Mars analogue terrain and communicating over an ad hoc network. We examine the impact of limited and intermittent communication on C-SLAM performance, as well as the unique localization challenges posed by planetary-like environments. Additionally, we introduce a novel dataset collected during our experiments, which includes real-time peer-to-peer inter-robot throughput and latency measurements. This dataset aims to support future research on communication-constrained, decentralized multirobot operations.

INDEX TERMS Collaborative simultaneous localization and mapping (C-SLAM), multirobot systems, planetary analogue environments.

I. INTRODUCTION

MULTIROBOT systems hold the potential to revolutionize space and planetary exploration. Teams of robots can parallelize work, be more resilient to individual failures, and, most importantly, collaborate to accomplish collective tasks that are out of reach of single-robot systems. However, operating robots on another planet presents some unique challenges, such as large communication delays with base stations on Earth, difficult and unknown terrain, or the absence of any preexisting infrastructure. In these conditions, robots need high levels of autonomy to operate safely.

One of the key enablers of robot autonomy is the simultaneous localization and mapping (SLAM) algorithm [1], which provides localization estimates of the robot and a map of the surrounding environment that can be used for terrain analysis, path planning, and decision-making. In the case of multirobot systems, the robots need to collaborate in the localization and mapping process in order to converge to a consistent perception of the environment across the team of robots. Without shared situational awareness, individual robots are constrained by their limited view of the environment and are not able to collaborate efficiently. Thus, collaborative SLAM (C-SLAM) [2], [3] is likely to be a vital component of future multirobot missions on the Moon, Mars, or other planets.

That being said, there are additional key requirements for the efficient deployment of C-SLAM algorithms in space. First and foremost, due to the high cost of space missions, the systems need to be as robust and resilient to failures as possible. Also, due to the lack of preexisting localization and networking infrastructure, individual robots need

enough autonomy from any base station (on Earth or on the explored planet itself) to survive frequent and lengthy disconnections over the course of their missions. Given those constraints, typical centralized approaches to C-SLAM [4], [5], in which robots send measurements to a central computing node that computes and shares back the merged global map and localization estimates, are highly vulnerable to network disconnections or the outright failure of the central computer.

To mitigate these risks, decentralized techniques that are purposely built to work with ad hoc networking and withstand prolonged disconnections between robots are preferable. In prior work, we introduced Swarm-SLAM [6], a decentralized C-SLAM framework satisfying those requirements. Building on this framework, we conducted a series of field experiments and collected a novel dataset, at Canadian Space Agency (CSA) Mars Yard [7], a planetary analogue terrain designed to simulate realistic planetary conditions. We use Swarm-SLAM as a case study to evaluate the current performance of state-of-the-art decentralized C-SLAM. We look in particular at the challenges related to limited and intermittent inter-robot communication, as well as the difficulties posed by the terrain in terms of vibrations, lack of distinctive features, and perceptual aliasing.

A. CONTRIBUTIONS

This article is the culmination of extensive multirobot experiments on the planetary analogue environments shown in Fig. 1. As a result, we present the following contributions.

1) The design of a decentralized three-robot system connected through ad hoc networking and its deployment on a planetary analogue environment.
2) A novel dataset collected during our experiments that includes LiDAR, IMU, and, unlike prior works, peer-to-peer inter-robot communication throughput and latency estimates that are valuable for evaluating the communication consumption of C-SLAM or other multirobot algorithms. Our dataset[1] is available on IEEE DataPort [8].
3) A thorough analysis of the accuracy and efficiency of decentralized C-SLAM, exposing limitations of current approaches and open challenges.

We believe that the challenges and lessons learned from our experiments will be valuable for both the space robotics and C-SLAM research communities.

The remainder of this article is divided as follows. Section II presents background knowledge and related work on decentralized C-SLAM, inter-robot networking, and localization challenges in space analogue environments. Section III describes our experimental setup comprising the robots and sensors used, and key characteristics of the terrain on which they were deployed. Section IV presents the accuracy results of our Swarm-SLAM decentralized C-SLAM algorithm and localization challenges inherent to planetary analogue terrains. Section V discusses the calibration trade-offs between accuracy and resource efficiency in terms of communication and computing. Finally, Section VI offers some insights gained during our experiments and open challenges that we identify for the future of C-SLAM for space robotics.

[1] https://ieee-dataport.org/documents/collaborative-simultaneous-localization-and-mapping-dataset-mars-analogue-terrain-inter

FIGURE 1. **Multirobot experiments at the CSA Mars Yard. In our experiments, three robots simultaneously explored the simulated Martian terrain. The robots collaborated through peer-to-peer communication to map the environment and localize themselves within the landscape, demonstrating the effectiveness of our C-SLAM system in a challenging, planetary analogue setting.**

II. BACKGROUND AND RELATED WORK

A. DECENTRALIZED C-SLAM

1) CENTRALIZED VERSUS DECENTRALIZED

C-SLAM systems are typically categorized into centralized or decentralized solutions. Centralized C-SLAM relies on a main server or base station that gathers mapping data from all participating robots and computes a common global localization and mapping estimate for the whole team. This setup requires robots to maintain a stable, continuous connection to the base station, leading to significant challenges in scalability due to potential communication bottlenecks. It is also vulnerable to failures in the central computing node [9]. Such stringent connectivity and reliability requirements can be impractical, particularly in planetary environments without preexisting networking architectures.

In contrast, decentralized C-SLAM systems operate without a centralized server, instead utilizing occasional, peer-to-peer communication among robots. This approach is advantageous in environments where constant connectivity is unreliable or impossible. However, decentralized systems face their own set of challenges, as they are limited by the robots' individual computing and communication resources. They also demand more complex strategies for data handling and coordination to ensure accurate and consistent SLAM results across the team of robots [3].

Both centralized and decentralized C-SLAM systems share a similar architecture with single-robot SLAM systems, consisting of two key components: the front end and the back end. The front end is responsible for tasks such as feature extraction and data association, which involve identifying and matching environmental features to aid in robot relocalization. The back end, on the other hand, is dedicated to state

estimation, determining the robots' poses (i.e., their positions and orientations) relative to the constructed map of the environment [1]. In the collaborative state estimation, the maps and poses of the robots are integrated into a common frame of reference, allowing for consistent and unified positioning across all robots [2].

2) FRONT END

A major challenge in the C-SLAM front end is efficiently detecting and computing inter-robot loop closures. These loop closures occur when two or more robots identify common landmarks or locations they have previously explored. Such shared features act as connection points that allow the integration of local maps from individual robots into a unified global reference frame. Because transmitting entire maps can be prohibitively expensive in terms of communication costs, inter-robot loop closure detection is typically handled in a two-step process [10], [11].

In the first step, robots exchange compact descriptors, which are simplified representations of their data, such as image descriptors (e.g., CosPlace [12]) or LiDAR scan descriptors (e.g., ScanContext [13]). These descriptors enable place recognition by calculating similarity scores to identify overlaps in the environments mapped by different robots. High similarity scores suggest potential loop closure candidates, indicating that the robots might have traversed the same place.

The second step focuses on these identified candidates. For each candidate with high similarity, more detailed and computationally expensive descriptors, like 3D keypoints or full scans, are exchanged. These are used to perform precise geometric registration between the corresponding data from different robots, establishing accurate positional and rotational links between them. The resulting pose measurements, called loop closures, are then integrated into the robots' pose graphs, to merge the maps and enhance the state estimation accuracy.

In the case of LiDAR scans, since point clouds often include noise and outliers, robust methods like TEASER++ [14] are employed to ensure accurate registration without needing an initial pose guess. This capability is especially valuable in C-SLAM, as the robots generally do not know their relative positions or trajectories prior to the first map merging.

3) BACK END

The C-SLAM back end is tasked with estimating the most likely robot poses and maps from the noisy data collected by the robots. To reduce the computational cost of large-scale SLAM problems, which is especially challenging for multirobot mapping, most approaches utilize some form of pose graph optimization. This method marginalizes environmental features into interpose measurements, solving the optimization problem by focusing only on the poses. While single-robot solvers can be directly applied to multirobot scenarios, as demonstrated in our approach [6], several distributed multirobot solvers have been developed [15], [16], [17]. These distributed approaches improve computational scalability by distributing the workload among connected robots. However, they require extensive bookkeeping, a large number of iterations during optimization, such that network delays can diminish their computational benefits. To mitigate these challenges, Fan and Murphey [18] propose a majorization–minimization method for distributed pose graph optimization, which accelerates convergence with theoretical guarantees. Meanwhile, McGann and Kaess [19] introduce an incremental distributed optimization approach that efficiently estimates robot states with only sparse pairwise communication. In contrast, our method streamlines optimization by dynamically electing a single robot to perform the computations during each rendezvous.

A common challenge in SLAM systems, including C-SLAM, is the occurrence of erroneous measurements due to perceptual aliasing [20]. Perceptual aliasing arises when distinct locations in an environment appear overly similar and lack distinguishing features, leading to incorrect data associations during place recognition. In C-SLAM systems, this challenge has been well-documented across various environments. For example, Ebadi et al. [21] highlight the impact of perceptual aliasing in subterranean settings, while Tian et al. [22] report frequent occurrences in forested environments. To mitigate this issue, Mangelson et al. [23] introduced pairwise consistency maximization (PCM), which enhances robustness by identifying the largest set of consistent inter-robot measurements. In another line of work, Yang et al. [24] proposed the graduated nonconvexity (GNC) algorithm, a flexible and robust tool for various optimization tasks. GNC is employed by the elected robots in Swarm-SLAM for pose graph optimization. Tian et al. [25] have extended GNC for use in distributed implementations.

Although visual loop closure detection has advanced significantly [12], [26], vision-based methods face major challenges in planetary analogue environments. Changes in appearance due to varying viewpoints—exacerbated by the limited field of view of cameras—hinder reliable detection. Additionally, variations in illumination and the presence of dust can degrade image quality, while the scarcity of unique visual features in these environments often leads to perceptual aliasing, complicating data association [27]. In contrast, LiDAR is well-suited for mapping in visually degraded conditions, as it operates independently of ambient light and provides precise 360° range measurements. To mitigate the limitations of both sensors, cameras are frequently integrated with LiDAR, leveraging their complementary properties to enhance loop closure detection [28], [29].

Recently, numerous C-SLAM frameworks have emerged, each contributing to advancements in distributed mapping. DSLAM [10] was a pioneering system that used CNN-based image descriptors for place recognition and distributed pose graph optimization. DOOR-SLAM [11] further developed these ideas by integrating PCM for outlier rejection and adapting the system to handle intermittent

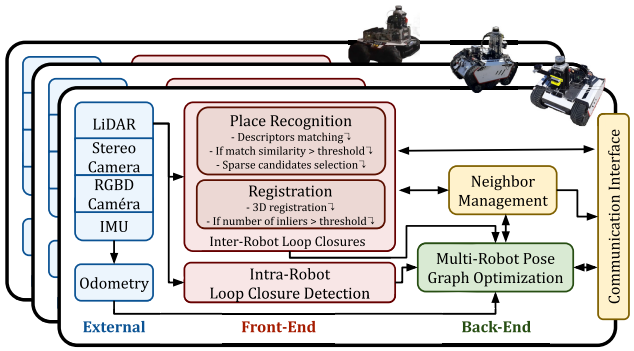

FIGURE 2. Swarm-SLAM overview. Our C-SLAM system is fully decentralized and supports intermittent inter-robot communication. Swarm-SLAM takes odometry and raw sensor data as input, performs place recognition and registration to produce loop closures, and dynamically elects a robot among the network neighbors to optimize the multirobot pose graph.

inter-robot communication, eliminating the need for full connectivity between robots. DiSCo-SLAM [30] expanded these concepts to LiDAR-based mapping, using ScanContext descriptors [13] for place recognition. Kimera-Multi [25] combines classical approaches to place recognition and registration with a distributed version of GNC. More recently, SlideSLAM [31] introduced a decentralized metric-semantic SLAM framework that enables heterogeneous robot teams to collaboratively build and merge object-based maps.

Centralized C-SLAM systems have also advanced considerably. COVINS [32] optimizes visual–inertial SLAM by reducing computational load through the elimination of redundant keyframes. LAMP 2.0 [33] employs a graph neural network-based prioritization mechanism to evaluate inter-robot loop closure candidates, predicting the optimization outcomes and selecting the most promising candidates for further processing. Maplab 2.0 [34] is designed to support varying sensor modalities and configurations, enabling flexible multirobot mapping.

Additionally, related to our work, the ARCHES project [35] explores heterogeneous robotic teams for collaborative navigation and sampling, demonstrating multirobot 6D SLAM [36] on a planetary analogue environment.

Swarm-SLAM [6], our recently proposed system described in Fig. 2, builds on these advancements by introducing a novel sparse inter-robot loop closure prioritization technique to reduce communication overhead. It uses ROS 2 and includes a neighbor management system that integrates smoothly with ad hoc networking, enhancing C-SLAM's adaptability to intermittent communication scenarios. For a comprehensive review of C-SLAM technologies, we refer the readers to [3].

B. AD HOC INTER-ROBOT COMMUNICATION

Ad hoc networks play a crucial multimodal role in enabling multirobot mapping, allowing robots to communicate directly with one another without relying on a preexisting infrastructure. In the early exploration of ad hoc inter-robot communication for collaborative mapping, Sheng et al. [37] proposed a 2-D grid-based approach that minimizes data exchange by leveraging known relative poses between robots. More recently, Varadharajan et al. [38] addressed the broader challenge of efficiently sharing large volumes of data, such as maps, within distributed robot networks by introducing a peer-to-peer data-sharing system specifically designed for high data loads. To ensure reliable inter-robot communication, it is crucial to consider the robot topology, as it directly influences the available communication paths between connected agents [39], [40]. For example, Varadharajan et al. [41] proposed a fully decentralized connectivity algorithm robust against individual robot failures. This approach allows robots to autonomously adjust their positions to maintain network connectivity with a ground station, ensuring stable communication despite dynamic conditions and potential disconnections. In the specific context of C-SLAM, Giamou et al. [42] and Tian et al. [43] acknowledge the communication bottleneck issue and developed near-optimal data-sharing strategies to avoid duplication during rendezvous between multiple robots.

While some C-SLAM systems, like those described in [10] and [32], require fully connected networks, recent approaches have been designed to be resilient to disconnections. For instance, our prior work [11] and Tian et al. [44] propose C-SLAM frameworks that can withstand intermittent communication losses. More recently, we successfully deployed C-SLAM with ad hoc networking in real-world settings [6], demonstrating the practical feasibility of decentralized multirobot mapping in challenging environments without the need for complex continuous connectivity maintenance.

C. MULTIROBOT MAPPING DATASETS

Existing datasets in C-SLAM typically fall short of capturing realistic multirobot scenarios, as they often involve only one robot at a time, resulting in the absence of dynamic objects, and systematically lack inter-robot network condition estimates. This represents a significant gap in the literature, particularly regarding experimental data from planetary analogue environments where inter-robot communication is intermittent due to large distances and obstacles that cause non-line-of-sight conditions between robots.

One of the early efforts in multirobot datasets was the UTIAS dataset by Leung et al. [45], which involved five robots operating within a single indoor room. This dataset set the groundwork for C-SLAM but was limited to a static, confined environment. More recently, Dubois et al. [46] proposed incorporating both ground and aerial robots in indoor settings, using stereo cameras. Collected during larger scale outdoor environment, Zhu et al. [47] introduce GrAco, a multimodal dataset featuring ground and aerial LiDAR and stereo sequences captured on a college campus. This dataset offered more diverse environmental settings but remained limited to structured outdoor spaces. A notable step forward in scaling and realism was introduced by Tian et al. [44], who developed an online dataset featuring up to eight robots

FIGURE 3. Images of the CSA Mars Yard terrain captured by cameras onboard the robots. (a)–(e) Diverse and challenging features of the terrain. (f) Instances where the robots are occasionally within line of sight of each other, acting as challenging dynamic objects.

TABLE 1. Content per robot of our novel dataset [8], comprising SLAM and inter-robot communication sensing data.

Sensor	Data Types	Frequency (Hz)
Ouster LiDAR OS0	Point Clouds	9.81
	IMU	98.54
	TF frames	624.78
VectorNav VN-100	IMU	196.32
GL iNet AX1800	Pairwise latency with each other robot (ms)	0.92
	Pairwise throughput with each other robot (Kbps)	0.93
U-Blox ZED F9 GPS	GPS fix	~ 1.0

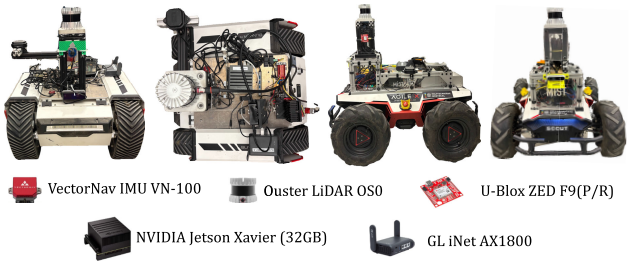

FIGURE 4. Robot design: the base of the robots is an AgileX Scout 2.0 Rover (Robot 1) and AgileX Bunkers (Robots 2 and 3). Each robot is equipped with a LiDAR, IMU sensor suite, and an ad hoc networking router enabling peer-to-peer communication.

equipped with cameras and LiDAR, operating in large-scale, indoor and outdoor environments with human-made structures. Feng et al. [48] introduced the multimodal S3E dataset, which specifically targets C-SLAM scenarios with three synchronized robots in multiple indoor and outdoor environments. Park et al. [49] introduce a novel C-SLAM simulated dataset designed for service environments, where the presence of dynamic objects poses significant challenges.

Most relevant to our work, Zhao et al. [50] addressed some of these limitations with the SubT-MRS dataset, which includes diverse robots operating in various environments, including challenging, degraded conditions similar to the planetary analogue field used in our experiments. This dataset significantly contributes to the field by simulating more realistic conditions for multirobot systems. Our dataset aims to further advance robustness and resilience in degraded environments while also providing data on real inter-robot communication capacity in the field. To that end, our dataset, described in Table 1, includes periodic pairwise latency and throughput estimates.

III. EXPERIMENTAL SETUP

Our experiments were conducted in a planetary analogue field designed to simulate the challenging conditions of planetary surfaces. The terrain included a mix of sand, various types of rocks, slopes, and uneven ground, closely mimicking the environment that robots would encounter on actual space missions (see Fig. 3). These features pose significant challenges for robot mobility, perception, and communication, making this setup ideal for testing decentralized C-SLAM algorithms in realistic conditions. It is important to note that since the environment is on Earth, it is not fully realistic, as there is vegetation in the distance and human operators moving around the area.

The experiments involved three robots exploring the field simultaneously, each remotely controlled by human operators. This setup allowed for testing the robots' ability to maintain ad hoc communication and correctly localize themselves in a dynamic and degraded environment. The robots followed roughly similar trajectories, but each in a different order and direction. This approach ensured that each robot covered nearly the entire field, maximizing overlap between the maps, and thus producing as many loop closing matches as possible for our system analysis in Section V.

A. ROBOT DESIGN

Our experiments were conducted using three robots: one AgileX Scout 2.0 Rover and two AgileX Bunkers, as shown in Fig. 4. All robots were mounted with NVIDIA Jetson Xavier (32 GB) for data processing; GL iNet AX1800 router for wireless network; and Ouster LiDAR OS0, Vector IMU VN-100, and U-Blox ZED F9 modules for GPS positioning (F9P for bunkers and F9R for the rover). Our Swarm-SLAM system was configured with LiDAR as the primary sensor for environment mapping and localization. For odometry, we used the LIO-SAM algorithm [51], which integrates LiDAR and inertial data to provide accurate and robust local pose estimation in real time. For place recognition, we employed ScanContext [13], a method that generates compact descriptors of LiDAR scans for recognizing previously visited locations. To determine similarities between locations, we used cosine similarity to compare the

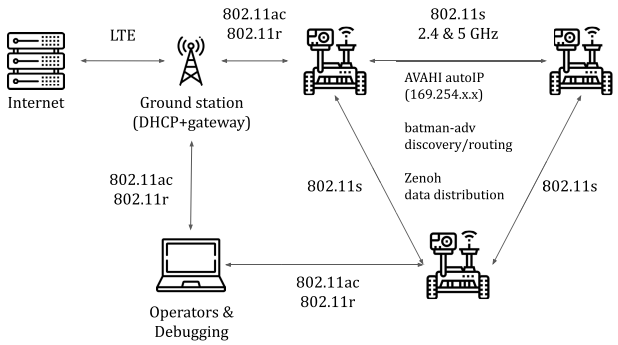

FIGURE 5. Network architecture: robots are interconnected via 802.11s on bonded 2.4- and 5-GHz interfaces, with routing provided by batman-adv and data distribution managed by Zenoh.

ScanContext descriptors. To ensure robust 3-D registration of the point clouds matched with ScanContext, we integrated TEASER++ [14], a state-of-the-art algorithm designed for fast and certifiable registration of point clouds. To evaluate Swarm-SLAM's odometry estimates, the robots were equipped with U-Blox ZED GPS modules. The modules on the robots were programmed to correct their GPS estimates by receiving RTCM correction data (i.e., Radio Technical Commission for Maritime Services messages) from our GPS station. Our GPS RTK (i.e., real-time kinematic positioning) station comprises a laptop connected to our ad hoc network and a U-Blox ZED GPS module. First, the module is configured to perform the *survey-in* procedure, where the module calibrates its position with an accuracy of up to 20 cm. After reaching the target accuracy, our GPS station starts a Networked Transport of RTCM via Internet Protocol (NTRIP) server to broadcast the RTCM correction data on the network. As a result, our system provides GPS estimates at 1 Hz, though occasional delays occur due to communication latency in transmitting correction messages from the GPS station to the robots. Meanwhile, the U-Blox driver publishes the most recent estimate at approximately 10 Hz.

B. AD HOC NETWORKING

We developed a mobile ad hoc network (MANET) using a custom network stack implemented on GL iNet AX1800 routers, which run OpenWRT [52]. Each robot, along with the ground station, was equipped with a router responsible for managing both internal communications among the robot's hardware components and external communications with other robots via the MANET. The inter-robot links were based on IEEE 802.11s, combined with batman-adv, a dynamic link-state routing protocol operating at the data link layer (Layer 2). Batman-adv continuously broadcasts network updates, maintaining a routing table that ensures seamless communication between nodes (i.e., robots) throughout the deployment. The overall communication architecture is illustrated in Fig. 5.

The MANET backbone relies on IEEE 802.11s, configured without frame replication to optimize bandwidth usage and allow batman-adv to provide adaptive routing. Key parameters were adjusted to support fast adaptation to network changes (root mode, active path timeouts, and so on), allowing for rapid disconnection and reconnection under low signal conditions. The 2.4- and 5-GHz radios were bonded into a shared interface managed by batman-adv, enabling dynamic frequency switching. This setup leverages the range of 2.4 GHz, which is more susceptible to interference but offers wider coverage, and the higher bandwidth of 5 GHz, which is better suited for short-range, high-data-rate communications. This dual-frequency capability allows the network to automatically choose the optimal frequency and routing path, supporting both direct communication and multihop communication. We also consider an optional 900-MHz channel for specific use cases requiring long-range communication (several kilometers), which was not needed given the size of the CSA Mars Yard. In fact, we used the standard antennas present on the AX1800 routers to have a more challenging communication environment with ranges up to 40 m, whereas high-gain antennas or more powerful routers could have provided communication range up to hundreds of meters and covered the entire terrain.

To ensure connectivity with the operator computers, the bonded interface was bridged with an 802.11ac link in the station mode and the robots' local Ethernet network. In other words, each robot acts as a wireless access point, and all the onboard computers share the same subnet. The network incorporates 802.11r for fast transitions, enabling operator laptops to seamlessly connect and switch between access points, whether on the ground station or directly to robots. All robots were configured to operate on a unified subnet, while virtual local area networks (VLANs) were used to segregate local components outside the bridge, reducing interference and preventing flooding (e.g., isolating LiDAR data streams from other local network traffic). All computers were set up to use AVAHI AutoIP (i.e., automatic IP address configuration) for service and name discovery under the.local domain. All data distribution between robots is provided by Zenoh [53], which offers a minimal overhead publish-subscribe communication API.

The ground station was configured similar to the robots but had an additional gateway for Internet access. This gateway facilitated software updates for the robots and synchronized their system clocks, ensuring consistent timing across the network. Furthermore, the ground station acted as a control hub, enabling operators to monitor robot status, trigger behaviors, and manage overall mission coordination within the network's communication range.

C. PEER-TO-PEER BANDWIDTH ESTIMATION

To assess the communication performance between the robots, we conducted pairwise peer-to-peer communication estimations at 1-s intervals throughout our data collection experiments. The cumulative throughput was calculated by

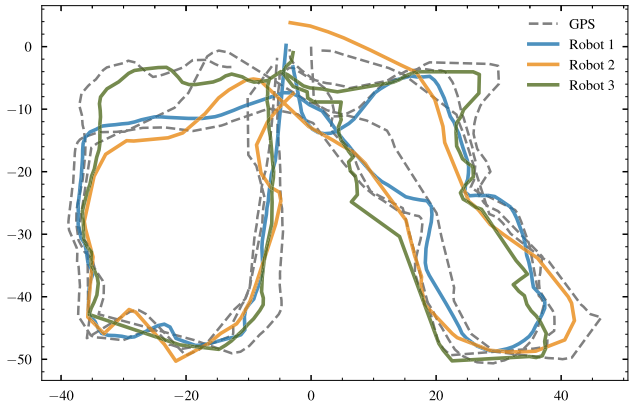

FIGURE 6. Swarm-SLAM pose graph estimates and GPS ground truth. Absolute translation error: 3.74 ± 1.63 m.

summing throughput over consecutive 1-s intervals. We measured throughput using iperf [54], a tool designed for active measurements of the maximum achievable bandwidth. For latency measurements, we employed fping [55], a program that sends the Internet Control Message Protocol (ICMP) echo probes to calculate round-trip times between the robots. This approach enabled us to monitor the latency of communications dynamically as the robots moved through the Mars analogue field, reflecting the impact of varying distances, obstacles, and network topology changes on communication delays. To ensure that our estimates were as accurate and reflective of real-world conditions as possible, we minimized additional network traffic by not running any other software that required data transmission. Thus, we ran our C-SLAM benchmark in postprocessing to assess how well the approach aligns with real-world limitations. On top of serving as a benchmark for inter-robot communication, we believe that our dataset provides valuable insights into network performance under dynamic conditions. It can assist researchers in optimizing data transmission strategies, evaluating adaptive bandwidth allocation methods, and developing robust communication protocols for multirobot systems operating in challenging environments.

IV. DECENTRALIZED C-SLAM

We used Swarm-SLAM to process the entire data sequences from the robots, generating the multirobot pose graph solution illustrated in Fig. 6. The GPS ground-truth data are also displayed for comparison. To quantify the accuracy of the SLAM estimates, we employed the evo software [56] to compute error metrics. We aligned the pose estimates with the GPS data points with the closest timestamps. The results showed an average translation error (ATE) of 3.74 ± 1.63 m between the estimated poses and the ground truth. For conciseness, in this article, we report the ATE as $\mu \pm \sigma$, where μ is the mean error and σ is the standard deviation. The trajectory estimates are jointly aligned with the GPS ground truth using the Umeyama method to evaluate the accuracy of inter-robot localization. We also provide the individual estimates in Fig. 7. For reproducibility, we provide our solution along with the dataset, as well as processing and evaluation scripts. Interestingly, while reasonable, our results are far from perfect, suggesting that future research could leverage our dataset to develop new techniques that improve localization and mapping accuracy in C-SLAM within this type of difficult environment.

A detailed breakdown of the error distribution per robot is presented in Fig. 8. Notably, the error for Robot 1 is significantly lower than that of Robots 2 and 3. We hypothesize that this variation is due to differences in robot design. While Robot 1 features large wheels and suspension, which is well suited for navigation on sandy terrain with numerous small rocks, Robots 2 and 3 are equipped with tracks, which are less effective in these conditions. To verify this hypothesis, we analyzed the linear acceleration data from the IMUs embedded in the 3-D LiDAR sensors of each robot, as shown in Fig. 9. This revealed that Robots 2 and 3 experienced significantly higher vibration levels compared to Robot 1, indicating that the wheeled configuration of Robot 1 is more compatible with the challenging terrain.

These increased vibrations in Robots 2 and 3 likely led to reduced LiDAR odometry accuracy as it may cause the loss of points and hinder data association [57]. To confirm this, we measured the ATE for the odometry of each robot individually: Robot 1 exhibited an odometry ATE of 2.45 ± 1.14 m, while Robots 2 and 3 had higher ATEs of 4.29 ± 1.76 and 3.61 ± 1.72 m, respectively.

Despite the wheeled robot's better performance in terms of reduced error, it faced its own challenges. Robot 1, with its large wheels, was more susceptible to becoming stuck, especially in wet sand, whereas the track-equipped Robots 2 and 3 demonstrated superior traction and reliability.

V. RESOURCE EFFICIENCY

This section delves into the resource efficiency of our decentralized C-SLAM solution when deployed in planetary analogue environments. Decentralized C-SLAM systems must operate within the constraints of available computing power, memory, and inter-robot communication resources, requiring strategic tradeoffs to ensure effective performance under these limited conditions.

In Section V-A, we analyze the inter-robot communication metrics gathered during our experiments and compare them to the default requirements of our C-SLAM approach. This analysis provides a critical evaluation of the realism and feasibility of our solution in practical scenarios, assessing whether the system's communication demands are compatible with the actual network conditions experienced in the field.

Section V-B addresses the calibration of key parameters to tune the C-SLAM system for real-world deployments. We explore the tradeoffs between map accuracy and available communication bandwidth, illustrating how adjusting these parameters can influence the system's overall performance. A clear understanding of these tradeoffs is essential, as it

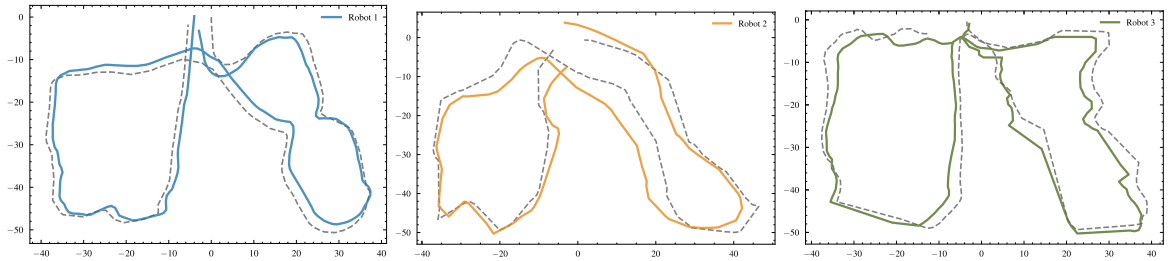

FIGURE 7. Swarm-SLAM individual robot pose graph estimates and respective GPS ground truth. (a) Robot 1 estimates. (b) Robot 2 estimates. (c) Robot 3 estimates.

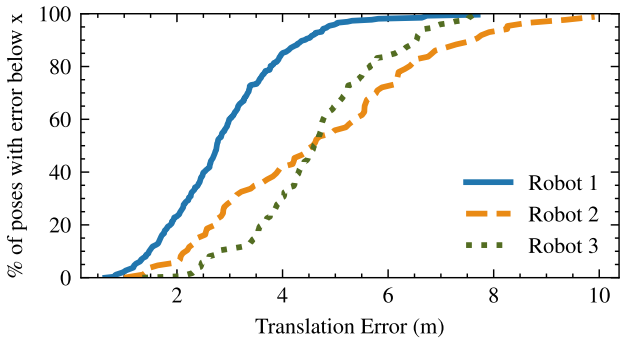

FIGURE 8. Absolute translation error distribution. On the *y*-axis, we report the percentage of poses with absolute translation error below the value on the *x*-axis.

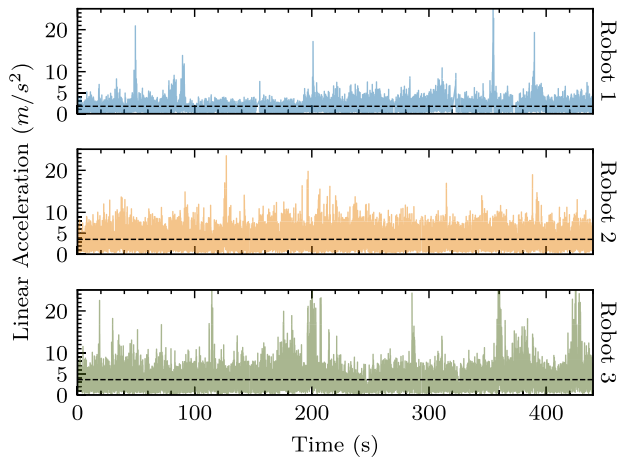

FIGURE 9. Linear acceleration magnitude from the onboard LiDAR IMUs. We removed the IMU bias and show the average as a dotted line for better comparison. We can see that Robot 1 (on wheels) was less affected by vibrations than Robots 2 and 3 (on tracks).

enables more informed decisions regarding the suitability of the approach for different mission scenarios.

Moreover, well-defined tradeoffs significantly enhance the tunability of the system. A solution that is easier to calibrate not only streamlines deployment but also increases the likelihood of adoption, particularly by users who are not SLAM

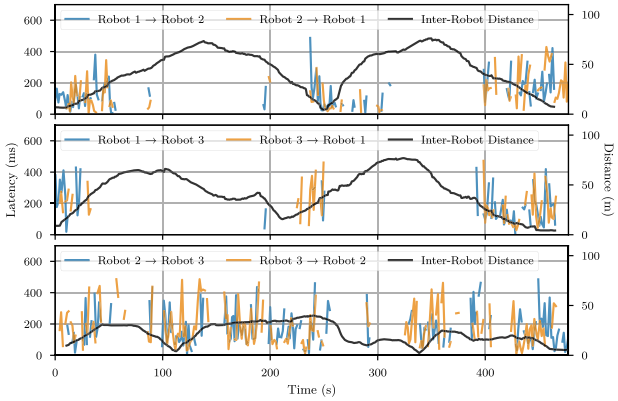

FIGURE 10. Inter-robot latency measurements between each pair of robots during the field mission. The plot shows latency values in both directions for all robot pairs, alongside the inter-robot distance, illustrating the variability in communication delays and the impact of inter-robot distance on latency. Higher latency values highlight the challenges of maintaining real-time communication in decentralized C-SLAM deployments.

experts. By simplifying the tuning process, the technology becomes more accessible and adaptable, making it a practical choice for a wider range of applications and user groups.

A. AD HOC INTER-ROBOT COMMUNICATION

We analyzed the available pairwise peer-to-peer inter-robot communication bandwidth during our field mission, as peer-to-peer communication is the backbone of scalable and resilient multirobot operations. A system that relies primarily on local inter-agent data transmission can better scale to large groups of robots because it avoids the need for a central communication node, which can become a bottleneck or critical point of failure [58]. To demonstrate the capabilities of our system, we present latency measurements in both directions for each of the three robot pairs in Fig. 10. The observed latencies range from 100 to 400 ms, which, even without considering computation time, imposes significant constraints on real-time C-SLAM deployment.

To address this challenge, our approach maintains a real-time, local single-robot SLAM estimates, which are

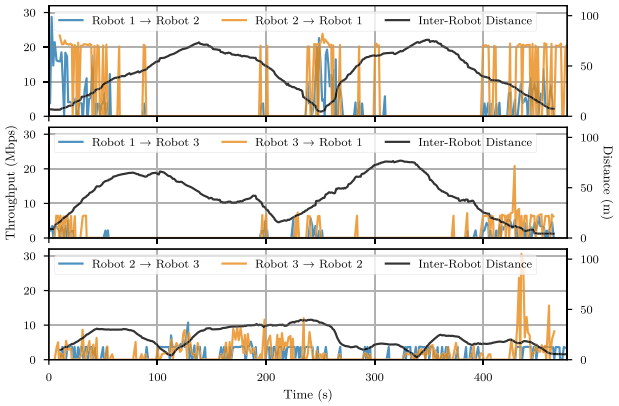

FIGURE 11. Inter-robot throughput between each pair of robots during the field mission, measured in both directions. The plot shows data transmission rates ranging from approximately 5 to 20 Mb/s when the robots are within communication range, underscoring the tight communication constraints in decentralized C-SLAM systems.

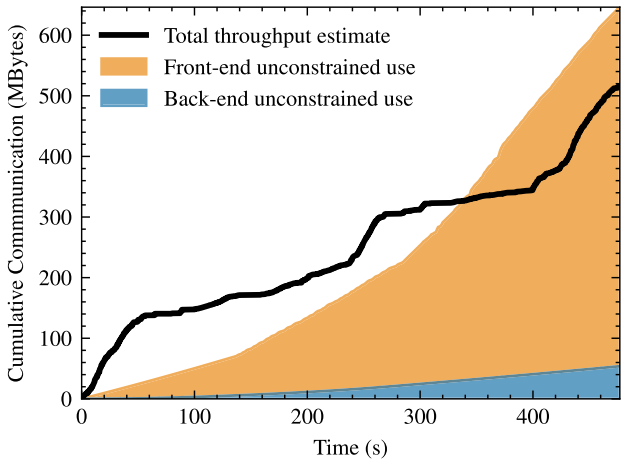

FIGURE 12. Comparison of communication usage versus total available throughput over time during the field mission. The plot shows the cumulative communication bandwidth utilized by the C-SLAM system against the measured total throughput, highlighting how communication demands increase and eventually exceed available capacity.

periodically updated and corrected using the multirobot estimates that incorporate the multirobot pose graph and inter-robot loop closures. Additionally, Fig. 10 shows the distance between robots at each timestep, calculated using GPS data. The comparison between latency and inter-robot distance indicates that our networking setup tends to lose connectivity when robots are approximately 40 m or more apart. This finding highlights the need for approaches that can handle disconnections and effectively recompute a consistent map when robots reconnect.

In Fig. 11, we also present the throughput between robots in both directions for each pair, with values ranging from approximately 5–20 Mb/s when connected. Comparing throughput with inter-robot distances confirms the consistency of our latency estimates with the throughput data.

Using these throughput estimates, Fig. 12 illustrates the accumulated communication throughput over time across all robot pairs, plotting the cumulative available communication bandwidth in megabytes at each timestep. We compared this field-measured bandwidth with the unconstrained bandwidth usage of Swarm-SLAM. To accurately measure the unconstrained bandwidth usage or maximum bandwidth consumption, we ran three agents in parallel on a single machine, each processing sensor data from one robot in our dataset. We measured all data transmission through the ROS 2 nodes of different agents, distinguishing between back-end and front-end processes to better identify their relative bandwidth requirements. Our findings indicate that front-end processes dominate the overall communication load. A key insight from Fig. 12 is that while available bandwidth is initially sufficient, communication demands rapidly increase, eventually exceeding the available capacity. This is attributed to the initially small individual robot maps with limited overlap, resulting in minimal need for resource-intensive 3-D registration. However, as each robot explores and expands its map, the level of overlap with other robots increases. Subsequently, the number of successful place recognition matches grows, leading to a rise in communication requirements. Interestingly, this shows that although increased map overlap will ultimately enhance map merging accuracy, it also demands more communication and computation to be effectively processed. Therefore, our experiments represent a challenging scenario in terms of inter-robot bandwidth due to the substantial overlap between robot trajectories and maps, making this a valuable case study for understanding the communication limits of C-SLAM systems.

It is important to note that the results in Fig. 12 were obtained using default parameters: a ScanContext cosine similarity threshold of 0.7 and a minimum of 80 inlier points for registration. As will be discussed in Section V-B, these settings generate numerous loop closure candidates, including incorrect ones, and may not be the most communication-efficient. We will explore how strategic tuning of these parameters can enhance communication performance without compromising the accuracy of the C-SLAM solution.

B. IMPACTS OF C-SLAM CALIBRATION

Calibrating key parameters of decentralized C-SLAM systems is crucial for meeting communication constraints, which is especially important in resource-limited environments like planetary analogue settings.

1) COMMUNICATION BUDGET

In Swarm-SLAM [6], we introduced a communication budget, defined as the number of inter-robot loop closure matches selected from all candidate matches identified through place recognition. This budget uses a spectral sparsification approach to prioritize candidate matches before they are

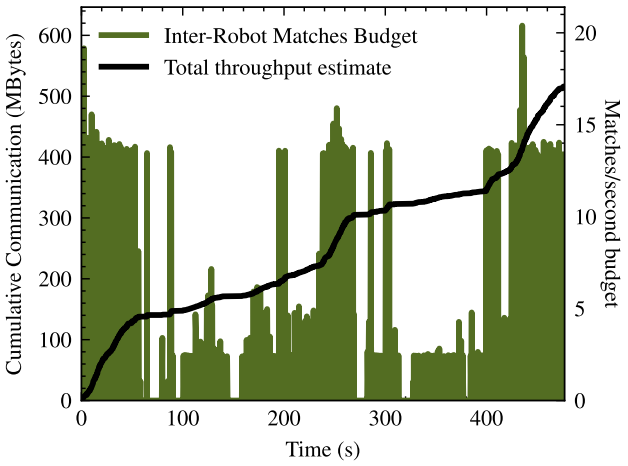

FIGURE 13. **Communication budget versus total available throughput during the field mission. The plot illustrates the relationship between the communication budget—defined as the number of prioritized inter-robot loop closures—and the cumulative available throughput.**

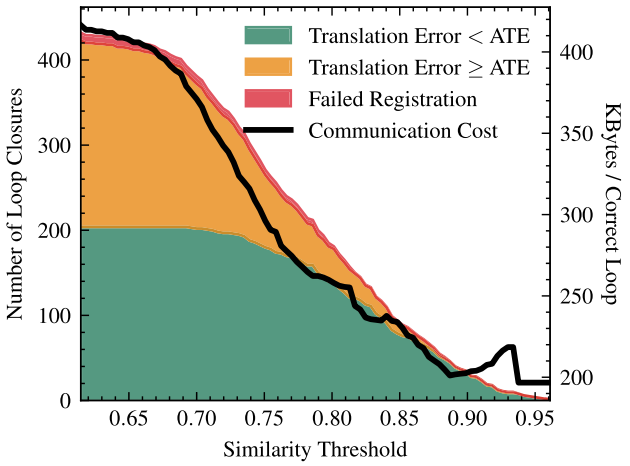

FIGURE 14. **Number of loop closures versus place recognition similarity threshold. The plot categorizes loop closures into correct, incorrect, and failed registrations, showing how varying the similarity threshold impacts the quantity and quality of loop closures. This highlights the tradeoffs involved in setting similarity thresholds for optimal performance.**

sent to the more communication and computation-intensive, 3-D registration step, as shown in Fig. 2.

The prioritization process focuses on selecting matches that are most likely to improve the accuracy of the multirobot pose graph, making it particularly valuable when high map overlap generates an excess of place recognition matches beyond available resources. It is also useful when robots reconnect after extended periods of disconnection, during which they accumulate a backlog of place recognition matches that could take significant time and communication to process entirely. By carefully prioritizing which matches to process, we can improve the tradeoff between communication and accuracy, allowing a small number of well-chosen loop closures to closely approximate the optimal C-SLAM solution.

Therefore, the communication budget directly controls how much data Swarm-SLAM transmits at each timestep. In Fig. 13, we show the optimal match selection budget at each timestep versus the cumulative throughput. In practice, this budget is often set to a fixed value, but our results suggest that adapting the budget dynamically could better utilize available bandwidth. However, evaluating throughput online during experiments poses a challenge as most estimation techniques require sending large volumes of data to test the limits of the network, which could interfere with ongoing communication.

2) PLACE RECOGNITION

In Fig. 14, we explore the relationship between the number of loop closures (y-axis) and the place recognition similarity threshold (x-axis). We categorize loop closures into three groups: correct loop closures (in green) with translation errors below the average error of the optimized multirobot pose graph (i.e., 3.74 m), less accurate or incorrect loop closures

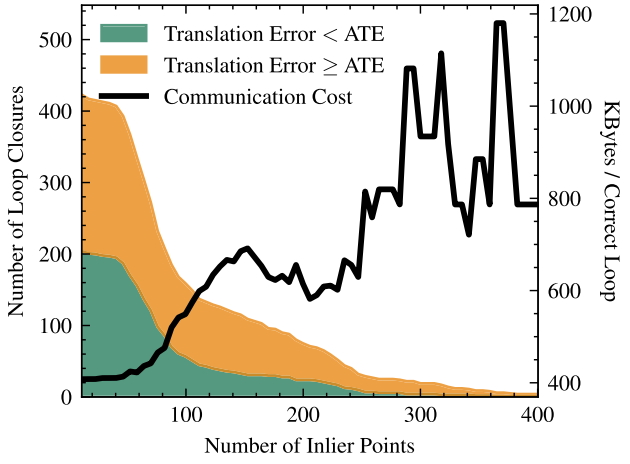

FIGURE 15. **Number of loop closures versus the number of inlier points during 3-D registration. The plot distinguishes between correct and incorrect loop closures, demonstrating how varying the inlier threshold affects the accuracy and quantity of loop closures.**

(in yellow) with errors above the average, and failed matches (in red) where high descriptor similarity did not result in successful registration due to insufficient inlier points. The plot demonstrates that lower, less conservative, similarity thresholds result in more loop closures, but many of these are incorrect or failed registrations. Conversely, increasing the threshold reduces incorrect matches but also significantly decreases the number of correct ones within a certain range (0.7–0.85), revealing a tradeoff between conservativeness and loop closure quantity. This gap indicates that current similarity measures, such as those used in ScanContext, cannot perfectly predict the quality of loop closures post-registration.

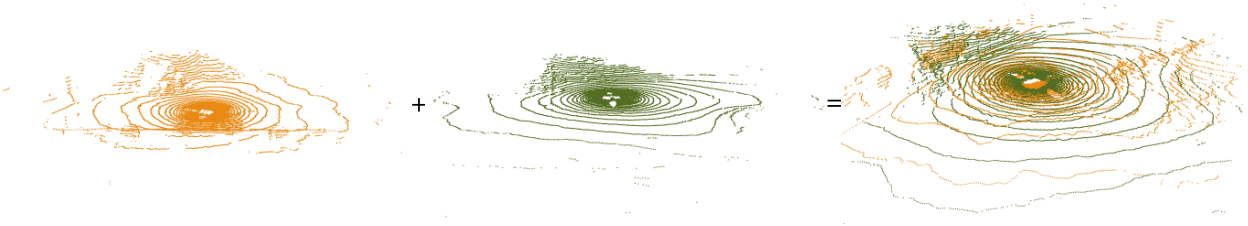

FIGURE 16. **Ambiguity in LiDAR scan matching, showing two scans with a ScanContext similarity of 0.757 and 381 inlier points during 3-D registration. Despite the high similarity and significant number of inliers, the ground-truth distance between the two scans is 17.28 m, illustrating the challenge of accurately distinguishing distinct locations in feature-sparse, flat terrain environments. This example highlights the limitations of current place recognition metrics in differentiating between similar but distinct places.**

To isolate the effect of the similarity threshold, we used a very loose threshold of only ten inlier points for subsequent 3-D registration. Although robust pose graph optimizers like GNC [24] can tolerate some incorrect or outlier loop closures, they incur significant computational costs, often requiring several seconds compared to milliseconds for standard optimizers. Reducing the number of incorrect loop closures through better front-end calibration could thus lead to notable efficiency gains in back-end optimization.

In Fig. 14, the right y-axis shows the communication cost in terms of kilobytes per correct loop closure. Generally, more conservative similarity thresholds lead to better communication cost due to fewer incorrect loop closures. However, while conservative thresholds may work well in high-overlap scenarios like our experiments, they risk missing inter-robot loop closures in environments with less map overlap, where loop closures are more rare. Thus, in low-overlap scenarios, it may be advisable to use less conservative thresholds in order to perform map merging and achieve a C-SLAM solution.

Our analysis of the relationship between similarity thresholds and the balance of correct and incorrect loop closures extends beyond our technique. It is relevant to many C-SLAM frameworks [10], [25] that employ a two-stage loop detection approach—place recognition followed by registration—to improve the efficiency.

3) REGISTRATION

In Fig. 15, we examine the number of loop closures (correct in green and incorrect in yellow) relative to the number of inlier points during registration, using a very low similarity threshold of 0.1 to focus on the number of inliers effect. The results show that, again, there is a tradeoff between setting more conservative thresholds and the total number of loop closures. Unfortunately, this parameter alone is not a reliable predictor of loop closure accuracy, as some loop closures with over 300 inliers were still incorrect. Consequently, more conservative thresholds can worsen communication efficiency, as they reduce correct loops without effectively filtering out incorrect ones.

As shown in Figs. 14 and 15, our experiments indicate that planetary analogue environments are prone to place recognition outliers and inaccurate 3-D registrations. The flat terrain and lack of distinctive features often cause different places to appear similar. In Fig. 16, we illustrate this phenomenon with two point clouds with a high ScanContext similarity of 0.757 and a substantial number of inliers (381), which, despite looking similar, represent distinct locations 17.28 m apart. Importantly, as we have shown, these outliers not only affect overall accuracy but also negatively impact the communication cost.

VI. CONCLUSION AND OPEN CHALLENGES

In this article, we presented a comprehensive evaluation of decentralized C-SLAM in planetary analogue environments, addressing the unique challenges posed by difficult terrain, limited resources, and the need for efficient communication strategies. Our experiments highlighted several critical insights into deploying decentralized C-SLAM in such challenging settings, where the terrain affects robot mobility and sensor data quality due to vibrations and uneven surfaces. These conditions underscore the need for robust, adaptable SLAM algorithms capable of maintaining accuracy in uncontrolled environments.

One of the primary challenges identified is the constraint imposed by limited resources—namely, communication bandwidth, computational power, and memory. Effective operation in such environments demands careful and adaptive tuning of system parameters, with a significant emphasis on optimizing communication, which frequently emerges as the most limiting factor. Our findings show that the C-SLAM front end consumes the bulk of the communication bandwidth, underscoring the need for future research to reduce its demands. Potential strategies include techniques such as advanced point cloud compression [59], descriptor-based representations [60], and voxelization [61] could be readily integrated into our approach. However, tuning lossy data compression in feature-sparse environments, such as the one in our dataset, presents challenges—important salient features could be inadvertently discarded, increasing the risk of perceptual aliasing. Alternative approaches, such as semantic-level descriptors [62], could further reduce communication overhead but would require a fundamental system redesign. These methods also come with their own

limitations, such as reliance on the presence of semantically rich features like man-made objects, which may not always be available in the environment.

Moreover, our study revealed a persistent tradeoff between communication and accuracy in current C-SLAM approaches. Enhancing this tradeoff remains an open challenge, with future research potentially focusing on improving accuracy without proportionately increasing communication and computational demands. This could be achieved by refining existing algorithms, exploring novel sensing and mapping paradigms, or developing more efficient data fusion techniques.

As the number of robots in a C-SLAM system increases, so does the complexity of data exchange. More robots can generate additional inter-robot loop closures, which could enhance mapping accuracy. However, this also introduces significant bandwidth contention, as multiple robots attempt to share data within the same network. Our decentralized approach offers a promising pathway for scalability, as it reduces reliance on a central server and allows robots to make localized decisions. However, to maintain communication feasibility in large-scale swarms, hierarchical strategies will likely be necessary. For instance, clustering robots into subnetworks or prioritizing loop closure transmissions based on relevance and certainty could help mitigate bandwidth congestion while preserving accuracy gains.

To advance C-SLAM research, we provide a dataset collected during our field experiments, which includes inter-robot latency and throughput monitoring over our ad hoc network. This dataset can be used to simulate or model realistic channel conditions. For example, one could integrate measured latency and bandwidth traces into a simulator or offline approach, gating data transmissions based on real bandwidth constraints. Additionally, we have included example scripts for communication modeling, particularly for analyzing latency and throughput as functions of inter-robot distance. We hope that this dataset will support other research groups in designing novel communication-aware C-SLAM approaches.

Our Swarm-SLAM approach is intentionally designed to be general and applicable across a wide range of scenarios, beyond just planetary analogue environments. It relies on inexpensive onboard sensors and simple peer-to-peer communication links, making it particularly suitable for early space missions where permanent networking or localization infrastructure—such as satellites or base stations—has yet to be established. However, the integration of infrastructure like orbital satellites or base stations with long-range, high-power networking and localization capabilities could significantly enhance SLAM performance by providing external or global sensing for the entire group of robots.

We believe that, in the future, the most effective approaches for space exploration will involve a hybrid strategy that fuses local sensing and estimation with global sensing capabilities. This fusion would ensure safe and reliable autonomy through local sensing and allow for decentralized inter-robot mapping during periods of communication loss or base station outages while benefiting from external sensing and larger computing resources when available. This balanced integration of local autonomy and global coordination is key to overcome the unique challenges of operating in extraterrestrial environments.

Ultimately, advancing C-SLAM technology for space exploration will require continuous refinement of these adaptive strategies to meet the evolving demands of complex and resource-constrained environments. By enhancing our understanding of the tradeoffs between communication, computation, and accuracy, we can better equip multirobot systems to navigate and map new frontiers—whether on Earth, the Moon, Mars, or beyond.

REFERENCES

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

[2] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *J. Field Robot.*, vol. 33, no. 1, pp. 3–46, Jan. 2016.

[3] P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, "Towards collaborative simultaneous localization and mapping: A survey of the current research landscape," *Field Robot.*, vol. 2, pp. 971–1000, May 2022.

[4] L. A. A. Andersson and J. Nygards, "C-SAM: Multi-robot SLAM using square root information smoothing," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 2798–2805.

[5] P. Schmuck and M. Chli, "CCM–SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *J. Field Robot.*, vol. 36, no. 4, pp. 763–781, Jun. 2019.

[6] P.-Y. Lajoie and G. Beltrame, "Swarm-SLAM: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 475–482, Jan. 2024.

[7] (Aug. 2021). *Canadian Space Agency Analogue Terrain*. [Online]. Available: https://www.asc-csa.gc.ca/eng/laboratories-and-warehouse/analogue-terrain.asp

[8] P.-Y. Lajoie, "Collaborative simultaneous localization and mapping dataset on Mars analogue terrain with inter-robot communication estimates," Polytechnique Montreal, Montreal, QC, Canada, Tech. Rep., Dec. 2024.

[9] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," 2021, *arXiv:2109.12343*.

[10] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2466–2473.

[11] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "DOOR-SLAM: Distributed, online, and outlier resilient SLAM for robotic teams," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1656–1663, Apr. 2020.

[12] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 4878–4888.

[13] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4802–4809.

[14] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[15] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *Int. J. Robot. Res.*, vol. 36, no. 12, pp. 1286–1311, Oct. 2017.

[16] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 2137–2156, Dec. 2021.

[17] R. Murai, J. Ortiz, S. Saeedi, P. H. J. Kelly, and A. J. Davison, "A robot Web for distributed many-device localization," *IEEE Trans. Robot.*, vol. 40, pp. 121–138, 2024.
[18] T. Fan and T. D. Murphey, "Majorization minimization methods for distributed pose graph optimization," *IEEE Trans. Robot.*, vol. 40, pp. 22–42, 2024.
[19] D. McGann and M. Kaess, "iMESA: Incremental distributed optimization for collaborative simultaneous localization and mapping," in *Robotics: Science Systems XX*. Los Altos Hills, CA, USA: Robotics: Science and Systems Foundation, Jul. 2024.
[20] P.-Y. Lajoie, S. Hu, G. Beltrame, and L. Carlone, "Modeling perceptual aliasing in SLAM via discrete–continuous graphical models," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1232–1239, Apr. 2019.
[21] K. Ebadi et al., "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 80–86.
[22] Y. Tian et al., "Search and rescue under the forest canopy using multiple UAVs," *Int. J. Robot. Res.*, vol. 39, nos. 10–11, pp. 1201–1221, Sep. 2020.
[23] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2916–2923.
[24] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1127–1134, Apr. 2020.
[25] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic SLAM for multi-robot systems," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2022–2038, Aug. 2022.
[26] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4937–4946.
[27] K. Ebadi and A. Agha-mohammadi, "Rover localization in Mars helicopter aerial maps: Experimental results in a mars-analogue environment," in *Proc. Int. Symp.*, Jan. 2020, pp. 72–84.
[28] P. Newman et al., "Navigating, recognizing and describing urban spaces with vision and lasers," *Int. J. Robot. Res.*, vol. 28, nos. 11–12, pp. 1406–1433, Nov. 2009.
[29] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5692–5698.
[30] Y. Huang, T. Shan, F. Chen, and B. Englot, "DiSCo-SLAM: Distributed scan context-enabled multi-robot LiDAR SLAM with two-stage global-local graph optimization," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1150–1157, Apr. 2022.
[31] X. Liu et al., "SlideSLAM: Sparse, lightweight, decentralized metric-semantic SLAM for multi-robot navigation," 2024, *arXiv:2406.17249*.
[32] P. Schmuck, T. Ziegler, M. Karrer, J. Perraudin, and M. Chli, "COVINS: Visual-inertial SLAM for centralized collaboration," in *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2021, pp. 171–176.
[33] Y. Chang et al., "LAMP 2.0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9175–9182, Oct. 2022.
[34] A. Cramariuc et al., "Maplab 2.0—A modular and multi-modal mapping framework," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 520–527, Feb. 2023.
[35] M. J. Schuster et al., "The ARCHES space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5315–5322, Oct. 2020.
[36] M. J. Schuster, K. Schmid, C. Brand, and M. Beetz, "Distributed stereo vision-based 6D localization and mapping for multi-robot teams," *J. Field Robot.*, vol. 36, no. 2, pp. 305–332, Mar. 2019.
[37] W. Sheng, Q. Wang, Q. Yang, and S. Zhu, "Minimizing data exchange in ad hoc multi-robot networks," in *Proc. 12th Int. Conf. Adv. Robot.*, Jul. 2005, pp. 811–816.
[38] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, "SOUL: Data sharing for robot swarms," *Auto. Robots*, vol. 44, nos. 3–4, pp. 377–394, Mar. 2020.
[39] C. Ghedini, C. H. C. Ribeiro, and L. Sabattini, "Toward efficient adaptive ad-hoc multi-robot network topologies," *Ad Hoc Netw.*, vol. 74, pp. 57–70, May 2018.
[40] L. Siligardi et al., "Robust area coverage with connectivity maintenance," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2202–2208.
[41] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, "Swarm relays: Distributed self-healing ground-and-air connectivity chains," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5347–5354, Oct. 2020.
[42] M. Giamou, K. Khosoussi, and J. P. How, "Talk resource-efficiently to me: Optimal communication planning for distributed loop closure detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3841–3848.
[43] Y. Tian, K. Khosoussi, M. Giamou, J. How, and J. Kelly, "Near-optimal budgeted data exchange for distributed loop closure detection," in *Robotics: Science and Systems XIV*. Los Altos Hills, CA, USA: Robotics: Science and Systems Foundation, Jun. 2018.
[44] Y. Tian et al., "Resilient and distributed multi-robot visual SLAM: Datasets, experiments, and lessons learned," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 11027–11034.
[45] K. Y. Leung, Y. Halpern, T. D. Barfoot, and H. H. Liu, "The UTIAS multi-robot cooperative localization and mapping dataset," *Int. J. Robot. Res.*, vol. 30, no. 8, pp. 969–974, Jul. 2011.
[46] R. Dubois, A. Eudes, and V. Frémont, "AirMuseum: A heterogeneous multi-robot dataset for stereo-visual and inertial simultaneous localization and mapping," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2020, pp. 166–172.
[47] Y. Zhu, Y. Kong, Y. Jie, S. Xu, and H. Cheng, "GRACO: A multimodal dataset for ground and aerial cooperative localization and mapping," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 966–973, Feb. 2023.
[48] D. Feng et al., "S3E: A multi-robot multimodal dataset for collaborative SLAM," *IEEE Robot. Autom. Lett.*, vol. 9, no. 12, pp. 11401–11408, Dec. 2024.
[49] H. Park, I. Lee, M. Kim, H. Park, and K. Joo, "A benchmark dataset for collaborative SLAM in service environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 12, pp. 11337–11344, Dec. 2024.
[50] S. Zhao et al., "SubT-MRS dataset: Pushing SLAM towards all-weather environments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 22647–22657.
[51] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.
[52] F. Fainelli, "The openwrt embedded development framework," in *Proc. Free Open Source Softw. Developers Eur. Meeting*, 2008, p. 106.
[53] A. Corsaro et al., "Zenoh: Unifying communication, storage and computation from the cloud to the microcontroller," in *Proc. 26th Euromicro Conf. Digit. Syst. Design (DSD)*, Sep. 2023, pp. 422–428.
[54] J. Dungan, S. Elliot, B. A. Mah, J. Poskanzer, and P. Kaustubh. *IPerf—The TCP, UDP and SCTP Network Bandwidth Measurement Tool*. Accessed: Jul. 2025. [Online]. Available: https://iperf.fr/
[55] R. Schemers. *FPing*. Accessed: Jul. 2025. [Online]. Available: https://fping.org/
[56] M. Grupp, "Evo: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo
[57] B. Schlager, T. Goelles, M. Behmer, S. Muckenhuber, J. Payer, and D. Watzenig, "Automotive LiDAR and vibration: Resonance, inertial measurement unit, and effects on the point cloud," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 426–434, 2022.
[58] M. Kegeleirs, G. Grisetti, and M. Birattari, "Swarm SLAM: Challenges and perspectives," *Frontiers Robot. AI*, vol. 8, Mar. 2021, Art. no. 618268.
[59] H. Houshiar and A. Nuchter, "3D point cloud compression using conventional image compression for efficient data transmission," in *Proc. 25th Int. Conf. Inf., Commun. Autom. Technol. (ICAT)*, Oct. 2015, pp. 1–8.
[60] X.-F. Han, Z.-A. Feng, S.-J. Sun, and G.-Q. Xiao, "3D point cloud descriptors: State-of-the-art," *Artif. Intell. Rev.*, vol. 56, no. 10, pp. 12033–12083, Oct. 2023.
[61] Y. Xu, X. Tong, and U. Stilla, "Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry," *Autom. Construction*, vol. 126, Jun. 2021, Art. no. 103675.
[62] Y. Chang, N. Hughes, A. Ray, and L. Carlone, "Hydra-multi: Collaborative online construction of 3D scene graphs with multi-robot teams," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 10995–11002.