Journal of Intelligent & Robotic Systems (2024) 110:22
https://doi.org/10.1007/s10846-023-02045-4

REGULAR PAPER

Communication-Aware Control of Large Data Transmissions via Centralized Cognition and 5G Networks for Multi-Robot Map merging

Gerasimos Damigos[1,2] · Nikolaos Stathoulopoulos[2] · Anton Koval[2] · Tore Lindgren[1] · George Nikolakopoulos[2]

Received: 5 May 2023 / Accepted: 16 December 2023

Abstract

Multiple modern robotic applications benefit from centralized cognition and processing schemes. However, modern equipped robotic platforms can output a large amount of data, which may exceed the capabilities of modern wireless communication systems if all data is transmitted without further consideration. This research presents a multi-agent, centralized, and real-time 3D point cloud map merging scheme for ceaselessly connected robotic agents. Centralized architectures enable mission awareness to all agents at all times, making tasks such as search and rescue more effective. The centralized component is placed on an edge server, ensuring low communication latency, while all agents access the server utilizing a fifth-generation (5G) network. In addition, the proposed solution introduces a communication-aware control function that regulates the transmissions of map instances to prevent the creation of significant data congestion and communication latencies as well as address conditions where the robotic agents traverse in limited to no coverage areas. The presented framework is agnostic of the used localization and mapping procedure, while it utilizes the full power of an edge server. Finally, the efficiency of the novel established framework is being experimentally validated based on multiple scenarios.

Keywords 5G · Edge · Map merging · Multi-agent

1 Introduction

Autonomous robots and vehicles have emerged as a vastly growing field in recent years, and they are deployed in various terrains, spanning from meticulously designed and known environments, such as factories, to moderately or entirely unknown environments, such as forests, roads, or other regions of interest; some examples could be found in [1–3]. All of the above scenarios require accurate and efficient localization, as well as environment and situational awareness. Without GNSS localization, robots have to localize and orient on individually produced maps. Such procedures are challenging to succeed in large-scale scenarios, hence, the collaboration of multiple robots or multiple robotic systems (MRS) is usually preferred to increase situational and environmental awareness; where in [4] a comprehensive example is being presented. Search and rescue missions are a critical paradigm of the above-mentioned considerations. Areas that have been affected by natural disasters are considered dangerous and often contain unstable structures and modified environment that makes precise GNSS localization over available maps uncertain. Additionally, indoor and outdoor search and rescue missions are of major importance. Commonly enough, robotic agents are used to localize and identify objects of interest in challenging environments [5].

✉ Gerasimos Damigos
gerasimos.damigos@ericsson.com;
gerasimos.1.damigos@associated.ltu.se

Nikolaos Stathoulopoulos
niksta@ltu.se

Anton Koval
antkov@ltu.se

Tore Lindgren
tore.lindgren@ericsson.com

George Nikolakopoulos
geonik@ltu.se

[1] Ericsson Research, Ericsson, Laboratoriegränd 11, Luleå 977 53, Norrbotten, Sweden

[2] Robotics and AI Group, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå 971 87, Norrbotten, Sweden

Published online: 25 January 2024

It is in general evident that MRS can benefit from a constant communication, while centralized processing units provide MRS with increased situational awareness and an increased understanding of the global frame, i.e. a global environment makes for a seamless localization of objects of interest [6, 7]. This research considers the definition of a global map that describes numerous agents to maximize every agent's situational awareness. To do so, the decoupling from a monolithic and isolated localization and mapping methodology for each agent should occur. This article proposes the utilization of a constant communication of multiple robots over a 5G network. More specifically, 5G-enabled robots operating within radio coverage are exploiting the available 5G infrastructure and an edge server, in order to establish a constant communication low-latency link and increase their situational awareness. While doing so, they share local environment information to an external virtual agent that is located in the edge server provided by the cellular network. Figure 1 illustrates the high-level components of the proposed architecture.

When establishing a constant communication between the connected robots, the potential benefits of the centralized architecture are evident and highlighted in the known literature, e.g. [8, 9], however, many challenges also appear in such networked architectures, while this research considers two main challenges for the considered scenario [10, 11]. The first challenge refers to the adjustment of the amount of data to be transmitted between multiple robots to fit the capabilities of the communication system, and the second one refers to the reliability of the individual autonomous robots. In terms of the former, modern robotic platforms that can support reliable autonomous missions must be outfitted with a variety of modern sensors, e.g. RGB-D cameras, 2D or 3D LiDARs, radars, and so on. Such sensors produce large amounts of data that can overcome the capabilities of state-of-the-art communication schemes [12]. Therefore, an intelligent mechanism for controlling the required data is essential to prevent that the communication link becomes the bottleneck that limits the performance. Concerning the second challenge, the robotic agents should benefit from the shared information without being dependent on it. Thus, the distribution of global awareness should be designed to provide additional information but, in case of a failure, it should not intervene with the individual robot's mission.

To that end, the authors want to highlight that the presented research does not target a collaborative SLAM solution. However, SLAM is being employed in each agent and consequently the solution specifically targets the real-time distribution and map merging of the individually produced maps. Taking into account the two main challenges, this research targets the integration and the real-life evaluation of state-of-the-art technologies and proposes solutions for some of these challenges. The contributions of this research can be concluded as follows:

1. The establishment of a framework designed for 5G-enabled robots. This framework is centered around multi-agent collaboration and communication awareness, with specific focus on 3D point cloud map merging. It empowers robots to seamlessly merge maps in real-time using a centralized approach, leveraging the high-performance capabilities provided by the 5G network. The proposed solution not only enables seamless collaboration among multiple robots and is crucial for swift and accurate inte-

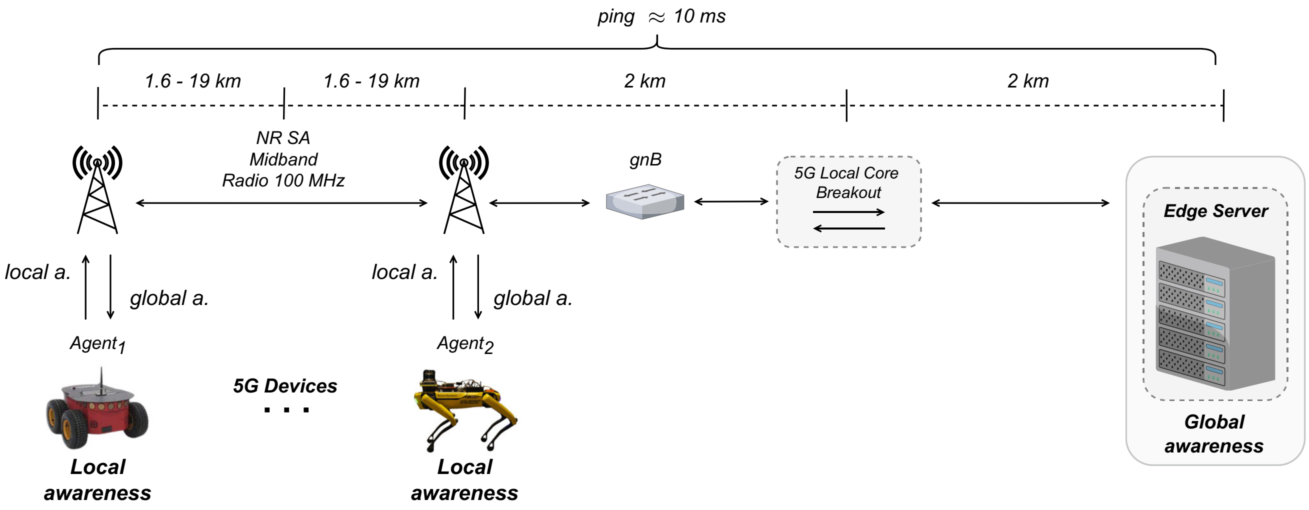


Fig. 1 Centralized communication scheme to increase situational awareness. In this concept, each agent shares local awareness information (abbreviated in the Figure as ***local a.***) to the edge server. The edge server processes and combines this information as global awareness; the centralized information is shared back to the robot-agents (abbreviated in the figure as ***global a.***)

gration of their 3D maps, facilitating advanced robotics applications in diverse fields, from autonomous navigation to complex environmental monitoring.

2. The development of a control function specifically tailored for 5G-enabled multi-robot systems. This control function is designed to manage large data transmission efficiently while considering crucial factors such as data latency and other Key Performance Indicators (KPIs) associated with radio communication. By seamlessly integrating these considerations, the control function ensures that each individual robot does not exceed its uplink throughput capabilities and data transmission within the multi-robot system is not only designed to prevent data loss but also aligns with the essential KPIs. The latter ultimately enhances the overall performance and reliability of the 5G-enabled multi-robot system in various applications.

Additionally, the field evaluation of the proposed framework is thoroughly tested on a multi-agent 5G enabled robotic system and a real-world full-scale 5G network. The field evaluation of the proposed framework enables the in-depth analysis of the proposed solution and helps the authors to identify and discuss the discovered limitations and future work. It is important to note that this is critical to the community and is intended to constitute a point of reference and evaluation, since, to the best of our knowledge there is no real-time physics simulator or emulator existing capable of capturing the real-time operation of radio-based communication phenomena along with different robot and sensor modalities. Hence, many applications do not translate to reality. Lastly, based on the observed performance metrics and the behavior of the system, a practical implementation guideline is presented.

The rest of the article is structured as such. In Section 2 the related work and some insightful comparisons are presented. Then in Section 3 the overall architecture is discussed and depicted in Fig. 2. The aforementioned section is divided into Section 3.1 that describes the map merging framework in detail, in Section 3.2 that describes the adaptive sampling parameters and to Section 3.3 that describes the communication control function. In Section 4 the experimental evaluation results are discussed, in two subsections, Sections 4.1 and 4.2. The first one presents the results from the real-world experiments and focuses on the merged maps while the latter analyzes the communication aspect of the proposed framework. In Section 5 the main limitations and challenges are discussed along with a proposed guideline for practical implementation. Finally the article is concluded with Section 6.

2 Related Work and Comparisons

In the existing literature, many MRS solutions utilize either a collaborative SLAM approach or a map merging approach as an intermediate step to achieve a certain goal [6, 13]. The common denominator often lies on the fact that a global frame of reference is needed to identify and localize objects of interest in the global frame of reference. Here we aim in considering works that are applicable in real-life scenarios, where wide-area connectivity and the corresponding said challenges arise. The considered solutions should also achieve the possibility of constructing a global map in relatively real-time scenarios so that the shared information can be utilized throughout the manifestation of the mission. Given the evident differentiation between collaborative SLAM and map merging, where SLAM algorithms persistently aim to estimate position and map the environment, while map merging focuses solely on aligning multiple local maps into a unified global representation, we opt to organize the related works into two distinct subsections, addressing

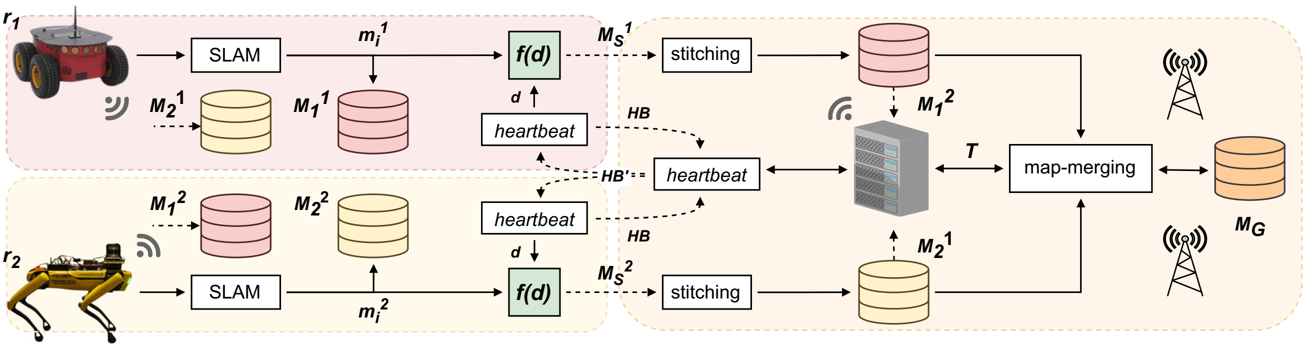


Fig. 2 The overall centralized architecture of the proposed framework. Each robot r_1 and r_2 performs SLAM individually and collects the map instances in a buffer M_n regulated by a control function $f(d)$. The map instances are transmitted to the other agents as well as the edge unit where the map merging process takes place

each aspect separately. It is worth noting that both approaches share the common objective of transforming local maps into a global frame.

2.1 Collaborative SLAM

One reasonably early characteristic example of collaborative SLAM is the CoSLAM approach [14], a vision-only collaborative SLAM solution that handles dynamic environments. CoSLAM uses powerful GPU processing to address the immersive requirements of processing data from multiple synchronized cameras. However, the authors do not elaborate on any component related to the communication and connection of the multiple camera set-up. Other works that exhibit remarkable performance, such as [15] and [16], do not consider any communication or connectivity feedback or rely on the assumption that a large amount of data could always be transmitted from the network. Frequently, this is a common practical problem that restricts the application of such techniques in realistic conditions.

The work of [17], presents a multiagent mapping approach and proposes connecting multiple vehicles utilizing cellular networks. However, they consider that GNSS information is available, which makes the solution unsuitable for GNSS-denied environments. Additionally, they do not consider network conditions or any feedback to address communication challenges on map information transmission. Due to that, the deployment of this solution will be challenging considering real-life cellular networks. Sboia et al. [7] presented a thorough solution designed to explore subterranean environments in light of intermittent communication conditions. This work consists of many parts that include the co-design of exploration and communication while mainly focusing on the mission objective, i.e. the exploration component. Nevertheless, the authors address the challenge of communicating various types of essential data (including maps), targeting the maximization of situational awareness between the multi-robot systems. The communication-related aspect targets two main objectives, the deployment of lower-level communication KPIs, like SNR, and a token bucket rate limiting algorithm to optimize the mesh network deployment and regulate data transmission. Data transmission regulation serves in many ways, e.g. congestion, available bandwidth, interference, etc. Although the authors considered network KPIs on the map information transmission, there is no application layer feedback mechanism in place in the map transmission phase. Thus, data transmission is safely performed while avoiding network overload, but does not take into account application layer KPIs and hence cannot operate in full performance. Finally, this work does not explicitly define the use of the merging of map information.

Lastly, a very comprehensive work from [13] takes a multiagent SLAM approach and intelligently accounts for the communication aspect of the solution. By doing so, this approach ensures that individual robots benefit from the collaborative scheme, and in the case of communication loss, each robot's autonomy is maintained. The proposed architecture employs a server that handles computationally intensive processes and is responsible for further data coordination tasks. An intelligent design limits the publishing rate to a maximum value for each robot. However, this solution does not consider the dynamic changes that could emerge in a public wireless network (e.g. cellular networks), thus limiting the framework's performance.

2.2 3D Map Merging

Another popular solution that targets the map merging approach appeared in [18] where a map merging framework has been developed suitable for multi-robot scenarios. This work initially preprocesses the point cloud maps to remove outliers, then performs a 3D feature extraction algorithm with SIFT points [19] or Harris corners [20] and finally compares features to find correspondences and align the two point clouds. Subsequently, the succession of this solution relies on the sensing equipment, various map characteristics, and the utilized SLAM algorithm. Finally, this work resulted in a ROS package that is commonly known and used throughout the robotics community [21]. Although the solution is validated in complex scenarios, it didn't examine the communication aspect of the problem nor the real-time applicability. The 3D feature extraction process is a time-consuming operation, resulting in processing times of 10 to 30 seconds or more, depending on the size of the maps. These amount of time can be considered substantially large, especially for aerial platforms and real-life, time-critical missions. Similarly to the aforementioned, the work in [22] proposed a 3D map merging scheme for multi-robot systems based on overlapping regions, and to be more specific, based on the SHOT descriptors [23] and the SAC-IA alignment method. Although the results were promising, it did not consider any communication aspects for a multi-agent system and from the comparisons made in our previous work [24] the hand-crafted global descriptor extraction operation results in processing times of 100 to 260 seconds, making it unsuitable for any real-time application.

A learning based approach in the localization and mapping problem is discussed by [25], focusing more on compressing and transmitting local maps. During this work, the authors propose a solution where the robot's surroundings are captured and decomposed in segments. Each segment is represented by a low dimensional descriptor encoding the robot's surroundings. Then, the same descriptor is fed to a neural network that reconstructs the initially observed surroundings. This approach is particularly interesting and presents an intelligent mechanism to reduce data transmission (since the data required for transmission is the lower dimensional descriptor) when the considered architecture is deployed over a wireless network. However, the authors do not present any communication aware mechanism to transmit the descriptors.

In summarizing the existing body of work and aligning it with the contributions of our proposed framework, a noticeable gap emerges in collaborative SLAM algorithms that incorporate a feedback signal and integrate KPIs related to the communication channel. The prevailing literature in map merging also falls short, lacking both speed in delivering solutions and consideration of the communication aspects inherent in exchanging maps among agents. Consequently, our contribution lies in the introduction of a communication-aware control function that seamlessly integrates with the map merging framework. This combined approach presents a comprehensive and real-time solution for the intricate task of map merging in multi-robot scenarios, addressing the identified shortcomings in the current landscape of collaborative SLAM and map merging methodologies.

3 The Proposed Architecture

This research aims to design and examine a large-scale communication-aware framework for MRS that communicates over a 5G system, while addressing communication and autonomy challenges. The proposed framework is designed, so individual agents can either perform exploration, be assisted, or be teleoperated from a control station. In this study, each agent is equipped with a 3D LiDAR scanner in order to individually localize and map its environment. The framework accounts for radio channel variations or loss in coverage by utilizing a communication-aware control function. The control function exploits a decision mechanism to decide when to transmit map instances to a centralized unit, i.e. an edge server, so that the computational intensive process of map merging can be later executed. More specifically, it accounts for the selection of the desired transmission data rate of the map instances, as well as the size of the individual map instances. Altogether, regulating the data rate throughput corresponding to the transmission of the map instances and the feasibility of the framework to perform the multi-agent map merging operation in arguably what is considered real-time. The edge server constitutes the centralized unit of the whole architecture and is responsible for maximizing the situational awareness of each individual agent, i.e. distributing local maps from a considered agent to the remaining agents.

3.1 Proposed Large-Scale Map Merging Framework

The map merging procedure, described in more details in [24], considers the following problem to be solved, under the only assumption that there is a sufficient overlapping region between the local maps to be merged. Assuming a system of N robots $R = \{r_1, r_2, \ldots, r_N\}$, in $\mathbb{R}^3$ space, each robot produces a local map $M_n(t)$, with respect to its local, static map frame $\mathcal{M}_n$ and the appropriate homogeneous rigid transformation $T \in SE(3)$ is calculated, so that the local maps are merged into a single global map frame $\mathcal{M}_G$, at any given moment. The global map can be expressed as a combination of the individual local maps, correctly transformed to the global map frame, $\mathcal{M}_G$ and is defined as $M_G = \{M'_1, M'_2, ..., M'_N\}$. For simplicity, this operation is described for two agents at a time, without loss of generality since we can make the assumption that more than two maps can be merged recursively. The merging procedure can be defined as a function f_m as:

$$f_m : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3 \tag{1}$$

More precisely, the merged map is denoted as:

$$M_G = f_m(M_1, M_2) = M_1 \cup T_{12} M_2, \tag{2}$$

where $T_{12} : \mathcal{M}_2 \to \mathcal{M}_1$ represents the homogeneous rigid transformation of the special Euclidean group, denoted as:

$$T_{12} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \in SE(3), \tag{3}$$

where $R \in SO(3)$ is the rotation and $p \in \mathbb{R}^3$ is the translation.

In order to calculate the transformation T_{12} the problem is divided into two parts. First, the two overlapping regions $S_1 \subset M_1$ and $S_2 \subset M_2$ are selected by utilizing the place recognition and yaw discrepancy regression descriptors, described in more detail in 3DEG [26]. This step will provide an initial transform T_0 that will roughly align the two local frames $\mathcal{M}_1$ and $\mathcal{M}_2$ and can be described as:

$$T_0 = \begin{bmatrix} R_{\hat{z}}(\delta\theta) & p_{1,k_i} - p_{2,k_j} \\ 0 & 1 \end{bmatrix} \tag{4}$$

where $R_{\hat{z}}(\delta\theta)$ is the predicted yaw discrepancy and $p_{1,k_i} - p_{2,k_j}$ is the translation difference between the pose of r_1 at the time step k_i and the pose of r_2 at the time step k_j. The indexes k_i and k_j are selected by querying the two place recognition descriptor vector sets Q_1 and Q_2 of the robots r_1 and r_2 respectively.

$$(k_i, k_j) = \underset{(i,j) \in \mathbb{N}}{\arg\min} \, f(Q_{1,i}, Q_{2,j}) \tag{5}$$

The second and final step of the merging procedure is applying the initial transform T_0 as a prior to the `FAST-GICP` algorithm [27], providing a considerably faster convergence and the final refined alignment. The `GICP` algorithm [28] is used to estimate the transformation matrix T between two sets of points, $\mathcal{A} = \{a_0, \ldots, a_N\}$ and $\mathcal{B} = \{b_0, \ldots, b_N\}$. The error in the transformation is determined by computing the difference between the transformed point $\hat{b}_i$ and the corresponding point in $\mathcal{A}$, $\hat{a}_i$, which are both assumed to be sampled from Gaussian distributions; $a_i \sim \mathcal{N}(\hat{a}_i, C_i^A)$ and $b_i \sim \mathcal{N}(\hat{b}_i, C_i^B)$. This difference is denoted by:

$$\hat{d}_i = \hat{b}_i - T\hat{a}_i, \tag{6}$$

and its distribution is given by the Gaussian distribution with mean zero and covariance matrix:

$$d_i = \mathcal{N}(0, C_i^B + TC_i^A T^T), \tag{7}$$

according to the reproductive property of the Gaussian distribution. The `GICP` algorithm seeks to find the transformation matrix T that maximizes the logarithmic likelihood of the distribution of $\hat{d}_i$, as expressed in Eq. 7. This is achieved by minimizing the sum of the squared Mahalanobis distances between $\hat{d}_i$ and the origin, weighted by the inverse of the covariance matrix $C_i^B + TC_i^A T^T$ for each point i. The resulting equation for the transformation matrix T is given by:

$$T = \arg\min_T \sum d_i^T (C_i^B + TC_i^A T^T)^{-1} d_i \tag{8}$$

3.2 Keyframe Adaption and Map Instances

In order to acquire the aforementioned descriptor vector sets Q_n and the corresponding poses p_n, a keyframe sampling process occurs. We define a keyframe $\mathcal{K}_{n,k}$ as the pair of a point cloud scan $\mathcal{P}_{n,k}$ and the corresponding pose $p_{n,k}$ of the robot, at a time step k:

$$\mathcal{K}_{n,k} = \{\mathcal{P}_{n,k}, p_{n,k}\} \tag{9}$$

When constructing global or local graphs, keyframes are commonly used as sampling positions. In many existing works, such as those in [29–31], keyframe nodes are selected at fixed intervals, such as every 2 meters of translational displacement or every 15 degrees of rotational change. However, inspired by [32, 33], we opt to adaptively adjust the sampling distance threshold for keyframe selection based on the spaciousness of the current point cloud $\mathcal{P}_{n,k}$. This approach is motivated by the observation that in large-scale environments, the characteristics detected by the point cloud scan are typically noticeable for a longer period and can be relied upon, whereas in confined or limited spaces, it is necessary to use a lower threshold to consistently capture small-scale attributes such as tight corners. The concept of spaciousness is defined as:

$$s_k = \alpha s_{k-1} + \beta S_k \tag{10}$$

The spaciousness of the current point cloud scan $P_{r,k}$ is determined by calculating the mean Euclidean distance, S_k, from the center of the point cloud to each point. Based on this, a smoothed signal, s_k, is obtained, which defines the sampling threshold th_k for selecting keyframe nodes. The values of α and β are set to 0.9 and 0.1, respectively, to ensure appropriate smoothing of the signal.

$$th_k = \begin{cases} 10\text{m if } s_k > 10\text{m} \\ 6\text{m if } 6\text{m} < s_k \leq 10\text{m} \\ 3\text{m if } 3\text{m} < s_k \leq 6\text{m} \\ s_k \text{ if } 0\text{m} < s_k \leq 3\text{m} \end{cases} \tag{11}$$

For the MRS, as depicted in Fig. 2, a *master-slave* architecture is assumed. For every robot r_n, the produced local map $M_n(t)$ can be defined as a combination of multiple map instances $m_{n,i} \in \mathbb{R}^3$ acquired between two keyframe instances. The map instances are defined as:

$$m_{n,i} = \bigcup_{k \in \{t-1,t\}} h(\mathcal{P}_{n,k}) \tag{12}$$

where $\mathcal{P}_{n,k}$ is the current LiDAR scan and $h(\cdot)$ is the function that denotes the post-processed point cloud scans (e.g. outlier removal, registration, etc). Therefore, the local map can be described as the union between the local map of the previous time instance and the current map instance. From this, in order to avoid duplicate points, the identical points from $M_n(t-1)$ and $m_{n,t}$ are removed.

$$M_n(t) = \{M_n(t-1) \cup m_{n,t}\} \ominus \{M_n(t-1) \cap m_{n,t}\} \tag{13}$$

Since the produced map $M_n(t)$ depends on the duration and total distance of the mission, the direct transmission of a complete local map $M_n(t)$ would require the transmission of immense data structures when considering larger missions. On the other hand, $m_{n,i}$ is independent of the mission's duration or distance and the maximum size of each $m_{n,i}$ depends on the SLAM algorithm, the LiDAR sensor and the keyframe sampling threshold th_k. Thus, a map instance $m_{n,i}$ is chosen for transmission to the edge server as it is a bounded data structure defined as:

$$\frac{th_{min}}{T}[\mathcal{P}_n] \leq [m_{n,i}] \leq \frac{th_{max}}{T}[\mathcal{P}_n], \tag{14}$$

where T is the mapping period depending on the SLAM algorithm and the $[\cdot]$ symbol denotes the size of the scan in bits. The selection of $m_{n,i}$ is the first measure taken to reduce the transmitted data. The edge server collects each local map instance $m_{n,i}$, and stitches them to replicate each local map $M_n(t)$ of the considered robot r_n. Subsequently, the edge server distributes the received local map instances to the remaining agents, $R_r = R - \{r_{tr}\}$. Finally, after the succession of the merging process, the computed transformation T is distributed to the remaining robots R_r. Since throughout the mission, the edge server has distributed the local map instances $m_{n,i}$, all the agents are capable of transforming their local maps and creating the global map M_G. To do so, the transformation matrix T has to be distributed to all the respective agents R. It is important to note that the distribution of the local maps has already taken place and the map merging operation is hosted on the edge server; thus, after the successful map merging operation, the only information to be transmitted back to the agents is the produced transformation matrix. Lastly, note that the process of transforming the one map to the coordinate space of the other map is significantly lighter compared to the map merging process. The latter also justifies the placement of the map merging process on the edge server, while the transformation process can easily happen onboard the robotic agents.

3.3 Communication Aware Map Transmission

The utilized network architecture is depicted in Fig. 1. Here it is essential to highlight the main principal components of a 5G network. Generally, a 5G network consists of two main components, the 5G radio access network (RAN) and the 5G core. The RAN represents the wireless interface of a 5G network and is responsible for tasks like connecting user equipment (UEs), modulation, signal transmission and reception, managing radio resources among users, and more. The 5G core serves as the backbone of the 5G network and is highly flexible and programmable (usually runs on server clusters). Some main functionalities of the 5G core include security, packet routing, authentication, session management, network slicing, and much more. Additionally, the 5G RAN can be found in the 5G radio equipment, either indoor or outdoor, i.e. cellular base stations; while the 5G core is a centralized unit that manages multiple 5G base stations. For reference, a 5G core could potentially handle enough base stations to cover a small city [34], while a 5G base station or a 5G cell tower could cover a few city blocks and up to a few kilometers depending on the environment and the utilized radio network configuration.

One important note is that traffic is commonly routed through the 5G core (in this case the 5G local core breakout). Thus, all packets in the utilized network have to go through the 5G local breakout of the system so that they can be routed to the correct destination. Since this research focuses on the application layer of the 5G network, i.e. the 5G-connected robots, the two potential bottlenecks that we are considering refer to the uplink throughput capabilities of a 5G-connected robot. For example, if the amount of data scheduled for transmission from a robot exceeds the systems capabilities in the current radio and network conditions, a user in the application layer has to adjust its data transmission so that seamless communication is preserved.

To that end, a critical aspect of the proposed framework relies on the intelligent transmission of map instances $m_{n,i}$. The proposed solution proactively avoids excessive communication latency that may be created by extensive buffer build up on the agent when overloading the agent's uplink capabilities by adjusting the amount of data to be transmitted. The received map instances $m_{n,i}$ are necessary to create the correct replica of the robot's local map in the edge server. In case multiple instances get lost, the stitching of the local map might include incoherent regions and thus negatively affect the framework's performance. Another problem that may occur is the congestion of the network. As mentioned above, robotic platforms can output a significant amount of data; thus, in the case of multiple robots, each robotic agent should regulate the transmission of map instances. Therefore, a control mechanism that accounts for communication variations and regulates transmission is included in the proposed solution.

To regulate transmission data and account for communication variations, e.g. radio channel variations, cell load variations and others, the first expression refers to transmission data. Let

$$T_x = \sum_{i=1}^{R}[m_{n,i}], \tag{15}$$

define the amount of the transmitted data, i.e. map instances within the time interval of 1 second. Then, $[m_{n,i}]$ represents the size in bits of the corresponding map instances, and R is the frequency at which the map instances $m_{n,i}$ are being transmitted. Furthermore, $m_{n,i}$ is an element of the local map $M_n(t)$ which defines the part of the map that due to communication conditions is not yet transmitted. After the decision of the data rate transmission, the remaining map left on each agent can be defined by the following expression:

$$M_n(t+1) = M_n(t) - \bigcup_{1}^{R} m_{n,i} \tag{16}$$

where, $M_n(t+1)$ is the local map remaining on the agent, and $\bigcup_1^R m_{n,i}$ defines the map instances along with the transmission rate of the map instances. Additionally, note that the size of each map instance $[m_{n,i}]$ is determined by the operation described in Section 3.2. For the considered architecture, the maximum map instance size $[m_{n,i}]$ presents an upper limit that corresponds to the raw LiDAR scans. Therefore, seeking a mechanism regulating T_x (defined by Eq. 15), the framework considers a feedback signal corresponding to the delays captured through the *heartbeat* (`HB`) message exchange. Note that the `HB` message captures the delay expressed from the time that the map instance was created until the time that the map instance was received in the edge server. The measured delay is communicated back to the agent with the `HB'` as depicted in Fig. 3. Hence, a control function that governs the transmission rate R of T_x is defined as such:

$$R = f(d) = \begin{cases} tr_{max} & , \bar{d} < \bar{d}_{min} \\ ad + b & , \bar{d}_{min} < \bar{d} < \bar{d}_{th} \\ 0 & , d \geq \bar{d}_{th} \end{cases} \tag{17}$$

where, $a = (tr_{min} - tr_{max})/(\bar{d}_{max} - \bar{d}_{min})$, and $b = (\bar{d}_{max} tr_{max} - \bar{d}_{min} tr_{min})/(\bar{d}_{max} - \bar{d}_{min})$. Then, tr_{min} and tr_{max} constitute the minimum and maximum desired transmission rates, while tr_{max} has an upper limit defined by the SLAM algorithm as discussed in 3.2. Then, $\bar{d}_{max}$ and $\bar{d}_{min}$ constitute the maximum expected latency and the minimum expected latency respectively (captured by the heartbeat signal), while $\bar{d}_{max} = \bar{d}_{th}$ for the considered design. In determining a suitable threshold $\bar{d}_{th}$, various approaches may be pursued based on the real-time demands of the application. One method involves gathering latency statistics and choosing a threshold, such as the 95^{th} percentile. This approach aims to control map instance transmission to the edge server, preventing excessive buffer buildup that could lead to latency escalation and impact other communication flows associated with the agent. Thus, for this problem, $\bar{d}_{th}$ and $\bar{d}_{min}$ are evaluated considering the statistics of the measured expected latency under normal conditions (as described below). The statistics were captured in a probability density function (PDF). The PDF captured the latency in the `HB` signal recorded in a series of pilot experiments with the same agent and network configuration used for the evaluation of the solution. The selected threshold $\bar{d}_{th}$ is derived by the 95^{th} percentile of the said PDF. Note that to perform a coherent set of pilot experiments and latency statistic measurements, the following aspects must be met. The bit rate selected for transmission during the pilot experiments and the capture of the latency statistics for the map instances transmission should always be less than the maximum available bit rate that the current link can support. The subsequent rationale suggests that, for varying channel conditions, data transmission should always stay within the limits of the radio channel capacity. Furthermore, at each position, the transmitted data rate during pilot trials should not exceed the data rate corresponding to the radio resources allocated to the specific agent. These two statements help define the expected latency for map instance transmission in traversed areas. The goal is to gather statistics without causing cell or UE overload. Subsequently, the control function utilizes these statistics to prevent significant buffer buildup on the considered robotic agent (i.e., the UE) based on network conditions. Figure 3 illustrates the closed-loop architecture, and the proposed algorithm for each robot is presented in Algorithm 1.

Algorithm 1: The algorithm for controlling the transmission of the map instances

```
Data: $m_{n,i}, \mathcal{M}_{n,i}, \mathcal{K}_{n,i}$ ;   /* map instance, map
      instances buffer (FIFO list), keyframe
      list */
Data: $\mathcal{HB}, \mathcal{P}_k, p_k$ ;   /* heartbeat, LiDAR scan,
      pose */
Result: $\mathcal{R}$ ;                          /* rate */
while exploring do
    $\mathcal{P}_k, p_k \leftarrow$ SLAM($lidar, imu$); /* registered scan
    */
    $s_k \leftarrow$ spaciousness($\mathcal{P}_k$);
    $th_k \leftarrow$ threshold($s_i$); /* adaptive threshold
    */
    if $dist \geq th_k$ then
        $\mathcal{K}_{n,i}$.append($p_k$);     /* new keyframe */
    end
    if not new keyframe then
        $m_{n,i}$.append($\mathcal{P}_k$);
    end
    if new keyframe then
        $\mathcal{M}_{n,i}$.append($m_{n,i}$);
        $i \leftarrow i + 1$;
    end
    $\mathcal{HB} \leftarrow$echo ;          /* measure latency */
    $\mathcal{R} \leftarrow f(\mathcal{HB})$ ;        /* control function */
    $k \leftarrow k + 1$;
end
while $\mathcal{M}_{n,i}$ not empty and $\mathcal{R}$ is not zero do
    offLoadToEdge($\mathcal{M}_{n,i}$.pop($m_{n,i}$));
    sleep($1/\mathcal{R}$);
end
```

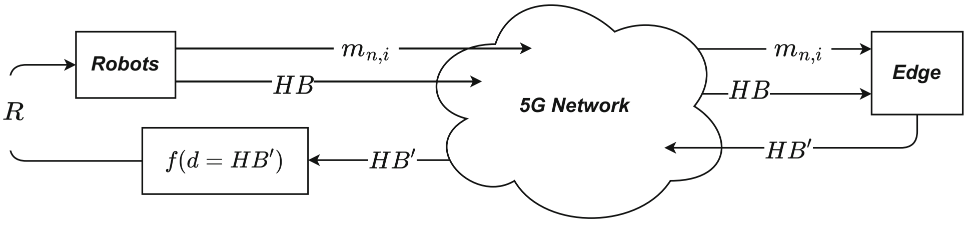


Fig. 3 Control scheme for the map instances transmission

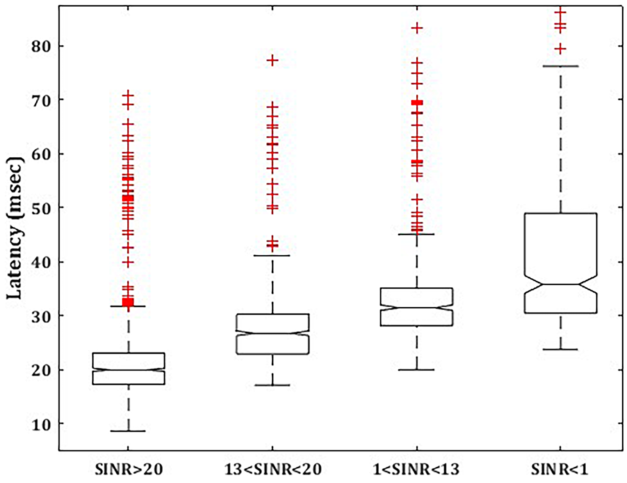
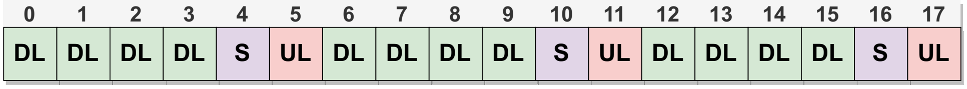


Fig. 4 The figure depicts the correlation between SINR values and the measured latency

Although the problem is quite complex, the proposed design accounts for degraded connectivity and out of coverage conditions. With the term degraded connectivity, the authors refer to circumstances in which the radio resources scheduled for the considered agent might be limited (e.g. due to scheduling algorithms and large background traffic) or the overflow of the throughput uplink capabilities for each corresponding agent [12] or to occasions that the robot might be positioned to the boundary of the cell's coverage. Furthermore, the complexity of public cellular networks in terms of the expected load and the mobile aspect of the robots may result in various combinations of the discussed key performance indicators. Ideally, the control function should be extended to depend on multiple KPIs, for example, the signal to interference noise ratio (SINR), the cell load and others, i.e. $f(latency, SINR, RSRP, cell\,load)$. During this research, the proposed solution was tested to capture correlations between the captured latency from the heartbeat signal and corresponding characteristic SINR values. Figure 4 depicts the relationship between the two KPIs. The experimental evaluation occurred in a real 5G network in indoor and outdoor scenarios. Note that the slight increase in latency corresponds to scheduling latencies when the resource utilization increases with reduced SINR. When the agent traverses to regions with decreased signal quality, for example close to the cell edge, the 5G system assigns additional radio resources to the specific agent so that the required bit rate is met. The use of the additional radio resources are captured by the slight increase in the latency of the transmitted map instances.

To further elaborate on the reasoning behind the control function design and its relationship with the corresponding 5G radio KPIs, the authors present the following reasoning. The utilized 5G network employs a mid-band deployment at 3.5 GHz with a time division multiplexing (TDD) architecture. The downlink to uplink ratio (DL/UL) is 4-to-1 and the corresponding 5G frame structure is shown in Fig. 5. In other words, the ratio defines the fraction of radio resources that are assigned for either uplink or downlink transmission. Here, there are several additional components that we have to account for to reason around the additional presented latency and its correlation to the radio KPIs. At each specific radio condition, the 5G system selects an appropriate coding scheme, or modulation code scheme (MCS) to transmit the considered data, either in the downlink or the uplink direction. Further, the amount of data that can be transmitted is partly defined by the selected MCS. Additionally, for the uplink case, since the uplink and downlink resources are disproportionately duplexed in time (according to the selected TDD pattern), longer scheduling delay will be experienced since the uplink resources are less frequently available. Finally, the radio resources are shared among all the users served by the corresponding base station. Thus, when the radio KPIs present lower values the appropriate MCS must be respected and the users or agents might have to wait for the next UL slot to transmit the complete considered data, hence the additional latency with respect to the radio KPIs.

To couple the latter analysis with the selection of the control mechanism, one has to consider that if the requirement is to control the map instances data rate (about to be transmitted over the 5G network), the dynamics of the considered system vary with respect to the agent's radio KPIs and the state of the 5G system, i.e. the load of the cell, the selected configuration, and the location of the edge server. Adaptive control schemes are more appropriate for the described problem. The map instances transmission is handled by the ROS framework, and an appropriate modeling would consider the modeling of the queue in the ROS publisher subscriber architecture. The queue can be modeled utilizing queuing theory such as M/G/1 queues, where the arrival of the data rate in the queue, i.e. the arrival of the map instances depends on the map instance generation process and the service distribution depend on all the aforesaid communication KPIs. That said, such analysis of the system is rather a future direction and out of scope for this research paper. The focus here lies on the design and implementation of a resilient framework that can handle the relatively real-time map merging operation. Thus, this paper considers a more simplistic approach where the signal responsible to capture both the radio KPIs and the latency of the system is encapsulated in the latency captured by the `HB` signal, and the corresponding performance is depicted and discussed in the following section.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
DL	DL	DL	DL	S	UL	DL	DL	DL	DL	S	UL	DL	DL	DL	DL	S	UL

Fig. 5 5G frame structure for a common 4:1 TDD DL/UL pattern. Note that with DL we refer to the downlink slots, with UL we refer to the uplink slots, and with S we refer to the special slot

4 Experimental Evaluation

For the experimental evaluation of the proposed solution, the complete architecture was tested in large scale deployment of robots in real environments at the premises of Luleå's University of Technology (LTU). Two robot agents were used to manifest the experimental evaluation, shown in Fig. 6. The two missions encompassed both indoor and outdoor terrains in their exploration paths. More specifically, in the initial mission, a segment of LTU's outdoor space and the basement corridors of the building were traversed for the outdoor and indoor paths, respectively. In the subsequent mission, the two robots navigated through the main roads in the outdoor areas of the LTU campus, broadening their exploration and covering a larger area. The utilized 5G network is a real-life innovation network that operates at the mid-band frequencies (3.4 - 3.8 GHz) and includes both an external macro base station and a Radio Dot System (RDS) installation covering most of the considered experimental area. Nevertheless, due to the indoor-outdoor character of the first mission, and the larger scale of the second mission, the experimental areas include regions with degraded coverage (depicted in Fig. 10 and discussed in Section 4.2). Regarding the two used robotic platforms, the first platform utilized was the legged quadruped platform Spot from Boston Dynamics. The Spot platform was equipped with a Velodyne Puck Hi-Res 3D LiDAR, an Intel NUC on-board computer, and a D-link 5G router. The second platform used was a Pioneer 3AT rover platform and was equipped with a Velodyne PuckLite 3D LiDAR. As for the onboard computing unit and the 5G router, the Pioneer platform was equipped with the same hardware as the Spot platform. For the indoor experiments, `DLO` [32] was utilized as the odometry and mapping source, while on the outdoor experiments, `LIO-SAM` [29] was preferred.

Section 4.1 depicts and discusses the performance of the overall proposed framework regarding the map merging, the increased situational awareness and the resilience of the system in intermittent communication conditions, while Section 4.2 focuses solely in the evaluation and analysis of the performance of the communication-aware distribution and control of the individual maps.

4.1 Overall Framework Evaluation

Experiments were carried out in realistic scenarios to evaluate the complete proposed framework. To further challenge the communication scheme, in the first experiment, the legged robot's mission includes transitioning from the outdoor to the indoor environment, while the rover's mission is contained in the indoor space. Throughout the mission, both robots connect to more than one 5G base station. One macro base station covers the outside area and a portion of the indoor area that is interconnected with the outdoor space. Subsequently, an RDS DOT system covers the majority of the indoor area. Therefore, the exploration of the complete site contains regions characterized by various communication conditions and includes handovers between network cells. Considering the setup of the described mission, it is evident that the covered areas do not provide coverage for GNSS localization. Thus, a ground truth comparison needed to be included manually. To address this and evaluate the framework's accuracy, expert researchers manually aligned the two maps and provided a ground truth transformation

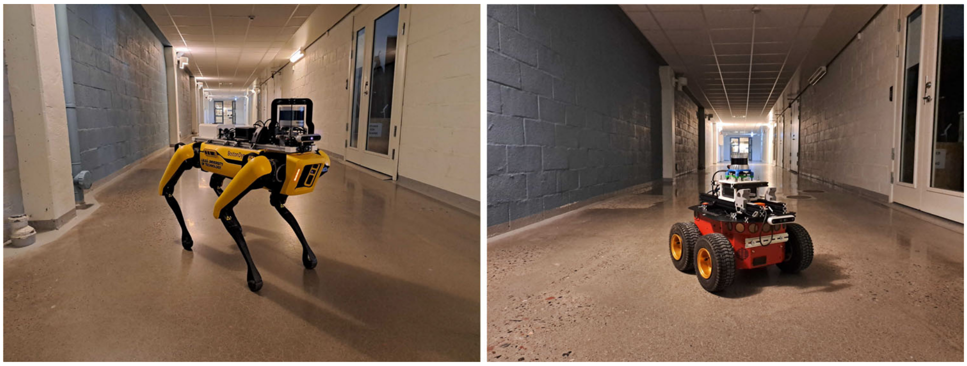


Fig. 6 The two robotic platforms, representing r_1 and r_2, utilized for the experimental evaluation in LTU's premises. (a) Spot from Boston Dynamics. (b) Pioneer 3AT, rover platform

Table 1 The experimental metric results, where M_1 and M_2 represent the amount of points in each point cloud and Tr_1 and Tr_2 represent the trajectory travelled

	M_1	M_2	Tr_1 (m)	Tr_2 (m)	OVERLAP (%)	T_e (m)	R_e (deg)	TIME (s)
Figure 7 (A) - (B)	$8.29 \cdot 10^5$	$14.15 \cdot 10^5$	843	556	35	0.183	2.318	0.648
Figure 8 (A) - (B)	$81.62 \cdot 10^5$	$48.90 \cdot 10^5$	1433	829	14	1.274	2.738	1.012

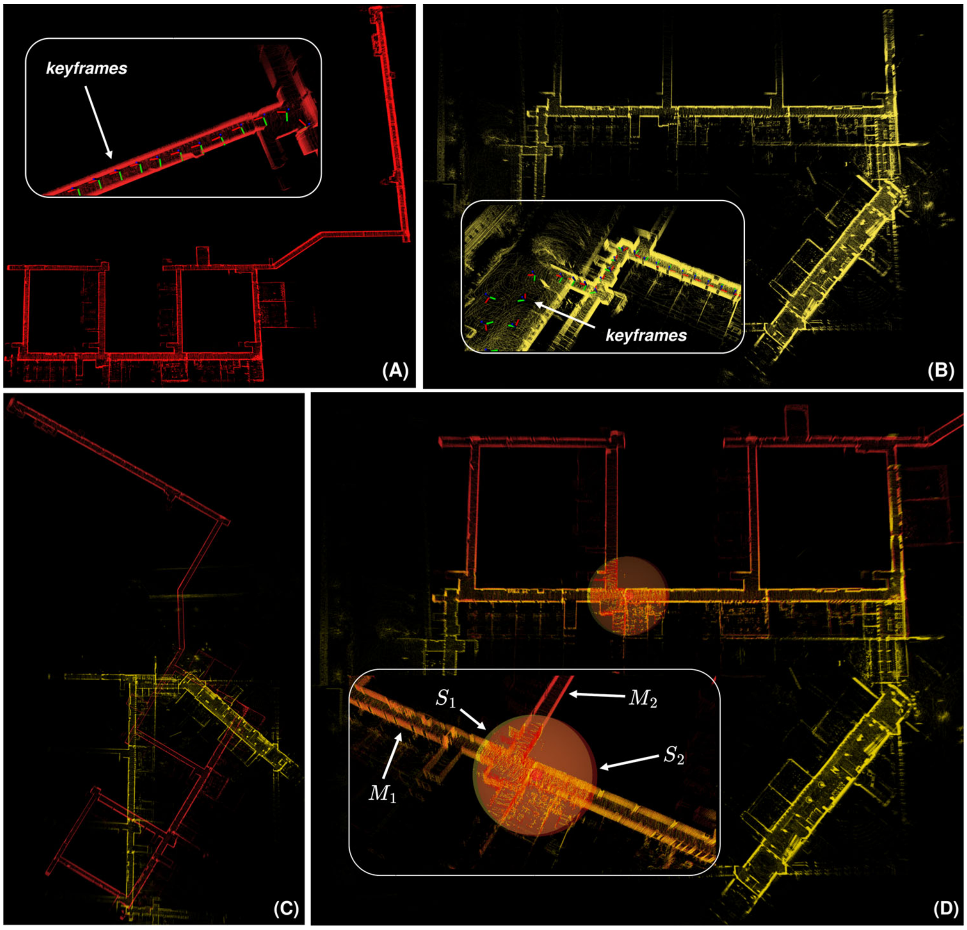
and a global map that was used for comparison purposes. More specifically, CloudCompare [35], an open-source software that provides point cloud alignment and merging, was used to create the ground truth representation. Namely, the researchers aligned the two maps, and the software provided a final rotation R_{gt} and a final translation T_{gt}. Then, two errors are defined as follows using the aforementioned ground truth transformation components:

$$T_e = ||T_{gt} - T|| \text{ and } R_e = ||R_{gt}R^{-1} - I_3||, \tag{18}$$

where T constitutes the translation part and R constitutes the rotation matrix of the final provided transform T which was yielded by the framework. Subsequently, the acquired scores for the considered scenarios are presented in Table 1, where for the first experiment the translational error is $T_e = 0.183$ m and the rotational error is $R_e = 2.318$ deg.

Figure 7 depicts the first mission's results described above. The rover robot r_1 produced M_1, consisting of approximately 829, 000 points and a total path of 843 meters, while the legged robot r_2 produced M_2, consting of approximately 1, 415, 000 points and a total path of 556 meters. For the full length of the mission, both robots constantly communicate with the edge server and transmit their locally produced map instances $m_{n,i}$. Consequently, the edge server distributes the map instances of the first robot agent to the second and contrariwise. Figure 10 depicts the map instance stitching procedure hosted in the edge server. Here, it is crucial to point out that the robots explore overlapping regions at different times; thus, the two robots do not co-locate at any

Fig. 7 The indoor environment from Luleå University of Technology, where the two robots explore different branches of the building. (A) Robot r_1 and its corresponding local map M_1. (B) Robot r_2 and its corresponding map M_2. (C) The maps M_1 and M_2 before the map merging operation. (D) The merged maps that represent the global map M_G and the overlapping regions S_1 and S_2



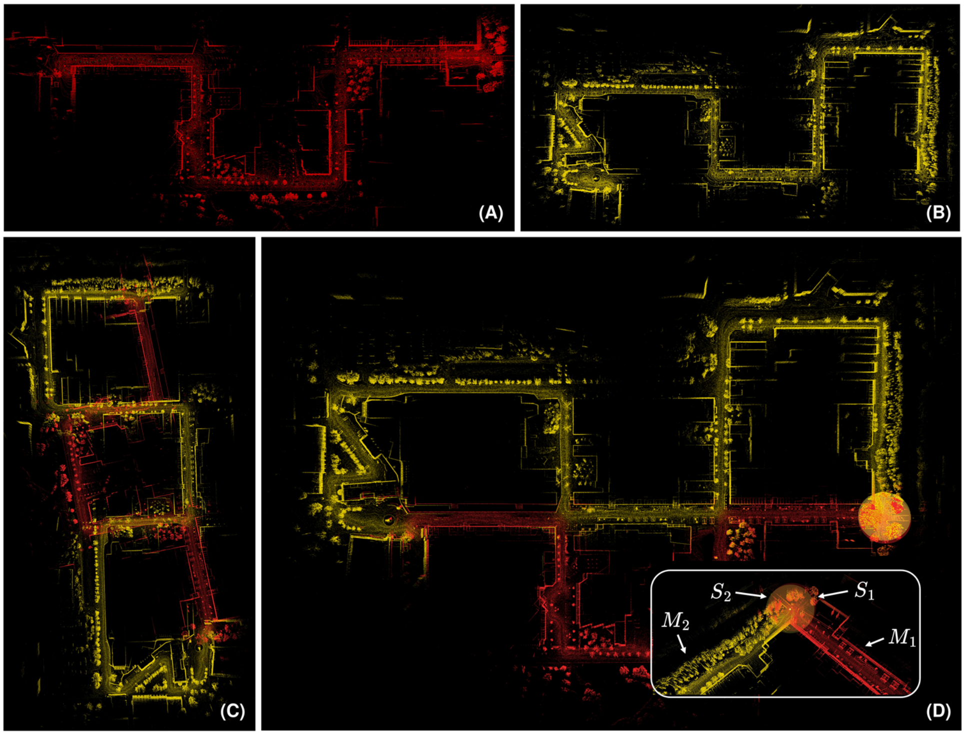


Fig. 8 Large scale outdoor environment from Luleå University of Technology, where the robots explore different paths of the campus. (A) Robot r_1 and its corresponding local map M_1. (B) Robot r_2 and its corresponding map M_2. (C) The maps M_1 and M_2 before the alignment. (D) The merged maps that represent the global map M_G and the overlapping regions S_1 and S_2

given moment throughout the described mission. Therefore, the map merging operator is triggered by an operator's input. This operation is possible since, as an outcome of the constant communication, the operator has access to the locally produced maps through the edge server. Note that this operation could also be triggered by utilizing cellular positioning [36] or a feature-matching mechanism between the two maps [37, 38]. As depicted in Fig. 7, the merging operation is successful and is performed in real time and with an average computational time of less than 1 second. From there on, the edge server distributes the yielded transformation to both robots, which consequently register the two local maps. Additionally, once the initial merging has occurred, the two robots can continuously combine the local maps as the robots continue the exploration mission, enabling the possibility of mission re-planing.

Proceeding to the second experiment, our emphasis shifts towards covering a more extensive area, thereby substantially increasing the accumulated instances and, consequently, challenging the framework's capacity to handle larger data over time. As illustrated in Fig. 8, the robots traverse two distinct routes within LTU's campus, resulting in the generation of respective maps and trajectories denoted as M_1, M_2, and Tr_1, Tr_2. As outlined in Table 1, the first map comprises 8,162,000 points, covering a total distance of 1.4 km, while the second map consists of $4, 890, 000$ points and a total trajectory length of 829 meters. Notably, the scale of this experiment is approximately ten times larger than the first, leading to a heightened translational error of $T_e = 1.274$ m. Despite this increase, the error is considered small for such a scale, with the primary contributing factor being the accumulation of drift in the employed SLAM algorithms. The map merging algorithm demonstrates efficiency with a processing time of approximately 1 second, showcasing its ability to identify overlapping regions and execute the registration algorithm within those bounds. In terms of 5G connectivity, the outdoor setting of the experiment ensures consistent connection to the 5G base station for the agents, though with occasional degraded radio channel conditions in areas affected by blockage, shadowing effects, etc. As depicted in Fig. 16, the proposed communication-aware function adeptly manages these degraded areas, facilitating a smooth exchange of information between the agents. Similar to the first experiment, the map merging operation is triggered in the same manner. Subsequently, the edge server distributes the resulting transform to both robots, enabling them to seamlessly integrate the updated information into their respective local maps for the remainder of the mission.

Finally, the definition of "real-time" depends on the intended use of the output from the map merging operation. In the described scenario, where the merged map improves the situational awareness of the MRS by identifying objects of interest, the proposed solution is considered real-time. The average computational time, as mentioned earlier, is around 1 second or less. A more detailed breakdown of the latency-dominant components is illustrated in Fig. 9. It is essential to note that, due to the experimental nature, the results are categorized into two groups. The first group

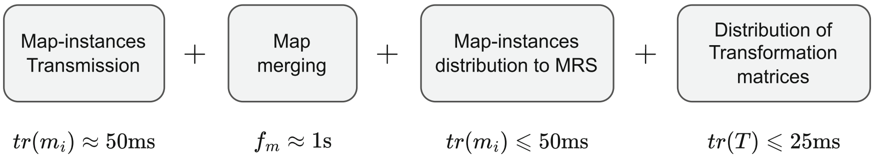

Map-instances Transmission
+
Map merging
+
Map-instances distribution to MRS
+
Distribution of Transformation matrices
$tr(m_i) \approx 50\mathrm{ms}$
$f_m \approx 1\mathrm{s}$
$tr(m_i) \leqslant 50\mathrm{ms}$
$tr(T) \leqslant 25\mathrm{ms}$

Fig. 9 Average component time resolution of operations in the proposed framework. The values correspond to good radio conditions

encompasses experiments where the robots maintained constant communication with the edge server, experiencing no loss of connectivity. The second group involves scenarios where the robots explored areas with coverage holes and potential momentary loss of connectivity. The results presented in Fig. 9 refer exclusively to the first case. In this context, without any loss of connectivity, the likely bottleneck is the map merging operation. In a related study [24] the authors evaluated two alternative map merging frameworks, previously discussed in Section 2, exhibiting varying computational times: 9.21 seconds for [21] and 260.47 seconds for [22]. Such extended computational times may cause substantial delays in the overall MRS system, affecting swift exploration and hindering critical tasks, especially in scenarios with Unmanned Aerial Vehicles (UAVs) having limited battery life. Emphasizing the need for a fast and scalable communication-aware map merging solution that delivers results within a second, particularly in expansive areas like the one shown in Fig. 8, is essential. In conclusion, the additional latency from communication components is considered negligible compared to the merging operation. As a result, the merged map can effectively enhance MRS mission situational awareness in real-time.

4.2 Communication Aware Map Transmission

To evaluate the performance of the controlled transmission of the selected map instances, the authors seek to evaluate three aspects of the proposed architecture. Initially, it is essential to showcase the transmission behavior of the map instances when the communication is degraded. Thus, a comparison is performed relating to the produced data output of the SLAM algorithm at any given moment and the selected map instances for transmission over the 5G network. Figure 11 corresponds to the same map merging experiment depicted in Fig. 7 which is presented in Section 4.1 and depicts the transmission behavior of the *controlled map instances*. More specifically, as previously mentioned, the described mission contained large regions with various communication conditions, i.e. regions with degraded coverage and a partially loaded network. The first condition correlates to the mobile nature of the 5G-enabled robots, while the second correlates to the somewhat unpredictable behavior of the cellular's network load.

4.2.1 Intermittent Communication and System's Performance

During the considered experiment, the two robots explored a large section of a basement building located at the LTU's premises. The outdoor macro base station and the RDS dots partially cover the said area. However, between the regions covered by the macro cell and the RDS units, some areas experience significantly degraded coverage. Therefore, the considered mission contains areas with high signal power values, hence ideal or satisfactory coverage, and areas that experience low signal strength. Figure 10 depicts a transition between the two coverage regions. Therefore, the latency significantly increases and reaches the peak value of approximately 1 second (compared to the average value of 18 milliseconds. Furthermore, as mentioned in Section 3.3, there is an evident correlation between the captured latency of the heartbeat signal and the lower SINR values of the 5G-enabled robots. However, its important to note that peak

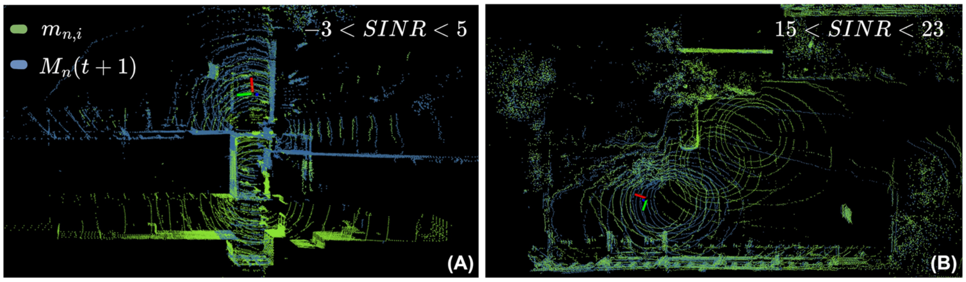
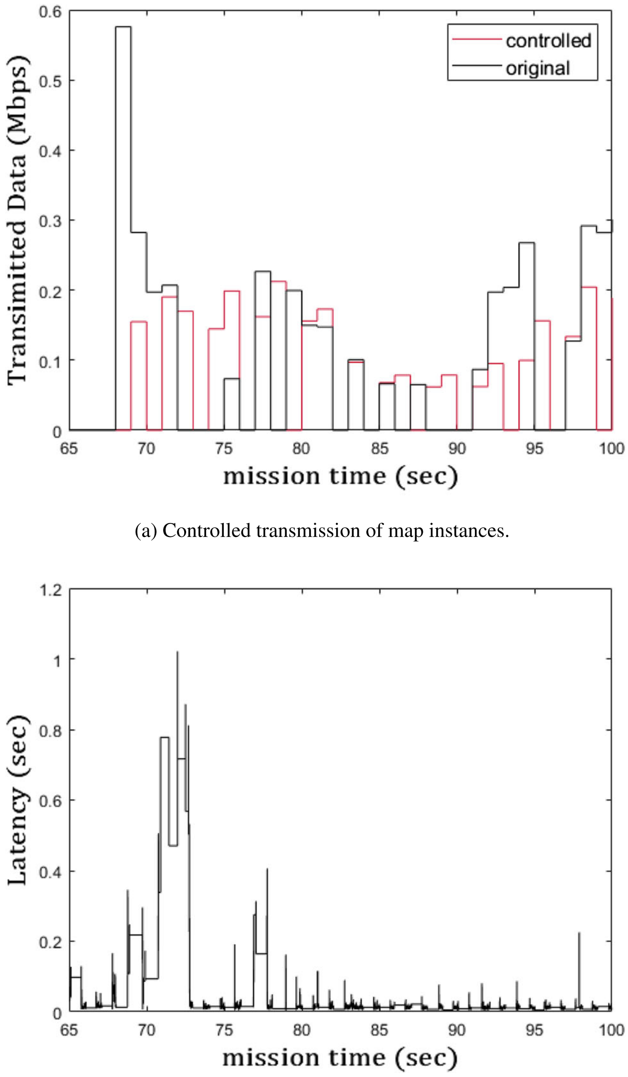


Fig. 10 Transmission of map instances in good and bad coverage conditions, captured by good and bad SINR values. The blue color points depict the remaining local map onboard the agent, and the green color points depict the transmitted points on the edge server. The left image depicts the transmission of images in good coverage conditions, thus minimum latency is observed on the figure. The right image depicts the map transmission under very weak coverage conditions, thus latency is observed due to buffered map instances on the agent

latencies close to $1\,sec$ relate to the robot's position in no coverage areas or an outcome of significant buffer build up (where in the considered scenario is handled by the control mechanism). Throughout the transition period, the SLAM algorithm constantly produces new map instances locally used to construct the local map. However, the connectivity of the 5G-enabled robot is degraded, and the controller receives the increased value of the heartbeat signal. Then, the controller adjusts the data rate produced by the SLAM algorithm. Subsequently, when the communication is restored, the output of the controlled action follows the produced data from the SLAM algorithm. The later behavior is also visible in Fig. 10. Here, the right image captures a region where the robot traversed under good SINR values, more specifically, $15 < SINR < 23$ and the map instance transmission occurred under minimum latency. On the contrary, the left image depicts a scenario were the robot traversed a region under very weak SINR values, more specifically $-3 < SINR < 5$; thus the robot traversed this region partially out of the coverage of the 5G NW. As an outcome, the robot's mission is continued completely autonomous, and the connectivity outage is detected by the `HB` signal and verified by the capture of the SINR values; this results in delayed buffered data that are transmitted in a controlled fashion when the robot re-obtains coverage. The two sub-images depict the robot's local map with the blue color and the selected map instances with the green color.

Furthermore, it is essential to mention that when a robot traverses out of coverage, it utterly loses the ability to communicate. That case is also identified by the complete disruption of the `HB` signal and the SNR values captured on the robot. Nevertheless, the robot can maintain its local autonomy and explore the environment as planned. The reason lies in the design of the proposed framework. More specifically, because of the map merging design, each robot can maintain a local map onboard, which can be utilized for autonomy. Additionally, if a map merging operation has already occurred, the robot has already updated its local map with the global map of the corresponding traversed area by the MRS. On the other hand, concerning centralized cognition (in the edge server), communication loss does not affect performance and is regarded a normal condition. More precisely, the robot communication is expected to be intermittent throughout such missions while it's entitled to the robot and the local autonomy to traverse back to coverage.

Now focusing on the transition between various radio conditions, i.e., good and bad coverage, Fig. 11 depicts that at certain moments, the output of the controlled data can be also greater than the original produced data. This behavior corresponds to situations where the control function initially reduces the data transmission. Then, when the communication is restored, the buffered data are transmitted with the corresponding rate that the control function dictates. During this process, the SLAM algorithm may or may not register any new map instances; the latter depends on the traversed environment. A potential overshoot depicts the case that the buffered data are transmitted along with newly created map instances and upon restored connectivity coverage.



(a) Controlled transmission of map instances.

(b) Latency captured by the heartbeat signal

Fig. 11 (a) The production of the local map instances with the black color. The red color depicts the controlled transmitted map instances. The behavior of the red curve corresponds to the output of the control signal (latencies of the system) which is depicted in (b). Note that the increased latency of approximately 1 sec is captured in very low SNR values

4.2.2 Large Data Transmission and Transport Layer Protocol Choice

Of particular interest are the cases that involve large data transmissions. Large data transmissions stress-test the ability of the framework and can highlight the performance of

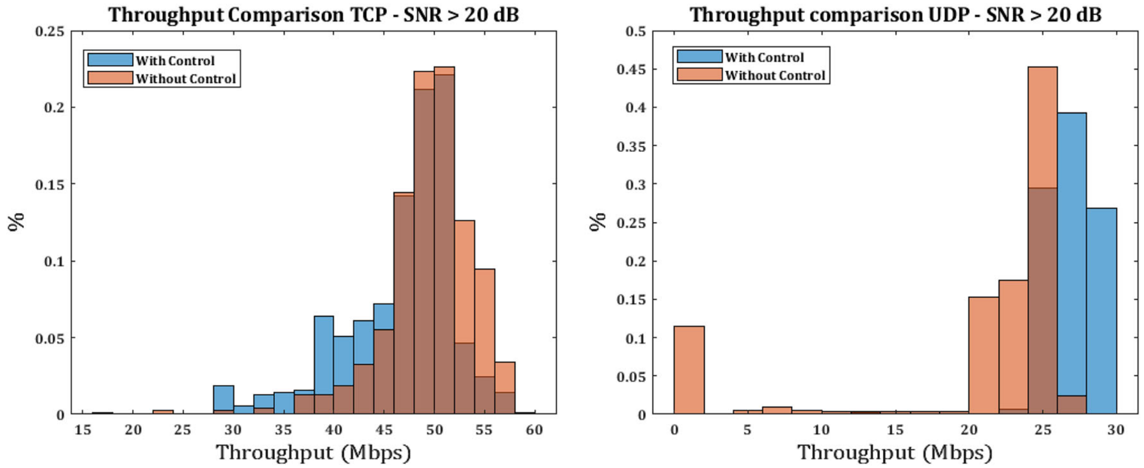


Fig. 12 Captured throughput for $SNR > 20$. The histograms compare between experiments with the application of the control mechanism and without. Left side depicts experiments conducted with the TCP protocol. Right side depicts experiments conducted with the UDP protocol

the proposed solution under various communication conditions (different SNR and network cell load values). For this set of experiments, the produced maps were on average ~ 42 Mbps. Note that such uplink data rates are high in relation to the maximum uplink cell throughput of 94 $Mbps$ that can be achieved with the given communication system. Additionally, the relationship between the achieved throughput and latency in the application layer and the robot's radio conditions is visible, as there is less margin for worse channel conditions when the data rate is high [39]. Moreover, different transport layer protocols (TLP) were utilized, either TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) and the performance of each protocol is discussed later on.

The UDP protocol is commonly used in networked robotics applications due to its lower latency in normal conditions, lacking retransmissions, handshakes, and a flow congestion mechanism compared to TCP. However, in situations where the client lacks enough throughput capacity, this can lead to full data buffers, dropped packets, and uncontrolled latency rise. Design considerations should align with the specific application and data flows. For time-critical data, sacrificing a few packets for lower latency may be acceptable, whereas for essential, reliable transmission, sacrificing latency may be tolerable. Also, consider that over-the-top congestion control mechanisms (over the UDP layer) can be employed to regulate transmission [40]. For example, in the presented scenario, the reliable transmission of the transformation matrix T is crucial, making TCP the suitable choice due to its reliability and tolerance for lower throughput. The proposed approach regulates data transmission based on the health of the `HB` signal, which oversees critical robot operations.

Figures 12, 13 and 14 depict the achieved throughput captured for different regions of SNR. The comparison has been

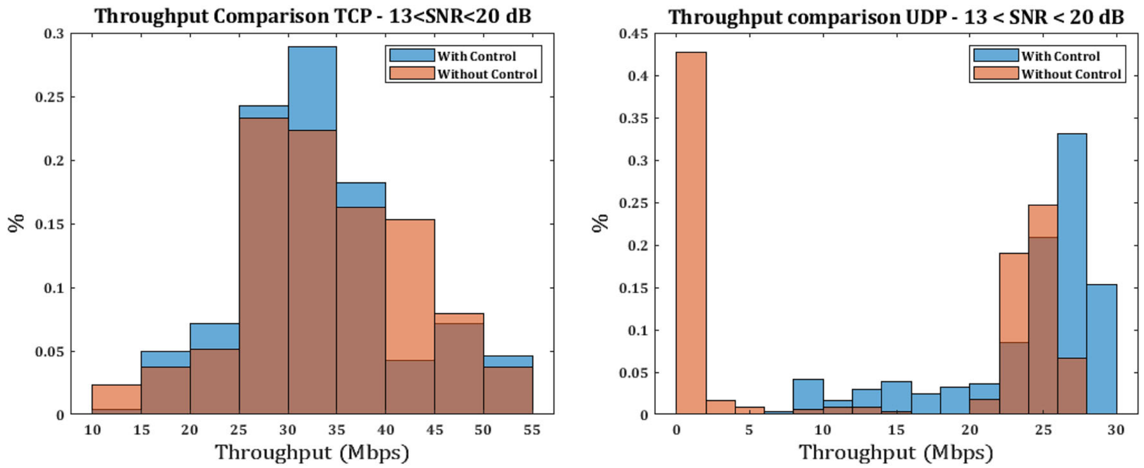


Fig. 13 Captured throughput for $13 < SNR < 20$. The histograms compare between experiments with the application of the control mechanism and without. Left side depicts experiments conducted with the TCP protocol. Right side depicts experiments conducted with the UDP protocol

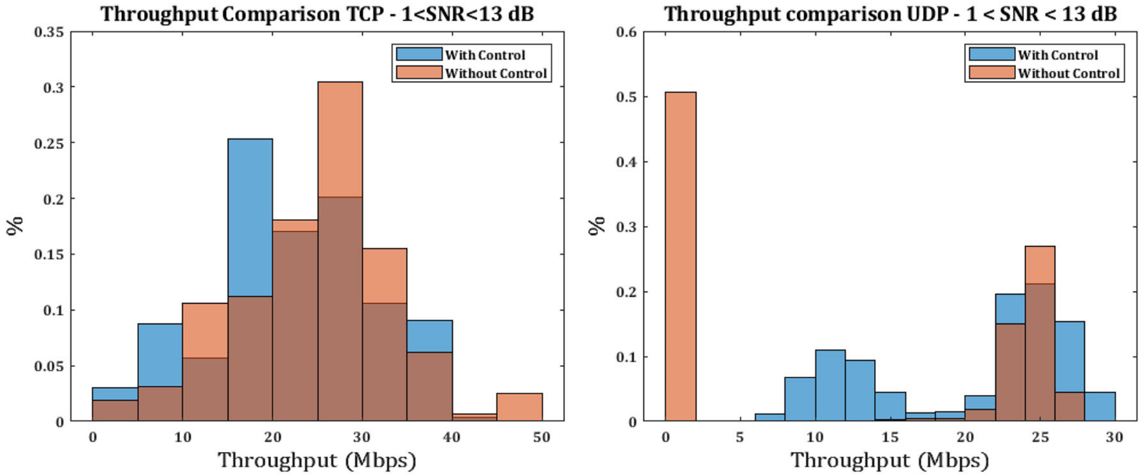


Fig. 14 Captured throughput for $1 < SNR < 13$. The histograms compare between experiments with the application of the control mechanism and without. Left side depicts experiments conducted with the TCP protocol. Right side depicts experiments conducted with the UDP protocol

made with and without the utilization of the proposed control strategy. The histograms in various radio conditions show that the throughput is not significantly reduced when the control mechanism is applied - this is an important finding when examining the behavior of the HB signal's latency. A set of 14 experiments over the TCP protocol and another set of 10 experiments over the UDP protocol have been performed. The network was congested with various loads across all the conducted experiments. Finally, please note that the raw and unregulated (before the application of the control mechanism) throughput for the map instances is less in the particular set of UDP experiments. The latter is an outcome of narrower traversed areas and the selection of different sensor resolutions. The average value for the raw map instance generation is ∼ 50 Mbps for the TCP set of the experiments and ∼ 30 Mbps for the UDP set. More specifically, in Figs. 12 - 14, the throughput comparison with and without the control mechanism for the TCP case is quite similar. The main difference is that the control mechanism spreads the histograms in lower captured throughput values - this observation is in benefit of the HB signal's latency and is depicted in Figs. 17 and 18. For the UDP case in Figs. 12 - 14, the benefit of the control mechanism is evident, as it is clear that without it, and when channel quality degrades the UDP link continues to send packets without any consideration for the transmission's success. The significant percentage of 0 Mbps received is an outcome of dropped packets in the application layer.

The main purpose of the control mechanism is to regulate the transmission of the map instances under various communication conditions so that constant communication is preserved and the robot's transmission buffers in the application layer are not flooded. The main reason that the HB's signal latency is selected for the control mechanism, and not the latency of the transmitted map instances themselves is that the HB signal is of higher importance. For example, it is essential to maintain constant communication with the robot (when the platform is within coverage) so that essential information is communicated. The HB signal usually contains robot telemetry data that capture the health of the robot (battery readings, temperature, position of the robot or objects of interest, etc.) and can be also used to command the robot to execute various tasks (emergency termination, navigation information, deployment of supplies, etc.). If the transmission of large sensor data (i.e., map instances) is not regulated it is probable that under varying channel quality conditions and varying network load the wireless connected robot might proceed to generate and transmit more data than the communication link between the robot and the base station can

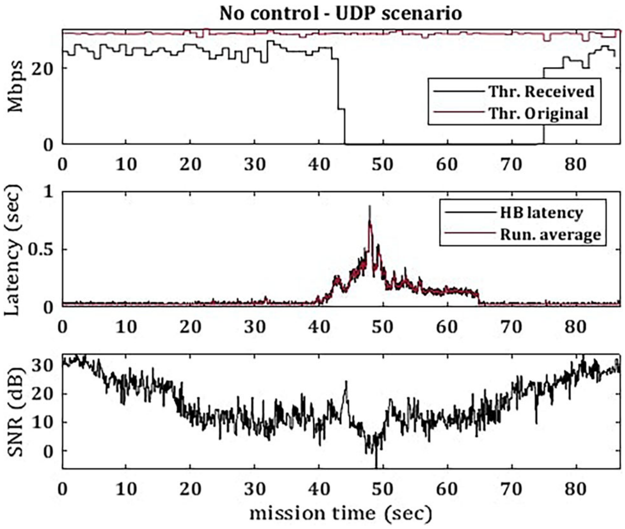


Fig. 15 Mission snapshot of the communication KPIs from one 5G connected robot. No control mechanism was applied. The experiment utilized the UDP protocol. On low SNR values, the map instances throughput goes to 0 Mbps

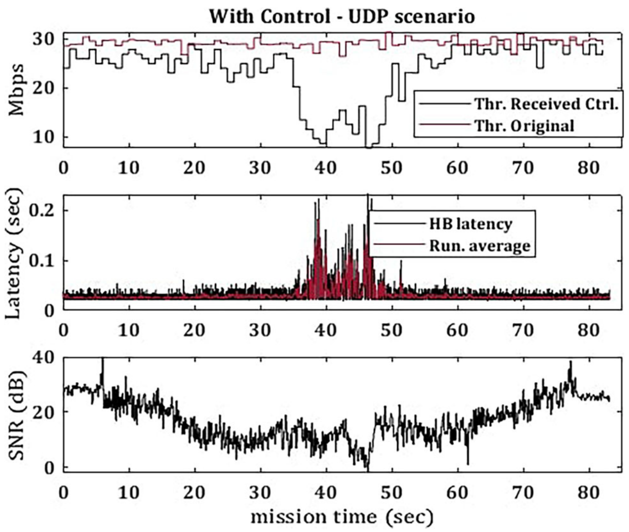


Fig. 16 Mission snapshot of the communication KPIs from one 5G connected robot. The proposed control mechanism was applied. The experiment utilized the UDP protocol. On low SNR values, the map instances throughput reaches a minimum of 7.68 Mbps

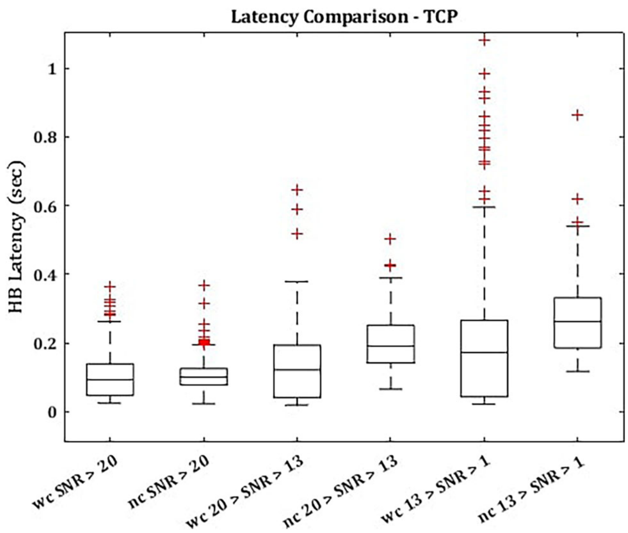


Fig. 17 Latency of the HB signal across 14 conducted experiments. The results are depicted in three SNR groups. The comparison has been made between the application of the control function (abbreviated with *"wc"*) and without abbreviated with *"nc"*. The corresponding experiments utilized the TCP protocol

support at the moment - thus, resulting in extended buffer delays. The later is also subject to the utilized transport protocol, e.g. TCP employs a congestion control mechanism, but UDP does not [41].

Figure 15 depicts a closer look with respect to time for the case of large data transmission on a partially loaded network and different radio signal conditions without the control mechanism and over the UDP protocol. Here, it is evident that as the robot traverses to worse signal conditions, the latency of the HB signal significantly rises, and the overall transmission of the corresponding map instances drops to 0 Mbps. The latter is due to dropped packets and the reception of damaged packets in the application layer while maintaining a highly congested channel and utilizing the UDP protocol that inherently does not account for dropped packets and congestion. On the other hand, Fig. 16 depicts the case for which the application of the control mechanisms occurs. Here, it is evident that the latency of the HB signal is significantly reduced even in the worst radio conditions. Also, note that the transmission of map instances never fails. More specifically, the global minimum throughput for the depicted mission takes the value of 7.68 Mbps. The latter two observations are a direct outcome of the control mechanism - reduced transmissions do not overload the application's layer data buffers for both the data flows of the map instances and the HB signal.

Figure 17 illustrates the RTT latency of the HB signal using the TCP protocol, while Fig. 18 represents the same over the UDP protocol. These experiments were conducted under varying radio conditions. Note that, in the considered scenarios, optimal radio conditions are characterized by an $SNR \geq 20$ dB. Then, within optimal radio conditions, the mean latency remains relatively consistent, irrespective of the application of the control mechanism. However, as the radio channel quality degrades, a noticeable increase in latency is observed in cases where the control mechanism is not applied. Table 2 summarizes the statistics gathered from 24 experiments, with 10 trials using the UDP protocol and 14 using the TCP protocol. Lastly, the complete proposed framework was assessed in a significantly larger area, and the entire

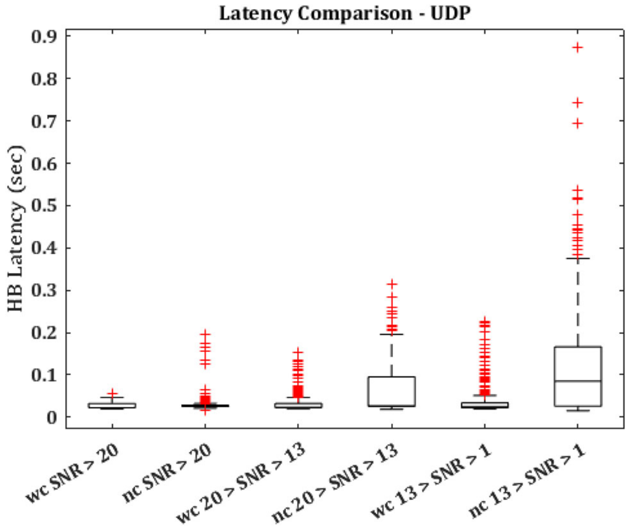


Fig. 18 Latency of the HB signal across 10 conducted experiments. The results are depicted in three SNR groups. The comparison has been made between the application of the control function (abbreviated with *"wc"*) and without abbreviated with *"nc"*. The corresponding experiments utilized the UDP protocol

Table 2 RTT Latency average across 14 experiments with the application of the control mechanism (wc) and without (nc) for the TCP case and 10 experiments for the UDP case

Mode	20 < SNR (dB)	20 > SNR > 13 (dB)	1 > SNR (dB)
Latency - wc - TCP (sec)	0.0687	0.0979	0.1485
Latency - nc - TCP (sec)	0.0764	0.1674	0.2375
Latency - wc - UDP (sec)	0.0256	0.0298	0.0371
Latency - nc - UDP (sec)	0.0288	0.0640	0.1250

mission, along with the merged maps, is depicted in Fig. 8. It's important to note that this particular mission is also part of one of the 14 conducted experiments shown in Fig. 17.

Lastly, in order to further assess the controller's behavior, an approach where simulated and experimental data are combined is chosen. For this evaluation, the captured heartbeat signal from the presented experiment is used in combination with simulated map instances. Utilizing such an approach, the production rate of the map instances can be constant. As a result, it is possible to observe the controlled transmission's raw output more clearly. The observed behavior is depicted in Fig. 19. Here a clear correlation with the latency of the system is visible. Further, some may argue that the output of the controlled transmission is aggressive. This behavior is attributed to the heartbeat signal's abrupt changes. In that case, to smoothen the output, one could investigate a non-linear control function or consider the addition of further KPIs that capture the complete transition between connectivity levels.

5 Discussion

In light of the presented performance and results of the proposed framework, this section highlights the central themes of the study, emphasizing critical challenges in technical, integration, and design aspects of the problem under consideration. Further, Section 5.2 concludes the discussion by presenting a proposed methodology that one should consider when designing similar applications.

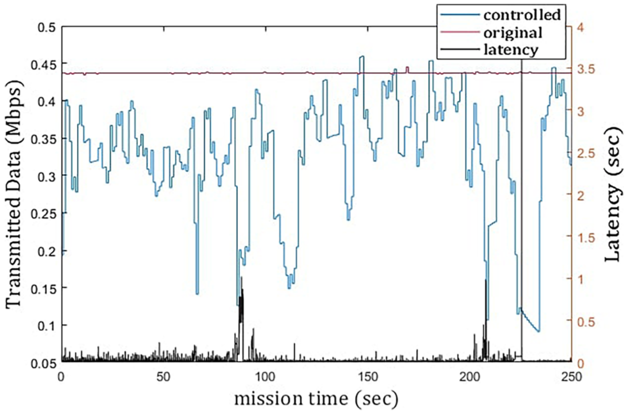


Fig. 19 Evaluation of the transmission control function on simulated constant production of map instances. The used heartbeat (latency) signal is obtained from the real experiment

5.1 Challenges and Limitations

Concluding this work, it is essential to highlight and summarize the important challenges, learnings and limitations of the proposed framework and the studied phenomena. Figure 20 depicts the complete mission architecture for each specific agent. The highlighted boxes with grey background represent the components of the proposed framework, while the components with white background represent other vital processes. The underline goal can be abridged to the few critical requirements: 1) transmit all the required map instances to the centralized agent without losing data, 2) do not exceed the throughput uplink capabilities of each robot, 3) distribute map instances to the whole MRS, 4) perform successful map merging operations in the edge cloud, 5) reliably distribute the transformation matrix $T \in SE(3)$. For the first challenge, although our framework relies on the utilization of a 5G network where data packets are rarely lost, it is important to understand that there are many other critical components where data packets can get lost. For example, TLP buffers can be one of the main ones, then ROS subscribers and publisher buffers, 5G module buffers, and so on. The second challenge is often a direct cause of the first one. Here, the discussion concerns the HB signal. If the 5G connected device is constantly trying to transmit more data than its current capabilities allow, then the likelihood of creating full buffers increases. This problem affects all the data connections of the connected robot. This scenario is possible since the robot is intended to potentially traverse regions with degraded connectivity.

The third challenge is also a direct outcome of the second challenge. If the centralized unit possesses all the produced information from the MRS, it can reliably transmit back to the MRS by looking at the `HB` signal. The latter is also valid because most 5G systems are optimized to provide higher throughput in the downlink direction. In addition to the challenges mentioned earlier, the fourth and final challenge concerns the performance of the proposed map merging accuracy, which is demonstrated and compared on multiple occasions. Concluding this rationale, buffers in each partic-

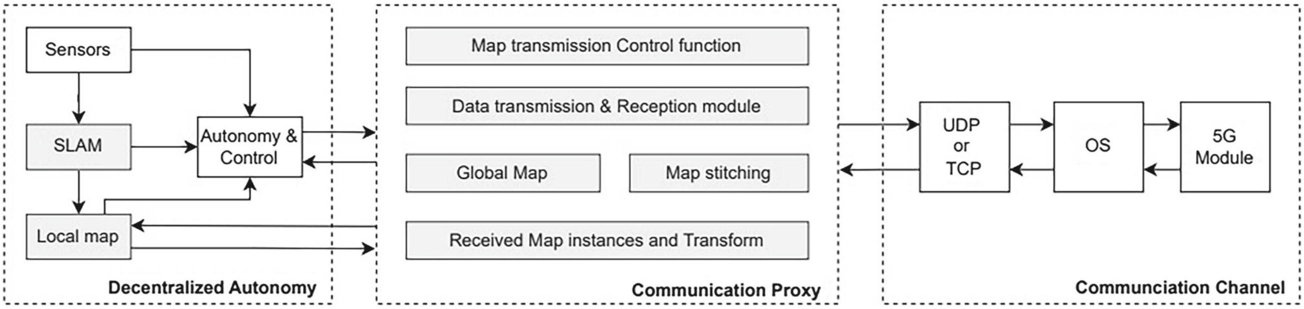


Fig. 20 Framework architecture on the 5G enabled robot side. Utilized components for sensor modalities, onboard algorithms, and the communication software stack. The main focus of this research is highlighted with gray

ular device are the key bottleneck of the presented problem. It is essential to prevent buffer overflow, which is one of the main objectives. Therefore, the latency of HB signal in the application layer is representative to the robot's communication with the centralized units and human operators - thus it was chosen as the most critical parameter to preserve.

Another important challenge concerns the potential of single point of failure (SPOF). In that case, the potential failing point is the centralized unit in the edge cloud. Although the current work does not focus on that particular challenge the proposed architecture is designed to enable the application of robust safety measures on the centralized component - the authors provide the following analysis and potential use of the cloud technologies:

Initially, it has to be addressed that due to the fact that each agent is designed to hold the potential for individual autonomy by maintaining a local map onboard, then, even on the complete failure of the centralized unit, each agent can continue the mission and apply safety fall-backs, for example, return-to-base activity [42]. Thus, it's up to the individual agent to guarantee its safety in that scenario. Nevertheless, the utilization of cloud computing technologies along with local robot autonomy makes the deployment of the centralized unit resilient to failures. More specifically, as mentioned above, the centralized unit is deployed on a Kubernetes (commonly noted as k8s) enabled environment [43]. Consequently, it is implied that the application is deployed in a software bundle along with all the required deployments in the form of a Docker container [44]. To that end, one can use k8s features to maintain a safe operation. k8s is a container orchestrator manager that is responsible for the docker container lifetime. For example, k8s are capable of container monitoring, container deployment and termination, container run-time memory, storage management, and others [45, 46]. Hence, for the proposed application, one potential use of k8s features is to store the received map instances, monitor the health of the application, and in case of failure re-deploy the centralized unit while maintaining the received map instances.

Regarding the limitations of the proposed map merging solution over a 5G network, the authors emphasize two key points. The first limitation is associated with the current control mechanism for transmitting map instances from the isolated SLAM components, which does not account for the latency of the transmitted map instances. While this is intentional, the authors suggest that additional communication KPIs could enhance the control mechanism, optimizing the joint latency aware transmission of map instances and latency of the HB signal. It's important to note that this stands out as a primary consideration for future improvements. The second limitation pertains to the manual initiation of the merging operation by a human operator. The authors propose the development of an automated triggering mechanism to enable optimal map merging operations, potentially aligning with the mission's requirements.

5.2 Proposed Guidelines

To ensure the success of the proposed map merging framework over 5G, several key components require careful consideration. Primarily, it is crucial to address the robot's autonomy and account for potential out-of-coverage regions. Specifically, the design should be tailored to potential scenarios. If robots are anticipated to venture into out-of-coverage regions, incorporating fallback actions such as “return to coverage" within the local autonomy of the robot is essential to ensure the uninterrupted and safe continuation of operations (note that this aspect was beyond the scope of the current work). Second, all the data flows connecting the robot to any external unit must be identified and classified in terms of importance. The important characteristics to consider for each data flow are the low and upper bounds of latency and throughput characteristics. Therefore, the latter will indicate which data flows must be developed with and without a data rate control mechanism and the corresponding transport layer protocol. Then, when the latency of the essential data flows is softly bounded under normal coverage conditions, this will characterize the potential for real-time map merging and the corresponding map merging triggering mechanism. Lastly, since the proposed map merging operation is a centralized mechanism, it is vital to consider SPOF and appropriate resilience mechanisms as proposed in Section 5.1.

6 Conclusions

This article presented a novel architecture for multiple 5G-enabled robotic agents that seek the maximization of situational awareness. The main focus was on creating a communication-aware framework that adjusts data transmission according to communication conditions to prevent significant buffer build up (on the agent side), extensive latencies and account for varying channel conditions. Moreover, this research introduced a master-slave architecture in which an extra agent located in the edge server of the cellular network is responsible for the operation of the map merging procedure and the distribution of local information throughout the MRS. The benefits of utilizing the existing cellular infrastructure become evident throughout a series of experimental evaluations. The experimental evaluation assesses a realistic scenario where an outdoor and indoor space has to be explored and mapped so that the produced maps can be further utilized for additional processing, e.g. object detection and localization. Additionally, processes that are hosted on the master agent can utilize the full resources of the edge server's computing cluster. Even though the proposed control function does not include additional KPIs, a correlation between the proposed one, i.e. the latency of the system, and the SINR values is identified and justified; the expansion in this direction is planned for future work. Further, some could consider KPIs that identify the cell load and the mobile characteristic of the robot agents. Evidently, the proposed solution introduces delays in data transmission that could affect more time-critical operations. Hence, other future research directions could consider the proposed architecture for operating time-critical operations, such as the closed-loop control of robots while utilizing remote controllers.

Author Contributions Gerasimos Damigos and Nikolaos Stathoulopoulos: Development, implementation, system integration and field work, relating to all presented submodules and developments, main manuscript contributors. Tore Lindgren, Anton Koval and George Nikolakopoulos: Advisory, manuscript contributions and supervision. All authors have read and approved the manuscript.

Funding Open access funding provided by Lulea University of Technology. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 953454.

Statements and Declarations

Competing interests The authors have no conflicts of interest with any related parties.

References

1. Chen, S.W., Nardari, G.V., Lee, E.S., Qu, C., Liu, X., Romero, R.A.F., Kumar, V.: SLOAM: Semantic Lidar Odometry and Mapping for Forest Inventory. IEEE Robot. Autom. Lett. **5**(2), 612–619 (2020). https://doi.org/10.1109/LRA.2019.2963823
2. Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., Hahnel, D., Montemerlo, D., Morris, A., Omohundro, Z., Reverte, C.: Autonomous exploration and mapping of abandoned mines. IEEE Robot. Autom. Mag. **11**(4), 79–91 (2004). https://doi.org/10.1109/MRA.2004.1371614
3. Jenssen, R., Roverso, D., et al.: Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. Int J. Electr. Power Energy Syst. **99**, 107–120 (2018)
4. Scaramuzza, D., Achtelik, M.C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., Achtelik, M.W., Chli, M., Chatzichristofis, S., Kneip, L., Gurdan, D., Heng, L., Lee, G.H., Lynen, S., Pollefeys, M., Renzaglia, A., Siegwart, R., Stumpf, J.C., Tanskanen, P., Troiani, C., Weiss, S., Meier, L.: Vision-Controlled Micro Flying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments. IEEE Robot. Autom. Mag. **21**(3), 26–40 (2014)
5. Lindqvist, B., Kanellakis, C., Mansouri, S.S., Agha-mohammadi, A.-A., Nikolakopoulos, G.: COMPRA: A COMPact Reactive Autonomy Framework for Subterranean MAV Based Search-And-Rescue Operations. J. Intell. Robot. Syst. **105**(3), 49 (2022). https://doi.org/10.1007/s10846-022-01665-6
6. Agha, A., Otsu, K., Morrell, B., Fan, D.D., Thakker, R., Santamaria-Navarro, A., Kim, S.-K., Bouman, A., Lei, X., Edlund, J.A., Ginting, M.F., Ebadi, K., Anderson, M.O., Pailevanian, T., Terry, E., Wolf, M.T., Tagliabue, A., Vaquero, T.S., Palieri, M., Tepsuporn, S., Chang, Y., Kalantari, A., Chavez, F., Lopez, B.T., Funabiki, N., Miles, G., Touma, T., Buscicchio, A., Tordesillas, J., Alatur, N., Nash, J., Walsh, W., Jung, S., Lee, H., Kanellakis, C., Mayo, J., Harper, S., Kaufmann, M., Dixit, A., Correa, G., Lee, C.-A., Gao, J.L., Merewether, G.B., Maldonado-Contreras, J., Salhotra, G., Silva, M.S.D., Ramtoula, B., Fakoorian, S.A., Hatteland, A., Kim, T., Bartlett, T., Stephens, A., Kim, L., Bergh, C.F., Heiden, E., Lew, T., Cauligi, A., Heywood, T., Kramer, A., Leopold, H., Melikyan, H., Choi, H.-Y., Daftry, S., Toupet, O., Wee, I., Thakur, A., Feras, M., Beltrame, G.A., Nikolakopoulos, G., Shim, D.-Y., Carlone, L., Burdick, J.W.: NeBula: TEAM CoSTAR's Robotic Autonomy Solution that Won Phase II of DARPA Subterranean Challenge. Field Robot. **2**, 1432–1506 (2022)
7. Saboia, M., Clark, L., Thangavelu, V., Edlund, J.A., Otsu, K., Correa, G.J., Varadharajan, V.S., Santamaria-Navarro, A., Touma, T., Bouman, A., Melikyan, H., Pailevanian, T., Kim, S.-K.,

Archanian, A., Vaquero, T.S., Beltrame, G., Napp, N., Pessin, G., Agha-mohammadi, A.-a: ACHORD: Communication-Aware Multi-Robot Coordination With Intermittent Connectivity. IEEE Robot. Autom. Lett. **7**(4), 10184–10191 (2022). https://doi.org/10.1109/LRA.2022.3193240

8. Jamshidpey, A., Wahby, M., Heinrich, M.K., Allwright, M., Zhu, W., Dorigo, M.: Centralization vs. decentralization in multi-robot coverage: Ground robots under UAV supervision (2021)
9. Gielis, J., Shankar, A., Prorok, A.: A Critical Review of Communications in Multi-robot Systems. Curr. Robot. Rep. **3**(4), 213–225 (2022). https://doi.org/10.1007/s43154-022-00090-9
10. Matignon, L., Jeanpierre, L., Mouaddib, A.-I.: Coordinated Multi-Robot Exploration Under Communication Constraints Using Decentralized Markov Decision Processes. Proc. AAAI Conf. Artif. Intell. **26**(1), 2017–2023 (2021). https://doi.org/10.1609/aaai.v26i1.8380
11. Zaki, O., Dunnigan, M., Robu, V., Flynn, D.: Reliability and Safety of Autonomous Systems Based on Semantic Modelling for Self-Certification. Robotics **10**(1), (2021). https://doi.org/10.3390/robotics10010010
12. Damigos, G., Lindgren, T., Sandberg, S., Nikolakopoulos, G.: Performance of Sensor Data Process Offloading on 5G-Enabled UAVs. Sensors **23**(2), 864 (2023)
13. Schmuck, P., Chli, M.: CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. J. Field Robot. **36**, 763–781 (2019). https://doi.org/10.1002/rob.21854
14. Zou, D., Tan, P.: Coslam: Collaborative visual slam in dynamic environments. IEEE Trans. Pattern Anal. Mach. Intell. **35**(2), 354–366 (2012)
15. Engel, J., Koltun, V., Cremers, D.: Direct Sparse Odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2018). https://doi.org/10.1109/TPAMI.2017.2658577
16. Forster, C., Lynen, S., Kneip, L., Scaramuzza, D.: Collaborative monocular slam with multiple micro aerial vehicles. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3962–3970 (2013). IEEE
17. Queralta, J.P., Gia, T.N., Tenhunen, H., Westerlund, T.: Collaborative mapping with ioe-based heterogeneous vehicles for enhanced situational awareness. In: 2019 IEEE sensors applications symposium (SAS), pp. 1–6 (2019). IEEE
18. Hörner, J.: Map-merging for multi-robot system (2016)
19. Siebert, J.: SIFT Keypoint Descriptors for Range Image Analysis. Annals of the BMVA (2009)
20. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Proceedings of the 4th alvey vision conference, pp. 147–151 (1988)
21. Hörner, J.: ROS package for merging 3D point cloud maps. (2018). https://github.com/hrnr/map-merge.git
22. Drwiega, M.: Incremental 3D Maps Server based on Feature Matching for Multi-robot Systems. In: 2021 25th International conference on methods and models in automation and robotics (MMAR), pp. 342–347 (2021). https://doi.org/10.1109/MMAR49549.2021.9528427
23. Salti, S., Tombari, F., Di Stefano, L.: SHOT: Unique Signatures of Histograms for Surface and Texture Description. Comput. Vis. Image Underst. **125**, (2014). https://doi.org/10.1016/j.cviu.2014.04.011
24. Stathoulopoulos, N., Koval, A., Agha-mohammadi, A.-a., Nikolakopoulos, G.: FRAME: Fast and Robust Autonomous 3D Point Cloud Map-Merging for Egocentric Multi-Robot Exploration. In: 2023 IEEE international conference on robotics and automation (ICRA), pp. 3483–3489 (2023). https://doi.org/10.1109/ICRA48891.2023.10160771
25. Dubé, R., Cramariuc, A., Dugas, D., Sommer, H., Dymczyk, M., Nieto, J., Siegwart, R., Cadena, C.: SegMap: Segment-based mapping and localization using data-driven descriptors. Int. J. Robot. Res. **39**(2–3), 339–355 (2020). https://doi.org/10.1177/0278364919863090
26. Stathoulopoulos, N., Koval, A., Nikolakopoulos, G.: 3DEG: Data-Driven Descriptor Extraction for Global re-localization in subterranean environments. Expert Syst. Appl. **237**, 121508 (2024). https://doi.org/10.1016/j.eswa.2023.121508
27. Koide, K., Yokozuka, M., Oishi, S., Banno, A.: Voxelized GICP for Fast and Accurate 3D Point Cloud Registration. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 11054–11059 (2021). https://doi.org/10.1109/ICRA48506.2021.9560835
28. Segal, A.V., Hähnel, D., Thrun, S.: Generalized-ICP. In: Robotics: Science and Systems (2009)
29. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D.: LIO-SAM: Tightly-coupled LiDAR Inertial Odometry via Smoothing And Mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5135–5142 (2020). IEEE
30. Palieri, M., Morrell, B., Thakur, A., Ebadi, K., Nash, J., Chatterjee, A., Kanellakis, C., Carlone, L., Guaragnella, C., Agha-mohammadi, A.-a.: LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time. IEEE Robot. Autom. Lett. **6**(2), 421–428 (2021)
31. Shan, T., Englot, B., Ratti, C., Daniela, R.: LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 5692–5698 (2021). IEEE
32. Chen, K., Lopez, B.T., Agha-mohammadi, A.-A., Mehta, A.: Direct LiDAR Odometry: Fast Localization With Dense Point Clouds. IEEE Robot. Autom. Lett. **7**(2), 2000–2007 (2022). https://doi.org/10.1109/LRA.2022.3142739
33. Kim, B., Jung, C., Shim, D.H., Agha-mohammadi, A.: Adaptive Keyframe Generation based LiDAR Inertial Odometry for Complex Underground Environments. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 3332–3338 (2023). https://doi.org/10.1109/ICRA48891.2023.10161207
34. TS 23.501: System architecture for the 5G System (5GS). 3rd Generation Partnership Project (3GPP) (2023). https://www.3gpp.org/
35. Cloud-Compare: 3D point cloud and mesh processing software Open Source Project. https://www.danielgm.net/cc/
36. Mogyorósi, F., Revisnyei, P., Pašić, A., Papp, Z., Törös, I., Varga, P., Pašić, A.: Positioning in 5G and 6G networks-A Survey. Sensors **22**(13), 4757 (2022)
37. Chen, X., Läbe, T., Milioto, A., Röhling, T., Vysotska, O., Haag, A., Behley, J., Stachniss, C.: OverlapNet: Loop Closing for LiDAR-based SLAM. In: Proceedings of robotics: science and systems (RSS) (2020)
38. Stathoulopoulos, N., Pagliari, E., Davoli, L., Nikolakopoulos, G.: Redundant and Loosely Coupled LiDAR-Wi-Fi Integration for Robust Global Localization in Autonomous Mobile Robotics. In: 21st International conference on advanced robotics (ICAR) (2023). https://doi.org/10.48550/arXiv.2310.06384
39. Ramadan, E., Narayanan, A., Dayalan, U.K., Fezeu, R.A., Qian, F., Zhang, Z.-L.: Case for 5G-aware video streaming applications. In: Proceedings of the 1st workshop on 5g measurements, modeling, and use cases, pp. 27–34 (2021)
40. Brunello, D., Johansson S, I., Ozger, M., Cavdar, C.: Low Latency Low Loss Scalable Throughput in 5G Networks. In: 2021 IEEE 93rd vehicular technology conference (VTC2021-Spring), pp. 1–7 (2021). https://doi.org/10.1109/VTC2021-Spring51267.2021.9448764
41. Tzes, A., Nikolakopoulos, G., Koutroulis, I.: Development and experimental verification of a mobile client-centric networked controlled system. Eur. J. Control **11**(3), 229–241 (2005)

42. Thangavelautham, J., Chandra, A., Jensen, E.: Autonomous robot teams for lunar mining base construction and operation. In: 2020 IEEE Aerospace Conference, pp. 1–16 (2020). IEEE
43. Kubernetes.io: Kubernetes Documentation - General Introduction. https://kubernetes.io/ Accessed 3-Nov-2023
44. Kubernetes: Kubernetes Documentation. https://www.3gpp.org/technologies/scheduling Accessed 3-Nov-2023
45. Xia, C., Zhang, Y., Wang, L., Coleman, S., Liu, Y.: Microservice-based cloud robotics system for intelligent space. Robot. Auton. Syst. **110**, 139–150 (2018)
46. Seisa, A.S., Damigos, G., Satpute, S.G., Koval, A., Nikolakopoulos, G.: Edge Computing Architectures for Enabling the Realisation of the Next Generation Robotic Systems. In: 2022 30th Mediterranean conference on control and automation (MED), pp. 487–493 (2022). https://doi.org/10.1109/MED54222.2022.9837289

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Gerasimos Damigos holds an M.Sc. in electrical and computer engineering from the University of Patras, Greece, and is presently dedicated to pursuing a Ph.D. with Luleå University of Technology. Concurrently, he is actively engaged in research at Ericsson Research and is a contributor to the European Horizon 2020 AERO-TRAIN Project. His research pursuits primarily revolve around robotics, unmanned aerial vehicles (UAVs), and cellular networks, with a primary emphasis on 5G networks.

Nikolaos Stathoulopoulos received his BEng degree with an integrated MEng degree in Electrical and Computer Engineering from the University of Patras in 2022, specializing in Signal, Systems and Control. He is currently pursuing a Ph.D. in the Robotics and Artificial Intelligence Group at the Luleå University of Technology, Sweden. His research interests include deep-learning algorithms, multi-agent systems, mapping and localization.

Anton Koval is an Associate Senior Lecturer at the Robotics and Artificial Intelligence Group at the Department of Computer Science, Electrical and Space Engineering at Luleå University of Technology. After receiving his PhD degree from the National Technical University of Ukraine “Kyiv Polytechnic Institute”, Ukraine, in 2012 he continued his work in academia with the focus on design and control of Unmanned Aerial Vehicles (UAVs). Currently, he has more than 9 years of experience in R&D of UAVs and their applications. In 2018 he received the Swedish Institute Visby scholarship for senior scientists for 6 months at Luleå University of Technology, Sweden and since then is working in the Robotics and AI team. He has been involved in the technical implementation, scientific or overall management of several European and National R&D projects. His research interests are mainly focused on control of legged and aerial robots, path planning and collaborative multi-agent area exploration.

Tore Lindgren received the Ph.D. degree from the Luleå University of Technology, in 2009. He is currently a Senior Researcher with Ericsson Research. His research interests include radio communication for UAVs and radio-based positioning.

George Nikolakopoulos was affiliated with the NASA Jet Propulsion Laboratory (JPL), Pasadena, CA, USA, for contacting collaborative research on aerial planetary exploration. His team participated in the DARPA Grand Challenge on Sub-T Exploration with COSTAR1 Team of NASA, where we won the Second Stage of the Competition, in February 2020. He currently works as a Chair Professor of robotics and artificial intelligence (RAI), while heading the Robotics Team with the Division of Systems and Interaction, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Sweden. He is a member of the Board of Directors at euRobotics, a member of the Scientific Council of ARTEMIS in the field of robotics and AI, and a member of IFAC TC on robotics. He is an Elected Expert of the Permanent Working Group (PWG) of A.SPIRE with a focus on process optimization and ultra carbon coal, and also an Elected Member on the Aeneas-XECS in embedded control systems.