Ground Robotics in Tunnels: keys and lessons learned after ten years of research and experiments

D. Tardioli [1,2], L. Riazuelo[2], D. Sicignano[2], C. Rizzo[3], F. Lera[2], J. L. Villarroel[2], and L. Montano[2]

[1]Centro Universitario de la Defensa, Zaragoza, Spain
dantard@unizar.es

[2]Instituto de Investigación en Ingeniería de Aragón, University of Zaragoza, Spain
{riazuelo, doxsi, lera, montano, jlvilla}@unizar.es

[3]Eurecat, Centre Tecnològic de Catalunya, Barcelona, Spain
carlos.rizzo@eurecat.org

Abstract

The work reported in this article describes the research advances and the lessons learned by the Robotics, Perception and Real-Time Group over a decade of research in the field of ground robotics in confined environments. This research has primarily focused on localization, navigation, and communications in tunnel-like environments. As will be discussed, this type of environment presents several special characteristics that often make well-established techniques fail. The aim is to share, in an open way, the experience, errors, and successes of this group with the robotics community so that those that work in such environments can avoid (some of) the errors made. At the very least, these findings can be readily taken into account when designing a solution, without needing to sift through the technical details found in the papers cited within this text.

I. INTRODUCTION

The use of robots in intervention and exploration in underground environments is gaining more prominence. Most of the proposed works, however, focus on a partial view of problems, such as mechanical aspects, Simultaneous Localization and Mapping (SLAM), or other specific aspects of the intervention tasks. Many of the problems in the field, such as localization, navigation,

Fig. 1: Three ActivMedia Pioneer P3AT deploying in the Somport Tunnel, an old railway tunnel between Spain and France used as the main experimental scenario in this work. It is 7.7 km long, with 17 lateral galleries, is poorly illuminated, has a relatively even floor and low-textured walls. There is no communication infrastructure. Each platform incorporates a SICK LMS200 LIDAR and a wireless device, Ubiquiti Networks SRC 802.11 a/b/g PCMCIA, based on the Atheros 5004 chipset. The onboard computer is an Intel Core 2 Duo T7250/2 GHz.

communication, and, above all, the integration of the same, remain open challenges to be addressed.

Search and rescue, or robotic intervention operations, for example, cannot typically rely on pre-constructed maps, and sometimes the environment is so challenging that common localization techniques (i.e., scan-matching, visual SLAM, etc.) do not function as intended. Navigation techniques, besides dealing with not completely reliable localization must consider that in such scenarios, the floor may be irregular, slippery, or contain potholes of water, which occur during the construction of tunnels. This can jeopardize not only the computed trajectory, but also the physical integrity of the robots. Communication is an additional challenge itself. These special characteristics, detailed in Section II, must be considered during the development of robotics applications in tunnels and gallery scenarios.

A. Related Work

A number of real-world applications in confined environments have been proposed in the last few years. However, few robotics systems have been reported to be capable of managing large distances. Yet, there has been a valuable effort in mapping underground settings in a semi-autonomous manner, as in [Zlot and Bosse, 2014] where a 3D map of a 17 km-long mine was obtained using two 2D perpendicular LIDARs, or in [Leingartner et al., 2016] where different sensors and well-established off-the-shelf techniques were used to map a 1.5 km-long tunnel disaster scenario to evaluate its usefulness for first responders. One of the general conclusions obtained from this study was that current mapping techniques are not yet robust enough and fail because of the low number of identifiable features. In [Jacobson et al., 2018], a monocular visual SLAM for vehicles moving inside an underground mine is proposed. However, the system is semi-supervised, requiring a manual selection of the relevant topological information. The experimentation is performed inside two parallel mine tunnels 300 meters long, but autonomous navigation is not reported. When it comes to autonomous robots, the experiments usually cover

much smaller distances. In [Bakambu, 2006] and [Losch et al., 2018] stretches of 200 m and 150 m, respectively, were covered in underground mines. An autonomous navigation system was proposed in [Nsasi and Vladimir, 2007] based on a previous semi-supervised surveying phase, using two perpendicular 2D LIDAR. The reported experiment was conducted inside a 150 m long mine tunnel. This is not the case of [Loupos et al., 2018] that presents the ROBO-SPECT, a platform capable of performing an autonomous tunnel inspection over a stretch of 3.5 km. However, it relies on artificial landmarks to be placed 15 m apart from each other. Aerial robots are also a promising technology for underground work (see for example [Özaslan et al., 2017] or [Mascarich et al., 2018]), however their limited battery life strongly restricts their use in large scenarios.

On the other hand, the problem of communication is rarely addressed. An exception is [Liu and Oh, 2012] where the authors analyze multi-hop communication in subterranean lane-ways, but only in a lab where small-size experiments are presented. The use of teleoperated robots into the Fukushima nuclear power plant was analyzed by [Nagatani et al., 2013], where the importance of a reliable communication between the operator and the robot system was highlighted. Despite the extensive work carried out over the last decade in the field of distributed robotics and robot communication with mobility restrictions [Hsieh et al., 2008], [Dixon and Frew, 2009], [Tardioli et al., 2010], how to deal with special propagation features (fading, wave guides, abrupt vanishings) of a signal in confined environments has received little attention.

Recently, the DARPA Subterranean Challenge Initiative [Chung, 2018] was launched under the observation that current technologies fail to provide rapid mapping and persistent situational awareness. The focus was to improve perception, autonomous navigation, mobility, and networking in three different underground scenarios: tunnels, urban areas, and caves.

B. Contribution

The Robotics, Perception and Real-Time group (RoPeRT) of the Aragón Institute of Engineering Research (I3A, University of Zaragoza, Spain), has been working on these types of scenarios and challenges for more than a decade (see Fig. 1). This work has been mainly focused on autonomous navigation, localization, and communications in large scenarios, using wheeled ground robots equipped with only 2D LIDAR scanners. The objective of this paper is to present the lessons learned through errors and successes, problems encountered, ideas, and reasoning to the robotics community.

Specifically, in a chronological fashion, we expound on our research from the first experiments in 2006 to establish a camera link between two ends of a tunnel, to the last experiment in 2018, in which a framework for robot intervention was successfully tested in the 7 km-long Somport Tunnel (see Fig. 1). The paper will address the aspects of communication, localization in large settings, and autonomous navigation found during our long-term research in the tunnels and other confined environments. The goal of this research paper is that the reader not only understands the solutions, but also what led to the choice of those solutions over others, supported by a ten-year research project that produced over 40 field experiments and more than 20 publications, some of the most relevant coping with the challenges of real-time communication ([Tardioli and Villarroel, 2007]), radio signal propagation ([Rizzo et al., 2013b], [Rizzo et al., 2013a]), localization based on topological information ([Romeo and Montano, 2006], [Tardioli and Villarroel, 2014]), communication-aware navigation ([Tardioli et al., 2010], [Rizzo et al., 2013c]), cooperative navigation [Urcola et al., 2008] and robotics intervention system where those proposals are put together ([Tardioli et al., 2012], [Tardioli et al., 2016a]).

In the following section, the main challenges in robotic tunnel applications are explored in-depth. In Section III, the evolution of the research over the years is presented, highlighting the main experiments and discoveries. Section IV summarizes the lessons learned and the main conclusions of our work. Finally, Section V presents the future work that we intend to conduct as continuation of that presented in this paper.

II. CHALLENGES

This paper will focus on the specific case of robotic intervention in tunnels, galleries (subterranean passageways) and similar confined places. Even in this niche application, there are many of possible situations to consider. From the maintenance operations of an in-use road or railway tunnel, to inspection works in construction galleries (see Fig. 2), to more stringent exploration or even search and rescue interventions in disaster scenarios in tunnels, sewers or mines. All these situations require a variety of solutions. These environments have become increasingly important in recent years, as evidenced by the DARPA Subterranean Challenge initiative [Chung, 2018].

In [Murphy et al., 2008], the robots that participate in search and rescue tasks are defined as extensions of responders to a disaster, providing real-time video and other sensory data. This definition is quite exact but does not show the extent of what a team of intervention robots must encounter. To be able to sense and/or send real-time video or data, the (team of) robot(s)

must be capable of reaching the scenario requiring intervention, deploying and establishing a communication network, among other things. These tasks become more difficult in the case of GPS-denied confined environments (tunnels, mines, and caves), where the large dimensions and the darkness make the problems even worse. Regular or emergency intervention in tunnels and galleries always poses severe risks for human beings, thus, the robotics research community has tried to replace humans with robots for these dangerous tasks. Personnel and robots must be protected against the relevant environmental threats, must be able to move through the debris in case of collapse, and must be able to communicate over relatively long distances in an infrastructure-less environment.

A. Challenges of Confined Environments

Working in confined environments, especially in long tunnels and galleries, poses a set of non-trivial challenges due to dimensions, the lack of proper illumination, and the lack of discriminatory elements that can aid localization.

In this kind of environment, the mission will be similar to that described in [Tardioli et al., 2012]: a team of robots deploys inside a tunnel, either autonomously, teleoperated by a human, or operated at a mid-point in between these two cases. The purpose could be to look for survivors, injured persons, or simply information while sending feedback to the base station located at a safe point inside or outside of the confined environment. The worst-case is a scenario in which: 1) no communication infrastructure exists or it has been destroyed, 2) no illumination system is present, it is poor, or does not work, 3) the floor is irregular and/or presents blockages and/or obstructions, 4) the tunnel walls are non-textured and no (artificial) landmarks can be identified, 5) smoke and/or suspended dust hinder perception, and 6) there is fire, water, or nuclear radiation that can disable or damage the robots. These issues create difficulties in several of the aspects involved in the development of a robotics team, such as localization, the establishment of a communication network, and navigation. In the next paragraphs a detailed discussion is presented.

B. Communication

Wireless communication has undergone extraordinary progress over the last few years. The IEEE 802.11 protocol, to quote an example, has reached rates of several gigabits per second in its most recent version. However, these bandwidths can only be reached in specific environments (a home or office) only when the distance between peers is relatively small. For example, the
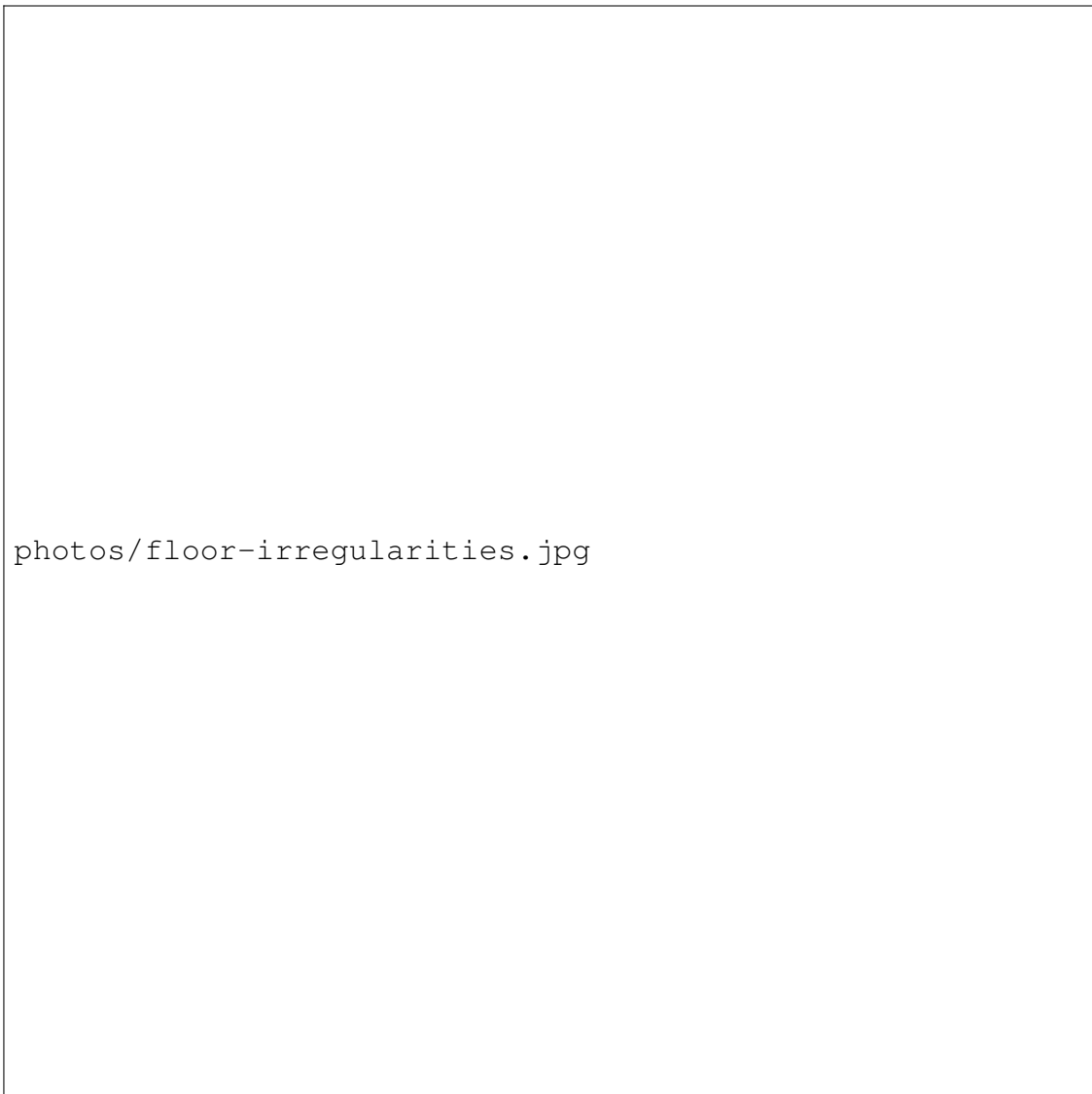
Fig. 2: A tunnel under construction (Spain). It is 1.2 km-long and the scenario changes because it is under construction, has bad illumination, uneven terrain with many obstacles, no communication infrastructure, and dangerous for humans (many vehicles moving around, risk of collapse, etc.).

802.11g protocol provided its maximum rate (54 Mbps) only for peers up to 45 meters away when indoors [Kapp, 2002].

On the one hand, communication in large environments, like those considered in this article, must link the different robots and the base station to the exploring units, which can be several kilometers apart. In practice, this makes it impossible to use tethered or Access Point-based communication, which requires all the robots to be in the same collision domain. Instead,

multi-hop communication, where intermediate robots act as relays to transmit the information from the source to the destination, is necessary. However, multi-hop communication has several drawbacks: 1) in the worst-case, the available bandwidth is reduced to roughly $1/(n-1)$, where n is the number of robots involved in the communication [Ng and Liew, 2007], 2) the routing protocol must be able to manage fast topology changes due to the robots' displacements, and 3) every additional hop increases the delay and jitter associated with the information transmitted, making control inefficient and inaccurate [Tardioli et al., 2018a]. While the first issue can be taken into account beforehand, planning the bandwidth demand to always be below the system capability, and the second can be solved using specific routing protocols, the third requires further consideration. Intervention tasks rely heavily on network behavior and efficiency, and this, in turn, depends on the delay and jitter that the underlying communication techniques/protocols are capable of offering. Having a high delay in communication means that interactive activities, that is, those that can be described as perception-actuation loops (with or without a person in the middle) become difficult and sometimes impossible, as the existence of a delay in the control-loop can easily make the system unstable; examples include teleoperation or voice communication. The jitter, which is defined as the deviation from true periodicity of a presumed periodic signal, can provoke similar issues, especially in tight control loops, as, for example, those needed to navigate around an obstacle. Moreover, the radio wave propagation in tunnel-like environments (much longer than wide passageways with uniform cross-sections) is substantially different than that in free space or other confined settings. This is due to the unavoidable strong multipath propagation– with multiple reflections on the tunnel walls, floor, and ceiling– and to the fact that, depending on the wave frequency and the cross section dimensions, the tunnel can act as an oversized dielectric waveguide (see for example [Emslie et al., 1975], [Delogne, 1991], [Lienard and Degauque, 1998] or [Rizzo et al., 2013c]). This will be explained in more detail in Section III-G.

C. Localization

While human beings do not necessarily need an exact localization in an environment, as they can also generally manage with topological information, a robot needs to be and stay precisely localized at all times to be autonomous or semi-autonomous. This was one of the first major concerns for the robotics community, and a large number of techniques have been proposed and implemented with amazing results. In fact, a growing number of researchers believe that the

SLAM problem has been adequately solved [Frese, 2010]. Solutions like [Grisetti et al., 2007] and [Serafin and Grisetti, 2015] have also been implemented and made available to the robotics community. Each performs well for localization with pre-built maps and/or mapping at the same time as localizing (SLAM).

These solutions rely on scan-matching-based techniques (generally using LIDAR data), occasionally on odometry, and, in the case of pure localization, on a well-built map. However, as shown in [Tardioli and Villarroel, 2014], all these methods fail when the environment is smooth and does not have any landmarks (irregularities, lateral galleries, and so on), which the cited scan-matching techniques rely on, as will be shown in Section III-N.

A relatively recent alternative to using LIDAR-based mapping are visual SLAM techniques. Proposals such as PTAM [Klein and Murray, 2007] or ORBSLAM2 [Mur-Artal and Tardós, 2017] do not rely on physical landmarks, but instead on visual descriptors that capture features (change of color, luminosity, etc.). These techniques are capable of providing online localization with errors of less than a few centimeters and are getting more robust every day. However, a proper and homogeneous illumination of the environment is necessary to obtain optimal results. Moreover, these techniques do not perform especially well in low-textured environments, where they encounter difficulties in choosing useful descriptors. Unfortunately, in disaster scenes or tunnel-like environments, the lighting is usually poor, and, in the best case, the robot itself carries the light onboard, creating nonuniform illumination throughout the scene; the movement of the light due to the terrain irregularities makes it even more difficult to localize. Figure 3 shows an attempt of localization inside a straight tunnel using the ORBSLAM2 algorithm, which summarizes the problem. Similar conclusions are discussed in [Jacobson et al., 2018], where a semi-supervised SLAM solution based on a monocular system and on a topological map which uses frames of relevant places manually selected beforehand is implemented. The authors propose in [Yang et al., 2017] a plane-based SLAM technique with a monocular camera to improve the localization and mapping in indoor low-texture environments applied to a 60 m long data set with loops. As the low-texture features are planar walls, this technique cannot be applied in underground scenarios were usually the walls are irregular.

D. Navigation

The main difficulties concerning navigation in such challenging environments are the uneven terrain and the large dimensions. The first is not explicitly taken into account in this paper,

given the moderate irregularities found in our testbeds. The second is a direct implication of the fact that the localization is usually unreliable. Making a robot autonomously traverse a tunnel without having a reliable map and without crashing against the walls, is far more complicated than intuition suggests. For example, concepts like 'going straight centered in the tunnel' are not easily translated to robot commands if the global localization is unknown or highly inaccurate. Classical global planner algorithms are, in general, no longer directly applicable. In these cases, it is necessary to implement alternative algorithms for navigation, such as wall- or corridor-following (e.g., [Pasteau et al., 2016]) based only on local information or specifically taking into account the walls, crosses, and lateral galleries that the robots encounter while navigating [Tardioli et al., 2016b] or just relying on relative localization [Alonso-Mora et al., 2018]. On the other hand, the walls play a double role for the obstacle avoidance algorithms: they can help defining navigation goals and also represent obstacles to avoid. In this sense, classic obstacle avoidance techniques (e.g., Obstacle-Restriction Method (ORM) [Minguez, 2005] or Dynamic Window Approach (DWA) [Fox, 2003]) must be adapted to this kind of environment.

E. Other Challenges

Working in the cited environments also presents other kind of challenges, such as floor irregularities, the presence of smoke and dust, and the need to deal with fire or radiation, among others. These aspects have not been dealt with specifically in this work, as they were out of the scope and possibilities of the established missions and experiments, though they deserve some comment.

The first is briefly analyzed in Section V, being the object of our future work. The presence of dust is common in tunnels and galleries, while smoke in a fire search and rescue scenario is an expected consequence. According to our experiments, dense dust or smoke jeopardizes LIDAR readings, rendering the sensors useless. RADAR may be an alternative for short-range operations through dust [Brooker et al., 2007]. However, its resolution may be insufficient for robot navigation. In case of fire or radiation, the most serious problem is how to maintain the robots operational. The fire can damage the robot, while the radiation can melt the wiring, for example. There are no simple solutions for these issues other than building robots strong enough to survive the specific circumstances.

On the other hand, other technological or logistic aspects, such as the limited power source of the robots used for longer-term deployment or environments with multiple branches that require

exploration, should be taken into account.

III. SETTING UP AN INTERVENTION TEAM

Over the last ten years many of the cited problems were examined and analyzed by this research group to find viable solutions for problems related to communication, navigation, and localization. While much work remains to be done, we believe it is useful and collaborative to offer our findings to the robotics community. In the next section a chronological story of the evolution of our research is presented, describing both successes and failures in the hope that others will avoid making the same mistakes.

When this research began, it had one primary concern that is often put aside by the robotics community: communication. However, this is one of the most important aspects to consider if the objective is for the intervention to be applied to a large scenario. Yet, the existence of a communication infrastructure is usually taken for granted in the robotics community, where it is typical to assume very high bandwidth availability no matter the scenario, instantaneous propagation, or exact ranges of communication. However, as we progressively discovered, this is simply not the case in the real world, where the exact quality of a link, for example, is highly unpredictable and dependent on several factors, such as distance, environment, and the presence/absence of a line-of-sight.

On the other hand, we did not expect to encounter great problems in other aspects, such as navigation or localization, given that most of them *seemed* to have been already solved by the robotics community. The first experiment conducted then was an attempt to establish a multi-hop link in a confined real-world scenario.

A. *The First Experiment: Tunnel-Wide Camera Link*

Performed in July 2006, this experiment's objective was establishing a video WIFI-based link between the two ends of the Manzanera Tunnel (see Fig. 4). This was done by means of a robot team composed by two ActivMedia Pioneer P3AT robots running the unaltered manufacturer's software, where the leader robot (carrying the camera) had to reach the far side of the tunnel supported by a second robot that acted as relay.

The experiment was very basic. The first robot had to start moving forward following a goal fixed to the end of the tunnel, and when it reached the halfway mark (300 m), the second robot had to start moving while also acting as a relay to the base station. The relay robot would then

be manually stopped once it reached the halfway mark of the tunnel. Once the leader robot reached the far end of the tunnel, the operator could take control and move the robot with a joystick while receiving LIDAR feedback and having the possibility of activating a camera.

From a communication standpoint, a simple fixed-topology chain network was set up: the base station sent data to the relay node (goal, joystick command, etc.) and the latter propagated it to the leader robot. In the reverse direction, the leader robot sent laser readings and camera images to the relay and the latter to the base station. All communications were based on ad-hoc connections among peers; the routing, which was static and configured to replicate the chain, was implemented to operate at the application level on top of the UDP protocol.

The localization was based on a simple particle filter, and the navigation, just reactive, on an implementation of the DWA [Fox et al., 1997]. Both algorithms were available in ARIA SDK [1], provided by the robot manufacturer (Mobile Robots, formerly, ActivMedia). The experiment was considered successful, as the robot did reach the far end of the tunnel, camera images were received, and the team teleoperated the leader robot. There were, however, a few issues found. From a communication standpoint: 1) the laser feedback and joystick commands suffered from intermittent disruptions, especially when the camera was activated, 2) images and LIDAR readings were sometimes received in bursts, and 3) the joystick commands were sometimes executed as if they were received in bursts, also causing occasional crashes. Analyzing the logs, it was discovered that the connections between the three nodes were not stable due to the long distances between them (about 300 m) causing heavy packet losses. The network rate changed frequently during the experiment and the laser readings, images, and commands were received in bursts, despite having being sent with constant periods. On the other hand, while the localization was quite precise in the y-axis, the particle cloud dispersed dramatically on the x-axis, offering poor longitudinal localization supplied, to some extent, by the odometry that worked reasonably well. Several lessons were learned by this experiment. For example, we began to realize that localization is difficult in this environment. The poor localization in the x-axis was due to the lack of features for scan-matching– the tunnel walls were completely smooth– to associate consecutive LIDAR readings. The navigation behaved better as assigning goals on the tunnel axes was effective due to the y-axis localization being precise.

From a communication standpoint, we realized that it was absolutely necessary to constantly

---

[1]http://robots.mobilerobots.com/docs/api/ARIA/2.9.0/docs/index.html

check the link between each pair of robots to avoid packet losses. Assuming that the link quality will vary as a function of the distance, as in common indoor scenarios, was not, in fact, a sound assumption. The strong multipath and guiding effect of the tunnel geometry had a strong influence, making it necessary to perform an online check of the signal strength.

It was also found that it was necessary to establish an *importance* relationship among the flows exchanged. For example, flows representing joystick commands or LIDAR feedback from the robot are certainly more important than camera images or information about the battery level. This is true, at least, in the sense that these latter are equally useful if they arrive a few milliseconds later or with a reduced rate, while the former, generally, are not. It was necessary to prioritize control flows, while discarding less important information, if needed. The problems found could not be solved with the scheme used in this experiment, hence, a solution capable of taking messages or flow priorities into account was needed. Unfortunately, common communication protocols do not provide such mechanisms. Thus, we began investigations on two different fronts: how to obtain the link quality and how to implement priorities at flow level.

B. Introducing Real-Time Communication

The first aspect was, conceptually, resolved quickly. The Received Signal Strength Indicator (RSSI) is a value that network cards provide to the kernel, reporting the strength a frame has been received with at a Media Access Control (MAC) level. The kernel uses (a function of) this value to modulate the network rate used during communication. This (roughly) describes the expected link quality with the sender assuming the link is symmetric. The idea was to check the RSSI to stop the movement of a robot (or to force another to move) when that value fell below a certain safety threshold. The problem consisted of how to extract this value for each received frame and provide it to an application level, where it would be used in the decision to stop/move the robots. The solution was found modifying the network card driver to provide such values to the application.

The second aspect, the priorities, required much more work and several months of testing and development. After a thorough literature review, we discovered that a wireless protocol capable of managing node mobility and supporting flow priority did not exist at the time, and only a few, partially-fitting proposals had been implemented. Hence, a new protocol was designed and implemented from scratch. The inspiration came from the RT-EP [Martínez and Harbour, 2005], a token-passing protocol for Ethernet LANs capable of supporting priorities and real-time

message exchange. We designed a similar scheme for wireless networks with the very noticeable difference, in that a routing scheme was added (which is not needed in LANs) to manage the frequent situations when not all nodes are in communication range of one another.

The protocol, named Real-Time Wireless Multi-Hop Protocol (RT-WMP) [Tardioli and Villarroel, 2007], offered real-time guarantees on the delivery of the messages (in the worst-case, the delivery delay is bounded and known) and, similarly to RT-EP, supported global priorities at message level.

The RT-WMP works in three phases, guaranteeing deterministic access to the medium (see fig. 5). In the first phase, called the Priority Arbitration Phase (PAP), a token travels throughout the whole network collecting information about which node holds the highest priority message at that specific moment. Notice that this is needed to guarantee a fundamental *global priority ordering*: the highest priority message *in the network* is the one that must be sent first. However, implementing priorities separately in each node is not enough because if two nodes generate only one flow each, for example, both will always be competing no matter the priority of the messages. The scheme used, instead, guarantees that higher priority messages can be only delayed by *one* lower priority message (that can not be preempted). When the token has reached all the nodes, the last one starts the second phase, called the Authorization Transmission Phase (ATP), in which it authorizes the node that holds the highest priority message to transmit. This is the last phase, called Message Transmission Phase (MTP), where the message is sent to the destination completing the so-called *loop*. Note that during all phases, the scheme must deal with a topology that is not necessarily completely connected. The routing is based on a connectivity matrix called, Link Quality Matrix (LQM), which contains information about the link quality between each pair of nodes, and is shared among the nodes during the PAP. This information is computed as a function of the RSSI, and is obtained as explained above. During the PAP the LQM is used to choose a safe path to visit all the nodes while, in the ATP and MTP phases, to compute the safest path to the destination using Dijkstra's algorithm [Dijkstra, 1959].

From a global point of view, the phases of the protocol repeat with worst-case duration $t_{PAP} = (2n-3)t_t$ for the PAP phase, $t_{ATP} = (n-1)t_a$ for the ATP phase, and $t_{MTP} = (n-1)t_m$ for the MTP phase. The values t_t, t_a and t_m are the duration of a token, authorization, and message pass, respectively, as shown in *Loop 2* of fig. 5. The absolute values of t_t , t_a and t_m depend on protocol parameters, such as the number of nodes, the data rate, the maximum transmission unit (MTU) that the network must carry, as well as on the underlying 802.11

protocol, but are globally bounded. The worst-case end-to-end delivery is thus:

$$t_{etoe} = (2n - 3)t_t + (n - 1)t_a + (n - 1)t_m$$

More details about the protocol, which is also available as a Robot Operating System (ROS) package,[2] can be found in [Tardioli and Villarroel, 2007].

C. Experiments at the Somport Tunnel

After months of experiments and simulations to refine the implementation and operation of the RT-WMP, a second field test was performed in the Somport Tunnel (see Fig. 6), a 7.7 km straight tunnel that connects Spain and France through the central Pyrenees.

The goal was to perform a similar experiment, this time with three robots, focusing principally on communication. Localization was not the main objective this time. The Somport Tunnel has small shelters (1.8 x 1.0 x 1.7 m) every 25 m, and it was assumed that this would have helped the particle filter to reduce the localization uncertainty of the robots at small regular intervals.

Surprisingly, this assumption was wrong. The partial shape of the shelters was not picked up by the laser sensor during movement and was not always sufficient for the localization algorithm to find a match on the map, especially for distances larger than 10-15 m, causing a dispersion of the particle over the x-axis. While most of the time, the algorithm was able to recover the localization when reaching a shelter, sometimes the robots got completely lost. In these cases, the navigation also failed.

We tried to enlarge the covariance of the filter but the limited memory and processing capabilities of the computers onboard the robots did not allow for much improvement. Additionally, the huge size and complexity of the grid-based map, which covered a 3.3 km stretch of the tunnel, was barely manageable with the scarce amount of memory and limited computation capabilities available at that time, provoking unexpected increase of grid width and frequent memory overload which, in turn, resulted in more than one robot crashing.

Communication did not go much better. Although the goal positions of the robots were chosen to establish a chain and to force the RT-WMP routing algorithm to relay all the messages from the base station to the leader node using two intermediate nodes, it did not happen all the time. Instead, many of the packets went from the base station to the leader directly or from the first

---

[2]https://github.com/dantard/unizar-rt-wmp-ros-pkg

robot to the last, and so on, according to the logs. It was later discovered that this was due to the high variability of the RSSI. While outdoors, the RSSI between two communicating nodes is stable (it can vary about 2-5%), we discovered that the RSSI is much 'noisier' in tunnel-like scenarios due to the fading effect that can cause a variance of up to 30-40%. This caused that, frequently, the link between adjacent nodes was reported to be worse than that between nodes further apart, which resulted in suboptimal paths chosen to route the messages. Additionally, it was discovered that with the implementation we had at that moment, the algorithm preferred a one-hop poor link over a multiple-hop safer link. In both cases, this provoked heavy packet losses and poor network performance. After this experiment, the following actions were taken: 1) replaced the localization algorithm to reduce the computation requirements and the memory needs, and 2) further improved the RT-WMP to allow more fluid communication.

D. Improving Localization and Communication Links

Because the large grid maps were unmanageable by the computers available at that time, it forced a move to another map-based localization technique. An algorithm was chosen in which the map is composed by a set of lines representing the edges of the environment. Although it would not actually solve the localization problems in featureless stretches, it would allow us to localize the robot in real-time without using large amounts of memory [Lazaro and Castellanos, 2010]. This localization technique, developed by the RoPeRT group, implemented a LIDAR data segmentation technique based on a tracking-like algorithm, grouping the range readings into regions. These regions were successively split into segments for obtaining a metric representation of the environment. Every time a new LIDAR reading was available, the algorithm attempted to match the features obtained with the existing map segments.

On the other hand, the RT-WMP was improved in following two ways: the raw RSSI values were replaced with a moving average, and the basic routing algorithm was enhanced by computing the routes over a reduced graph obtained applying Prim's algorithm [Prim, 1957]. Specifically, instead of using the graph reported by the LQM to compute the optimal path, an extra step was introduced: the Minimum Spanning Tree (MST). This was computed first on the graph described by the LQM to filter out bad links, while guaranteeing that the network representation was still connected. The Dijkstra algorithm was then applied over such a tree (see Fig. 7). This guaranteed that no unreliable links were included in the path. Although this can

favor the use of longer paths, the worst-case delivery delay was still guaranteed to be the same [Tardioli and Villarroel, 2012].

E. Communication-Driven Robot Deployment

The next experiments showed clear improvements. The new routing configuration gave the expected results with a drastic increase of the Packet Delivery Ratio (PDR). On the other hand, the localization showed increased efficiency but still offered unreliable results in the stretches between consecutive shelters.

The next step consisted of automating robot deployment according to the communication constraints. The robots must be capable of deciding when to stop movement to act as relay during the deployment, basing the decision on the link quality between adjacent nodes. A simple algorithm was implemented. The leader robot starts moving, and when the link quality with the closest follower falls below a certain threshold, the latter starts moving to avoid losing the connection. It will move until the link with the second follower starts degrading, which is where the follower will start moving, and so on. The algorithm continues until the maximum distance is reached (given that the base station cannot move), unless the leader robot is stopped manually, in which case all the followers also stop once the link qualities go above the threshold.

Notice that, for the first time, we established a dependency between communication and navigation.

The first approach was to assume that, besides the variations filtered out with the moving average, the relationship between link quality and distance was monotonically decreasing. A configuration was setup based on the described scheme and a fixed threshold that guaranteed a PDR close to 100%, according to a measuring campaign previously made to establish the relationship between RSSI and PDR (see Fig. 8). As opposed to the open space experiments where the communication range reached 150 m at most, it was easy to reach communication ranges of 300 m, or even 400 m, almost without packet losses in the tunnel. However, even with these conditions, several problems were found with the proposed scheme.

A certain level of instability and unpredictability in the robots' movements was detected. For example, when the link between the leader robot and its closest follower fell below the selected threshold and the follower started moving, two different outcomes were possible depending on the relative speed of the robot: 1) the link quality was quickly reestablished and the follower stopped for a while, then the leader robot got far again and the follower restarted moving, and

so on. A similar situation occurred with the other followers; or 2) due to the different speed of the robots, the leader continued getting farther and farther from the follower, and the connection was finally lost. Note that having a different speed can be common if, for example, a robot must avoid an obstacle or if the floor is in better or worse condition in various stretches of the tunnel.

F. Introducing Virtual Spring-Damper Systems

We realized that, to avoid this kind of behavior, the interaction between consecutive elements must be bi-directional. This means that the leader robot must check if the link has actually been reestablished before continuing to move. It is also not advisable to have a single fixed threshold, but instead have the movement begin when the link quality starts to deviate from a safe level. We came up with the idea of using virtual forces to connect the robots. It would have depended on the link quality: as the link quality decreases, the force would pull the robot that is getting farther, reducing its speed without necessarily immediately stopping it. To implement this behavior, we began investigating the use of virtual Spring Damper Systems (SDS). The objective was limiting the freedom of movement (speed) of the robots without resorting to an abrupt start/stop method while preventing network split.

We came up with the idea of taking advantage of the spring-damper model used to coordinate the motion of flexible robot formations first developed in [Urcola et al., 2008]. This work was then adapted in [Tardioli et al., 2010] for maintaining network connectivity in robot teams, replacing the relative distance between each pair of robots with a measurement of the link quality between them. In this scenario, we were pursuing a similar goal: maintaining the connectivity of a chain formation. An external virtual pulling force ($\mathbf{G}_i$) was applied to the leader robot to allow its movement toward the goal (the end of the tunnel, for example), while the virtual forces generated by the movement of the robots maintained the connection of the chain network. In this scheme, the force $\mathbf{F}_i$ applied to each robot is computed as:

$$\mathbf{F}_i = \mathbf{G}_i + k_s \left(\gamma(s) - \gamma_0(s) \right) \mathbf{d_u} + k_v \mathbf{v}_{ij}$$

where k_s and k_v are the spring and damping coefficients, chosen to have a slightly overdamped behavior, $\mathbf{d_u}$ is the unit vector linking i, j robots, and $\mathbf{v}_{ij}$ is the relative velocity between robots i and j. $\mathbf{G}_i$ is the force from the goal for the leader and the force from the leader robot for the followers. These virtual forces were transformed into velocity commands for the robots. The function γ depends on the link quality s, which represents a measurement used to establish the

connection between each pair of robots when that quality decreased. The lower the link quality, the higher γ is. γ_0 is the rest value of γ.

Thus, the force generated by the SDS was not dependent on the distance between the robots but on the link quality between the robots. The range of the link quality was divided into three zones: safe, controlled and forbidden. When the link is in the safe zone, the robots can move freely. When it enters the controlled zone, an SDS connects two robots and a force is generated between them such that the lower the link quality, the stronger the force, equating the distance for a common spring with the inverse of the link quality. With adequate parameters, the link quality should not ever reach the forbidden zone (the one in which the robots can no longer communicate), because the robots would be stopped before. The damper, meanwhile, helped to avoid undesired oscillating behaviors. The key is that the SDS are created only when needed, leaving the robot free to move most of the time. To automate this behavior, the RT-WMP's LQM matrix was used, which offered a complete view of the global network state.

G. The Appearance of the Fadings

However, we realized that when the experiments were performed in larger settings, where consecutive robots reached distances of several hundred meters from one another, the spanning tree was not always a chain, and sometimes, the SDS were created between non-adjacent robots. The path followed by the message showed a similar behavior: it did not follow the chain, but jumped, for example, from the leader to the last follower or directly to the base station. Analyzing the logs, it was discovered from the LQM that, surprisingly, adjacent nodes had worse link quality– they were in the forbidden zone, in fact– than non-adjacent nodes. This was due to the appearance of *fadings*, a set of deep radio frequency (RF) signal drops to almost vanishing, that appeared in some places along the tunnel. This phenomenon is known to occur in tunnel-like environments, where the electromagnetic waves are scattered by the walls, floor, and ceiling, giving rise to a complex radiation field propagating along the tunnel. If the wavelength is much smaller than the cross-section dimensions, the tunnel behaves as an oversized dielectric waveguide, extending the communication range but affecting the signal with strong fadings, as shown in Fig. 10.

Any careful measurement campaign will reveal that this chaotic-like RF power distribution is reproducible [Rizzo et al., 2013b]. This was verified in several measuring campaigns we conducted during 2013 (see Fig. 11). Modal Theory (or ray tracing analysis) can be used to

understand this behavior[Emslie et al., 1975]. The RF field inside a tunnel can be described as the superposition of *modes* with different amplitudes. Each mode can be understood as a different periodic ray bouncing path and has a specific wavelength and attenuation constant. The energy coupled to each one depends on the emitting antenna position and orientation inside the tunnel. The field received anywhere inside the tunnel is the sum of all the interfering modes. In the positions where the interference is destructive, there is a strong signal decay, the so-called fadings. At first, the near and far sectors can be distinguished (see Fig. 10) [Dudley et al., 2007], [Zhou et al., 2015]. In the near sector, rapid fluctuations of the signal, or fast fadings, appear. However, the higher order modes attenuate faster than the lower order ones and farther away from the emitter, in the far sector, where only two or three modes survive, the RF field structure becomes periodic and clean. Real-world arched or horseshoe tunnels are usually modeled by means of an equivalent rectangular tunnel of width a and height b. The modal fields of rectangular tunnels are a good approximation, see for example [Dudley et al., 2007]. Figure 12 shows a qualitative picture of the electric field distribution of the three first modes in a rectangular tunnel.

For vertical antennas radiating at frequency f MHz, the period D in meters of the fadings appearing from the interference of the dominant mode ($m = 1$), and the higher order modes ($m = 2, 3...$) is computed as:

$$D = \frac{8fa^2}{75(m^2 - 1)}$$

The RF power inside the tunnel also shows a structure in the transverse dimension [Rizzo et al., 2013a]. For example, if only the two first modes are relevant, whenever a fading minimum appears in the right side of the tunnel, a maximum appears on the left side. The RF power also decreases toward the tunnel floor or ceiling, which is a relevant issue when UAVs operate within a tunnel.

For a given tunnel cross-section, the choice of RF frequency and the selection of appropriate emitter/receiver positions can be used to tailor the RF power distribution inside the tunnel [Rizzo et al., 2014b]. As a general rule, higher frequencies attenuate less but create a more complex fading distribution, and thus, a compromise choice is mandatory. In the Somport Tunnel, 2.4 GHz was a better choice than 5 GHz among the available off-the-shelf devices. Moreover, there is no particular need to use highly directive antennas, as the wave guiding effect of the tunnel geometry is far more efficient. Knowing where the fadings appear (or forcing their appearance

where desired), will ease navigation without coverage issues. This is a key factor when an intervention requires maintaining communication while optimizing the number of robots that act as repeaters. These procedures and theory can be used in other tunnel-like structures, such as sewers [Rizzo et al., 2018] or pipes [Rizzo et al., 2014a].

To carry out interventions in this environment, propagation patterns must be considered. This can be done in a variety of ways. The easiest is by maintaining adjacent robots in each other's near sectors, where the path loss is assumed to be a monotonically decreasing function of the distance. Another option is to consider the environment and the known repeatability of this phenomena. This can be done by (1) using two or more far-enough receiving antennas to take advantage of the fadings' transversal structure (spatial diversity) [Rizzo et al., 2016]; (2) using navigation strategies (S-like trajectories, for example) to avoid the fadings that appear in alternate sides of the tunnel [Rizzo et al., 2013a], [Rizzo et al., 2016]; or (3) by using a fading-aware coordinated deployment strategy, such as that presented in [Rizzo et al., 2013c] which allows seamless base-to-leader communication over a 4 km-long stretch of the Somport Tunnel with only two repeater robots.

These two possibilities pushed us to travel both ways, opening a new line of research in our group dedicated to the study of radio wave propagation in tunnel-like environments for robotics applications.

H. Managing Communication in Corners

In the light of this, we temporarily circumvented the fadings problem by restricting the distance of the robots to the near sector (which was verified to be about 700 m in the Somport Tunnel). This got us closer to having a system capable of conducting an intervention in a real-world setting. However, what if the intervention must be carried out in a lateral gallery rather than in the main tunnel? Or what if a communication needed to be established with victims who have taken refuge in a shelter? Until now, the only concern was deploying robots in the longitudinal direction of the tunnel, which may not be enough. To navigate inside lateral galleries is also necessary, while maintaining communication with the base station. A communication link worsens when the direct line-of-sight from sender to receiver is intercepted by an obstacle. This may be the case when a robot enters a lateral tunnel. What we did not know at the time was by how much the signal was affected. After a measurement campaign, we discovered that the RSSI falls abruptly as soon as the line-of-sight is lost, which often implies the loss of connection. Other authors [Zhou et al.,

2015] have also noticed the same effect, that is more pronounced for higher frequencies. The obvious solution to this problem is to maintain the line-of-sight between consecutive nodes. This means that it is necessary to put relay nodes in the corners to guarantee such a connection, as shown in Figure 13.

This is not trivial, given that it involves aspects of localization and cooperative navigation. The idea was for the leader robot to reach the lateral gallery, stop in front of the entrance, move the first follower closer to the leader, and then send the leader inside. In this way, the line-of-sight between the pair of nodes was never broken and the follower could act as a relay as in a normal chain. This process, however, had to be implemented on top of the scheme just described to guarantee that if additional relays were needed, each would be pulled up by the leader. For the first time, the algorithm not only included a reactive behavior (pulling/set free the robots depending on the link quality) but required to maintain a state. The leader must wait for the follower to arrive at the meeting point (the corner of the gallery) and then move toward the gallery. A simple State Machine (SM) was used to implement the described behavior. The SM was located at the base station. This implied that all the information necessary to change the state and make decisions must reach the base station itself. Therefore, additional flows were added to the set managed by the RT-WMP.

The experiments gave interesting results even though the tuning of the parameters to maintain line-of-sight between robots at all times was far from straightforward.

I. From Lazy to Eager Behavior

With the deployment scheme used, the leader robot had to wait for the follower over a long period of time. To overcome this problem, a new paradigm named *eager* was introduced, in contrast to the one used until then, which, in hindsight, was named *lazy*. In the eager configuration, all robots move in chain close to one other sharing an ahead goal, However, when the link between the last follower and the base station goes below the threshold, an SDS is created to avoid losing the connection. The same happens with the other followers. This scheme has similar implications as with the lazy scheme, and it is theoretically capable of covering the same distance, yet, it has the advantage of reducing the time needed to carry out the deployment. There are also drawbacks: it is less efficient from an energy consumption viewpoint in non-worst-case situations, and, in the presence of deep fadings, it runs the risk of getting all robots into a signal valley together, which would result in loss of communication with

the base station. After testing this scheme, another problem was discovered. Due to the different speeds of the robots, the followers have the possibility of overtaking the leader, or the followers may overtake each other. Thus, a precise relative localization became important to maintain the chain.

However, as mentioned before, the global localization was far from precise over featureless stretches, which implied an imprecise relative localization.

To solve this problem, reflective panels were installed on the back of the robots and raw LIDAR readings (which provided the intensity level registered) were used to obtain the relative localization among the members of the team (see Fig. 14(a)). During deployment, each follower received a goal a few meters behind the robot in front of it, which avoided any overtaking. If a robot stops or slows down, the robot behind will do the same. Although the range over which the LIDAR intensity level was reported was limited to less than 30 m, it was sufficient for this experiment.

In this way, the expected behavior was maintained, reaching the lateral gallery while maintaining connection to all the robots at all times, and especially between the leader robot and the base station. Figure 14(b) shows the two robots involved in the experiment: the leader approaching the shelter at the end of the lateral gallery and the follower standing at the corner, acting as a relay. Figure 15 shows the leader entering the shelter, while the communication is maintained with the base station 500 m away, through the follower robots in the chain. Figure 16 shows the RSSI registered during an experiment performed with two mobile robots and a base station, and reports the link quality between each pair of nodes. The two mobile robots ($R1$ and the leader, $R2$) start moving together; as expected the link quality between them is constant yet noisy due to fast fadings. Additionally, the link quality of both robots with the base station decreased slowly with the increasing distance. After 430 seconds, the leader robot entered the gallery, and the follower stopped itself on the corner. The signal between $R1$ and $R2$ fell from 80% to about 30% due to the effects of entering the gallery, however, it remained constant after, while the direct link between $R2$ and BS fell abruptly, making the direct connection impossible, thus, requiring a multi-hop communication, transparently managed by the RT-WMP protocol.

J. Adding Teleoperation

The robots were able to reach the location of a hypothetical incident even if it was in a lateral gallery, while maintaining communication with the base station. However, it would be useful

being able to move the leader from the base station (with a joystick, for example) while receiving sensor readings, to determine the actual situation.

This means that two additional flows needed to be incorporated, as in the first experiment: 1) movement commands (those generated by the joystick), and 2) LIDAR readings to observe the environment, and, above all, act as feedback to the operator during movement. The list of flows was updated and is presented in Table I. Figure 17 shows a screenshot of the user interface at the base station.

This change, that could seem just an addition of two flows, meant instead a profound conceptual change: we were converting the system in a feedback system with the (multi-hop) network and the person controlling the robots in the loop. The timing of the messages, at this time, became paramount. Unfortunately, multi-hop paths intrinsically introduce delay as the information must be relayed one or more times before reaching the destination. These relays also contribute to incrementing the jitter [Tardioli et al., 2018a] given the non-deterministic scheme used by common wireless protocols (like 802.11) to access the medium. The intrinsic low reliability of the wireless medium (at least compared with its wired counterpart) worsens the situation. The delay introduced can destabilize the control-loop in a way that can cause the crash of the robot. We were facing, in other words, the first actual test of the real-time capabilities of the RT-WMP protocol.

Two high priority flows were set up: one for joystick commands and one for leader LIDAR feedback separate from those for debugging, which were assigned the lowest priority.

The tests were carried out again in the Somport Tunnel in 2009. The results showed various problems related to the approach taken.

First, we figured out that sending direct velocity commands to the robot was not a good idea, unless the network could guarantee an ideal $100\%PDR$. In fact, if the communication

TABLE I: Flows involved in communication.

Qty	Name	Priority	Period (ms)	Direction
1	Joystick	11	50 (O/D)	BS $\rightarrow$ R
1	Laser	10	100	R $\rightarrow$ BS
3	Position	9	100	R $\rightarrow$ BS
3	Goal	8	O/D	BS $\rightarrow$ R
3	Debug	1	500	R $\rightarrow$ BS

fails, even during a small time-interval, the robot moves without any control and can crash against something, which it did. Network disruptions can be reduced as much as possible, but the possibility of a relay breakdown or sabotage will always exist. However, the robot must maintain control in that situation as well. The easy, fast, and classic solution is to use a watchdog. If no command data is received within a certain time interval, the robot stops. However, the disruption or delay can also affect feedback data. The operator can get into a situation, where the command sent corresponds to a different scenario than that provided by the feedback, with a similar outcome.

To avoid both issues, the commands sent to the robot were not velocity commands, but position goals. This means that the position of the joystick indicated an (x, y) coordinate location. Given that, the robots rely on the underlying obstacle avoidance system for displacement. With this scheme, the robot moves safely at any time.

A method to link perception and control was also implemented: a control command was sent only after feedback information was received and vice-versa. In this way, the command the operator sends corresponds to the situation that the robot (and the operator) perceives. However, this scheme has the drawback that if the feedback flow is blocked or is slowed down, so too will be the command flow. With these changes and several iterations, teleoperation was successfully integrated into the system.

K. Introduction of Voice and Video

Exploring the environment with only LIDAR feedback is somehow insufficient, especially if the operator is gathering information about the situation to prepare for human intervention.

It also turned out that at the end of each lateral gallery of the Somport Tunnel, there is a large shelter equipped with a survival kit, water, etc. It is likely that in the case of an accident or fire, people would take shelter there. It would certainly be useful to have a means of communication with those in the shelter. This then motivated the addition of a video flow from the leader robot to the base station, as well as a full-duplex audio communication between both ends. Thus far, the use of RT-WMP protocol provided management of real-time requirements of flow information for robot control. But, how will another class of flow, such as audio and video be managed? The first idea was to integrate audio and video as additional real-time flows with appropriate priorities. Testing this solution began in the laboratory using *Speex* audio and *Theora* video

codecs and forcing the network to have a chain configuration to evaluate the performance of the protocol in the worst-case situation.

The results were promising and it appeared that the protocol was capable of supporting the bandwidth required by the, now, eight flows (two per position, one per robot, joystick commands, laser feedback, video, and two-way audio), plus debugging (see Fig. 18). However, the lower PDR and bandwidth available in the field experiment due to the greater distances covered, resulted in disappointing behavior with audio/video messages, discarded or noticeably delayed, as well as the video flow being reproduced in fits and starts, while there was an irritating delay in the audio conversation. More concern was centered around the fact that both control and feedback messages were often discarded due to the limited size of the protocol transmission queue, which was already filled with unsent audio/video low-priority and old messages. Therefore, the RT-WMP protocol was extended to make it capable of routing multimedia flows without jeopardizing real-time traffic.

The rationale behind this idea was to take advantage of the time not used by the routing algorithm in the majority of situations. In a real-time system, like the RT-WMP network, planning must always consider the worst possible scenario, which is, in the case of RT-WMP, a worst-case PAP, ATP, and MTP with $2n - 3$, $n - 1$ and $n - 1$ hops, respectively (see Section III-B). However, worst-case loops are unlikely to occur in practice, even with unfavorable network topologies (like those used in our experiments). If a loop lasts for less than the worst-case, there exists an amount of time in which the protocol could be idle and still all the deadlines would be met. This time can be taken advantage of by sending non-real-time data without jeopardizing the real-time behavior of the protocol.

With this premise, an addition was implemented, named the Quality of Service (QoS) extension, with two main characteristics: no altering of the real-time features of the basic protocol and support of a variable priority for multimedia packets. Two extra phases, named QoS Authorization Phase (QAP) and QoS Message Transmission Phase (QMP) were added to the RT-WMP scheme (see Fig. 19). Each is executed only after non-worst-case loops and can be repeated several times until the maximum duration of a worst-case (plain) loop is reached. With this extension, during the normal PAP, the nodes not only reach a consensus about which holds the highest priority message, but also negotiates which has the right to send a (burst of) QoS message(s). After the MTP, the QAP is started and the winner of the QoS negotiation is authorized to transmit. During the QMP, the message is sent and the receiver re-authorizes the source by means of another QAP,

if there is enough time for another round (see [Sicignano et al., 2010] or [Sicignano et al., 2013] for details). The extension was successfully tested in field experiments showing that audio/video flows could be supported simultaneously with the real-time traffic.

L. The First Intervention Experiment

After putting all the parts together, the first complete experiment was conducted. Two Pioneer P3AT robots were used, equipped with onboard laptops and LIDAR sensors, along with running the mentioned line-based localization algorithm and RT-WMP with the QoS extension as the communication protocol. The goal, as shown in Figure 20, was that the leader robot reached a shelter in a lateral gallery autonomously, but supported by the other robot acting as relay. Then the robot would be teleoperated to enter the shelter and establish an audio and video communication with the people there.

Figure 21 shows the velocity of the leader robot during the experiment. The experiment was finally successful, but had to be repeated several times before achieving a satisfactory behavior. The main problem was the frequent loss of global localization due to the lack of features in the tunnel, especially in the stretches without shelter or galleries. Also, entering the shelter through a narrow door was not easy for the DWA obstacle avoidance system used, and it produced many problems that could not be solved by adjusting the algorithm parameters.

M. From DWA to ORM

To solve the problems we had in crossing doors, we decided to replace the DWA with the Obstacle-Restriction Method (ORM) [Minguez, 2005], an obstacle avoidance algorithm based on the Nearness Diagram method (ND) [Minguez and Montano, 2004], specifically designed for dense and narrow environments.

The ORM divides the space around the robots into sectors of one or two degrees, and analyzes which sectors are occupied by obstacles and which are not. The unoccupied sectors represent the possible pathways toward the goal. The algorithm identifies one of six possible states representing the relative situation of the robot and obstacles, thereby applying a specific action for each cycle (robot linear and angular velocities) of the situation. For example, if the algorithm detects that there are obstacles on both sides of the robot, it computes a movement that will move the robot away from both obstacles, ignoring the goal. When it detects an obstacle on the right, it moves the robots away from the obstacle, and then, when safe, it will direct the robot back toward the

goal. As opposed to other obstacle avoidance methods, the few ORM parameters to be tuned have a clear physical representation, are easy to adjust, and the actions applied do not depend on a balance of optimization criteria.

One of the parameters of this method is the safety distance, used to reduce the velocity of the robot when it comes close to an obstacle; it must be tuned according to how cluttered the environment is. However, we were facing a mixed situation: the tunnel has clear and narrow areas, but crossing a doorway is comparable to moving in a dense environment. Thus, having a small safety zone would hinder this last task, while having a large safety zone would have slowed down the robot during the displacement along the tunnel, such that the walls would have been inside the safety zone.

To deal with this issue, a simple modification of the ORM was made by introducing an extra ring around the robot, as visible in Figure 22. With this method, named ORM High Speed (ORM-HS), when an obstacle is within the ORM-HS safety zone, the maximum speed of the robot is reduced proportionally to the distance to the obstacle itself, until reaching the safety zone, where the standard ORM algorithm is applied. This solution allowed for reducing the safety zone, while maintaining a high level of safety.

N. Localizing without identifiable features

Demonstrating that a complete system could work was a good starting point. However, not all problems that a robotic team encounters in a mission have been solved. The one question that stood out was what to do if the tunnel does not have enough features to help the localization algorithm to match successive scans. In other words, will the localization work in a tunnel with completely smooth walls?

Figure 23 shows (x, y) position of a robot moving in a straight line inside a simulated scenario as the one just described and computed with different scan-matching algorithms. While the y-position is correctly computed, the x-position shows considerable error. GMapping, which is one of the best-performing SLAM methods, results in an error of 15 m, while the Hector SLAM is incapable of estimating any displacement. This was due to the fact that successive LIDAR readings can be successfully matched no matter the real x-axis displacement. The scan-matching algorithm must then guess what the real displacement has been and the best alternative to rely on the displacement reported by raw odometry (if available) or to assume that there has been no displacement at all.

This problem does not have a simple solution, and, to start with, invalidates all the scan-matching algorithms for continuous localization. While looking for other solutions, we came up with a peculiar idea: to move the robot in a zig-zag pattern. In this way, the x displacement may be obtained with a trivial calculation, as explained in [Tardioli and Villarroel, 2014].

Consider Figure 24. The robot is moving toward the left wall with an angle α. At a certain time, its distance from that wall is h_1. After a period of time (usually after the reception of the following laser reading) the distance from the wall is computed to be h_2, while the angle is still α. The x-axis displacement, x_d, in that period can be calculated as: $x_d = x_1 - x_2$ where $x_2 = h_2 \cdot tan(\beta)$ and $x_1 = h_1 \cdot tan(\beta)$ being $\beta = \frac{\pi}{2} - \alpha$. If the process is repeated in each cycle and the value of x_d is accumulated in a variable, it is possible to determine the global displacement of the robot. When the distance of the robot from the wall reaches a value below a certain threshold, h_{min}, the orientation is then changed to have the same approaching angle α to the opposite wall (that is, $-\alpha$). In a real application, the path does not have to necessarily be a zig-zag. The only necessary condition is that the robot must not move exactly straight. This scheme was aptly named, the Dizzy Robot Algorithm (DRA). The results were interesting and are shown in Figure 25. In terms of precision, the DRA incurs an error of 1.5%, overestimating the final position by 0.67 m, while both odometry and GMapping underestimated the ground-truth (1.6% and 5.5% respectively). However, we realized that for this method to work, it is necessary that the robot went straight while approaching the wall. This is easy to achieve in structured facilities like, for example, nuclear plants or square pipes, but difficult in environments where the soil is irregular.

After this point, we questioned whether an absolute and precise localization was actually necessary in these kinds of missions. Let us explain our point. Would it be useful to a human rescue team to know that the hypothetical position of injured people is $(x, y) = (27, 432)$ in the frame of a pre-existent map? Or would it be more valuable to know that the survivors are in a shelter at the end of the second gallery on the right which is approximately, for example, 200 m away? And if a human can manage with such inaccurate information, why would a robot be any less able?

With this in mind, an investigation into whether it was possible to get a robot to the destination point relying only on high-level topological-semantic information (such as 'go to second gallery and reach the survivors'), and a potentially inaccurate metric localization was begun. A system capable of interpreting this kind of information would have made it much easier to localize the

robots and to execute a navigation plan based upon that information. Indirectly, this approach imposed the need for developing a semantic feature recognition system capable of recognizing, for example, lateral galleries or diaphanous corridors.

O. Semantic Feature Recognition

To begin, an appearance-based recognition system was integrated based on the preliminary work presented in [Romeo and Montano, 2006]. It is capable of learning various semantic features, represented by classes ω_j, typically present in indoor maze-like environments, such as *corridor*, *cross*, *left hand*, *right hand*, *left turn*, *right turn*, *end of corridor* and *T-intersection* (see Fig. 26). The features are computed from the raw information provided by a two-dimensional LIDAR range-finder sensor. The system was trained from a set of these features artificially generated; that is, they do not belong to a specific environment.

The method first performs an input space transformation based on *Principal Component Analysis* (PCA), followed by a *Linear Discriminant Analysis* (LDA). Alternatively, a PCA can be substituted by a *Kernel Principal Component Analysis* (KPCA) for nonlinear discriminant analysis. This process captures the main characteristics of the features from the normalized raw data ($\mathbf{x}^{norm}$), reducing the information to the discriminant components vector $\mathbf{z}$ for feature identification:

$$\mathbf{z} = \mathcal{W}_{PCA/KPCA+LDA}(\mathbf{x}^{norm})$$

Once trained, a Bayesian classifier is applied for the feature recognition step. A posterior probability $p^o(\omega_j|\mathbf{z})$ is computed from the likelihood $p^o(\mathbf{z}|\omega_j)$ and the prior $p^o(\omega_j)$. A probability above the threshold will allow the feature to be accepted or rejected. The posterior probability is used in every cycle of recognition, as is the prior probability for the next cycle, which provides a robust behavior to the classifier, acting as a filter of sporadic misclassifications.

The method was evaluated with LIDAR data obtained from the Santa Marta mine (Figure 26), where 346 features over a path of $4,5km$ were involved. The results depended on the likeness of the features with those used as training data. In the more structured zone (Area A), $97,5\%$ of them were correctly recognized, against 57% in the non-structured one (Area B), where some misclassifications and unrecognized features appeared, especially in the portions in which the features differed noticeably from those learned. No tests were performed in Area

C for being excessively unstructured. Additionally, we performed tests in the Somport Tunnel, where all the 12 semantic features (lateral galleries) encountered by the robot while navigating were successfully recognized: neither false positives nor false negatives were found.

The satisfactory results opened up the possibility of using this technique for two complementary objectives: 1) a robot deployment plan definition based on those topological-semantic features explained above, and 2) an absolute topological localization while moving. The method prevented the robot from losing localization when only basic geometric features (points, lines) are used, and more importantly, continuous and accurate robot localization were no longer required. More details about these techniques and their performance can be found in [Tardioli et al., 2016a].

P. Dealing with Inaccurate Localization

This gave us the first construction block. The following step was how to solve the problem of navigating to a hypothetical gallery identified semantically and doing that without a map. More importantly, also, how to compute a relative localization among the robots when a coordinated movement is required, such as in the case of maintaining communication (which, in most cases means maintaining line-of-sight) when a robot enters, for example, a lateral gallery.

The solution came from the consideration that to relatively localize the robots, an accurate global localization is not necessarily required. The line-of-sight, for example, can be maintained without knowing the absolute localization, but instead, knowing the relative position of the two robots involved in the task. This reinforces the idea developed in the previous subsection, which is that continuous and precise robot global localization is not needed. Therefore, an 'inaccurate' shared map could be used.

First, the problem was limited by assuming that no pre-built map of the environment was available. This is a common assumption in intervention or search and rescue tasks. On the other hand, a map is typically necessary for globally localizing a group of robots. However, if each of the robots involved in the intervention task builds its own map, it would then be necessary to merge the maps into one, used to localize the robots.

This is a complex task for at least two reasons: the bandwidth necessary to share (the data to build) the map is high and not always available, and the merging of different maps is a difficult task, especially when, as in this case, loop closing is not possible.

Luckily, according to the chain configuration of the exploring team, the follower robots would always perceive portions of the environment the leader has already seen, given that a very similar path is covered slightly later. Thus, there is little sense in building a merged map with data that would be similar all along. Another option is to build a map with the leader robot data and use it to simultaneously localize the other robots. This solution has several advantages. The map is unique and all the robots share the same. Even if it is not precise in absolute terms, it will allow precise relative localization, at least in the featured portion of the environment. These are the zones where localization is most important (the zone around a lateral gallery, for example, whose T shape offers enough features for scan-matching). The amount of data that the robots need to exchange is also considerably reduced, given that only one map exists in the system.

Q. Assigning Position Goals Without a Map

Another issue to be resolved with this scheme was how to compute position goals for the leader robot. While the followers can use their localization on the shared map, the goals to make the leader move forward cannot be computed easily given that the map is being built as the robot advances. One can think that it can be easily done by facing the robot toward the tunnel and sending goals in a forward direction. However, this is not a good idea for two reasons. First, the robot may not have a zero yaw angle with respect to the tunnel axis at the beginning of the experiment. Second, the odometry drift would rapidly make this method useless. Additionally, the tunnel does not have to be exactly straight. In both cases, assuming that the tunnel axis matches the x axis of the robot frame would be a mistake.

With this in mind, it was determined that the robot should simply move parallel to the walls of the tunnel almost all the time. This means that the navigation goal should simply be a point at a fixed distance from the wall, no matter the orientation of the map or the robot. By detecting the walls with LIDAR readings, a goal can be sent that would allow the robot to follow the longitudinal axis of the tunnel. This requirement was met by applying the Hough Transform [Duda and Hart, 1972] to find the straight lines corresponding to the right and left walls (see Fig. 28). The relative robot orientation and its distance to both walls are then obtained from them (see [Tardioli and Villarroel, 2014] and references therein).

Once the orientation of the walls with respect to the robot frame is detected, it is trivial to compute a goal position in the center of the tunnel in the same frame, and thus achieve the required movement.

R. Centralizing the Computation

From a conceptual viewpoint the scheme proposed, in which the leader moving along the longitudinal axis is the only building the map, can be implemented using different schemes. One is that it can share the map every so often and the followers can run a localization algorithm (for example, a particle filter) on it, updating the map when new data is received.

However, this approach has several drawbacks: 1) it is necessary a network capable of supporting the burst of traffic that takes place when the map is shared, 2) the onboard computer of the leader must be powerful enough to build a map, and 3) assume that the followers computers are capable of running a localization algorithm, 4) some of the information is local to the robots so if one of them breaks, gets lost, or cannot get back to the base station, all the information collected will be lost.

Another possible approach comes from the consideration that real-world intervention or search and rescue missions have usually a centralized nature, i.e., a central crisis unit manages several rescue teams that can be considered independent entities. The information is centralized in a single place to avoid duplication and to allow for the efficient management of resources. In this approach, the robots can only carry the most basic sensors/actuators, together with the basic functionalities to provide autonomy, and to send all the information (e.g., laser readings, camera images, etc.) to the base station to be collected and processed in real-time. This proposal has two main advantages: 1) the cost of the robots is reduced without the need for high computational power that is centralized at the base station, where limitations in terms of size or power are not as stringent and failures can be recovered more rapidly, and 2) all the information is instantly available at the base station.

These two factors convinced us to explore this option. Thus, most processing would be executed at the base station; the only tasks for the robots would be to send sensor readings to the base station, where the map will be built, and perform local reactive navigation to avoid obstacles. At the same time, one localization algorithm per follower would be run over the (growing) map.

Naturally, this choice has also drawbacks. With this scheme, on the one hand, all the weight falls on the network; if it fails, everything fails. On the other hand, this choice indirectly implies that maintaining communication at all times is a primary requisite that cannot be overlooked. Also, the network is required to guarantee a real-time behavior –that is, bounded end-to-end

---

Algorithm 1 Deployment Planning.

1: $state \leftarrow BEGIN$
2: $plan[] \leftarrow read\_plan()$
3: $last\_mobile \leftarrow n - 1; id \leftarrow 0$
4: $set\_rope(R_{n-1}, R_0)$
5: **for** i in $1..n - 2$ **do**
6:     $set\_rope(R_i + 1, R_i)$
7: **while** true **do**
8:     **if** $state == BEGIN$ **then**
9:         **if** $found\_feature(plan[id])$ **then**
10:           **if** $last\_feature(plan[id])$ **then**
11:             **break**
12:           $state \leftarrow GROUP\_ROBOTS$
13:         **else**
14:           $move(R1)$
15:           $find\_feature(plan[id])$
16:     **else if** $state = GROUP\_ROBOTS$ **then**

17:         **if** $robots\_grouped()$ **then**
18:           $compute\_feat\_pos(\&feat\_x, \&feat\_y)$
19:           $send\_xy\_goal(R_1, feat\_x, feat\_y)$
20:           $state \leftarrow ENTER\_FEATURE$
21:         **else**
22:           **for** i in $1..last\_mobile - 1$ **do**
23:             $send\_rel\_goal(R_i, R_{i+1})$
24:     **else if** $state = ENTER\_FEATURE$ **then**
25:         **if** $robots\_grouped()$ **then**
26:           $id \leftarrow id + 1$
27:           $fix(last\_mobile)$
28:           $state \leftarrow BEGIN$
29:         **else**
30:           **for** i in $1..last\_mobile - 1$ **do**
31:             $send\_rel\_goal(R_i, R_{i+1})$
32:     **sleep_until_ms**(500)

---

delay and reduced jitter– given that the LIDAR and position data provided to the localization and SLAM algorithms must be periodic and respect precise timings for offering the expected performances.

In this case, a system was setup where the global map building was taken care of by the GMapping algorithm [Grisetti et al., 2007] and, specifically, by the respective ROS node. The localization was performed by an AMCL algorithm [Fox, 2003], as well as through its ROS node. The second was modified to allow regular online updating of the map.

S. Testing the Intervention Team

All the construction blocks necessary to set up a complete intervention team were now available. The system was organized into several interconnected modules, as shown in Figure 29, which shows the conceptual architecture. The system was composed of a set of n mobile robots, one of which assumed the role of leader and $n - 1$ the role of followers. The leader and follower robots had very basic characteristics, such that each could be low-cost devices easily replaced or disposed easily as just expounded.

The robots were built on top of a basic platform (Pioneer P3AT) that provides a 2D LIDAR and odometry readings, and were capable of reacting to velocity commands. The robots were equipped with three onboard modules: the Reactive Navigation Module (RNM) which manages collision avoidance, the Navigation Control Module (NCM) which computes trajectories, and

the COmmunication Module (COM) which connects all the nodes of the network using the last version of RT-WMP protocol [Tardioli et al., 2015].

The robots mostly executed commands received from the base station through the COM. These commands were translated by the NCM into one of the two predefined navigation strategies. The first made each robot capable of reaching a goal provided by the base station and given on the map frame (which was built by the base station itself with the data provided by the leader as anticipated). The second allowed each robot to navigate forward, following a lateral wall using the mentioned navigation algorithm based on the Hough Transform. In both cases the RNM translated the goals into velocity commands compatible with the avoidance of potential obstacles. The leader also had a camera that, when active, sent images to the base station through the COM.

The base station was responsible for carrying out most of the heavy computational tasks. It was implemented by taking advantage of several modules that interpreted the scenario through the recognition of semantic features, localizing the robots, building the map, planning the goals, and managing the communication.

The SLAM module received laser and localization data from the mobile robots through the COM and, simultaneously, localized the leader and built the map of the environment in real-time using the cited GMapping algorithm. This map is provided with a non-constant rate to the $n - 1$ MCL (Monte Carlo Localization) modules that, together with laser and odometry data from the followers, localize the latter. This enables all the robots to be localized over the same map built on-the-fly.

The Feature Detector module (FD) also processed the laser data from the leader and detected the different semantic characteristics of the environment (left or right gallery, end of corridor, etc.), while interpreting each in order to topologically localize the robots. All this information was provided, together with the localization of the robots from the SLAM and MCL modules to a State Machine module (SM) that was responsible for making all decisions. Finally, the JOY module generated the commands that allow the teleoperation of the robot when manual control was requested.

From a communication viewpoint, a total of three flows were established between each robot and the base station: *LIDAR* (with a source period of 200 ms), *Odometry* (100 ms) and *Camera* (260 ms). Moreover, an additional *Goal* flow was established between the base station and each robot with a period of 100 ms.

The algorithm used for deployment is shown in Algorithm 1: the state machine starts with the state $BEGIN$, in which the leader robot will move looking for the $n - th$ occurrence of a feature (lines 15, 16). When it finds what it is looking for, it checks if this is the last part of the plan, in which case, it exits (line 9, 10). Otherwise, all the robots are grouped behind the leader, maintaining the chain formation (lines 20, 21). It is worth noting that one or more robots can be stopped during this phase, according to the scheme described in Section III-F. When the grouping is complete, the position within the feature is computed and the leader robot is sent inside (line 17-19). The formation is then restored (lines 30, 31). When all followers are at the final position, the last mobile node is fixed, and the algorithm restarts after moving the pointer id to the next feature (lines 25-27).

T. Refining the Framework

The framework was extensively tested in simulation and in real experiments. Figure 27 shows a simulation in which a five-robot team had the task of reaching a specific location within the Santa Marta mine. The corresponding high-level plan was 'take third gallery on the left, then second on the right, and the first on the left, and stop at the cross' specified through a YAML configuration file. In this experiment, the configuration was exactly the same as that used in the real-world experiments, except that all nodes were run in the same machine and the communication was local. The link qualities were simulated using the Manhattan distance between each pair of robots. The map was created on-the-fly by the leader robot and distributed to the followers. It is possible to notice that all followers were acting as relays and three of them (R4, R3 and R2) stopped autonomously at the corresponding corners to maintain the line-of-sight.

Although it took hard work to get the system working as expected, due to the number of parameters to be adjusted, the simulations were trivial compared to performing the experiments in the real-world. Most of the issues came from navigation, especially when the movement of the robots required some level of precision or when the movement of two (or more) had to be synchronized.

The first problem was how to enter a feature (a lateral gallery, for example) without knowing the localization of the gallery on the map in advance. Remember that the map was built on-the-fly, thus the coordinates of the features on the map itself were unknown. This problem was solved by analyzing the LIDAR readings, as shown in Figure 30(a). When the robot got close to a lateral gallery, the latter was detected by the semantic recognizer, and the shape of the LIDAR

readings were similar to those shown in Figure 30(a). It was then easy to compute the maximum of the function (M_1 and M_2) and the points G_1 and G_2 at a certain distance from the walls. Considering the orientation of the robots with respect to the walls, the point G_1 was chosen and used as the goal for the leader robot on its own reference frame.

Another problem faced in the field experiments was due to the method used to coordinate the movement of the robots, and specifically, the one that allowed the leader to pull the followers or the followers each other. As discussed in Section III-F, when the signal quality decreases, an SDS is created between the leader and the immediate follower (or between followers) and the first pulls the second. This is sufficient when the environment is completely straight but the method will be hindered when it is not. Consider Figure 30(b). When the signal quality between $R1$ and $R2$ falls below the threshold a force, represented by the red arrow (a), will pull R_2, resulting in it moving closer to the wall and eventually stopping before crashing. To avoid this behavior altogether, the robot R_1 will leave breadcrumbs (position of the goals chosen on the path followed by the leader) that the robot being pulled must follow. The corresponding force, represented by the blue line (b), has the same intensity as the original force but in a different direction as shown in the figure. In this way, the follower can traverse the corridor safely while maintaining the link with the robot directly ahead.

U. The Final Experiment

After solving the problems described, a complete intervention experiment in the Somport Tunnel was performed, where the goal was to reach '*the end of the second gallery on the left*' and then teleoperate the leader to look for the status of a set of wrecked cars parked in the lateral gallery itself. Figure 31 shows the scheme of the entire experiment conducted on a 500 m stretch of the tunnel (left), as well as the data logged in the last part of the test, including the path followed by the leader robot and the position of the first follower acting as relay (right). As planned, once the robot had reached the end of the lateral gallery, it was teleoperated toward the wrecked car from the base station by an operator that was receiving LIDAR and video feedback (see Fig. 32).

In this field experiment we temporally went back to a lazy behavior to reduce the logistic effort: taking into account that we did not exceed the distance of the near propagation sector; the difference is only in terms of time. Figure 33 shows the real RSSI among the robots in the first part of one experiment, in which the robots were deployed until reaching the gallery, and the

distance with respect to the base station. As can be seen, the RSSI between R_1 and R_2 begins to decrease as soon as R_1 starts to move. When the RSSI between these two robots falls below the fixed upper threshold of 60%, R_2 starts to move to maintain a good link quality. The RSSI between R_1 and R_2 stabilizes around that value, while that between R_2 and R_3 starts to decrease due to the increasing distance. Later, the RSSI between the latter two robots also falls below the threshold and R_3 starts moving toward R_2, stabilizing the signal. However, later, the RSSI again overcomes the upper threshold and R_3 temporarily stops. On the other hand, the RSSI between R_3 and the base station decreases gradually until it falls below the lower threshold (fixed at 35%) and robot R_3 is immediately stopped by the SM. Meanwhile, R_2 continues to move and reaches R_1. At that moment, the node R_2 is fixed at the corner and R_1 restarts its movement inside the lateral gallery.

The experiments, whose details can be consulted in [Tardioli et al., 2016a], were successful and confirmed that most of the obstacles found were overcome and that the framework was finally ready for intervention. Figure 34(a) presents the message inter-arrival time for the different flows. The figure shows that the periods are mostly respected with clear Gaussian distributions around the expected values. This confirmed that both the SLAM and the AMCL modules were being fed with adequate frequency and timing.

The perception-actuation loop period is presented in Figure 34(b). This has been measured considering the delay between the sending of a control command and the reception of a laser reading from the leader robot. The graph shows a Poisson distribution with a mean of 262 ms that is close to the laser period, as expected, guaranteeing a satisfactory teleoperation.

IV. CONCLUDING REMARKS: LESSONS LEARNED

In this paper our journey of over a decade of research on robotics in tunnels was presented. These environments contain a long list of peculiarities and issues that often make well-established methods of navigation, localization or communication fail. Our research began back in 2006, when a simple video communication link was attempted between the two ends of a short tunnel, and this still continues today in more complex tasks. In the last few years, a broad range of problems and challenges were faced, which were learned from or overcome, by focusing on the issues involved in localization, navigation and communication. This paper aimed to offer an overview of the problems and challenges that those who decide to investigate in intervention tasks in confined environments will probably encounter. Another aim of this paper is to openly

share our experiences with the robotics community, backed up by more than 20 publications and nine national and international research projects, as well as eight PhD theses. It is hoped that this will be especially useful to those researchers who struggle with real-world robotics problems. Technical or implementation details can be found in the papers cited within.

We hope we transmitted what should (and what should not) be packed in a robot's luggage for a successful trip into a tunnel, and the other aspects that must be taken into account:

- The environment is nothing like a lab nor a parking lot: it is usually dark, the floor is irregular, slippery, and there is no GPS coverage.

- Localization is complicated: the odometry is more unreliable than usual due to the irregularities of the floor, the walls are smooth and thus there are less visual or physical features such that scan-matching, LIDAR-based SLAM or Visual SLAM algorithms may not perform adequately. Also, in intervention tasks, a pre-existent map may not be available and cannot be easily built with the required precision. This means that unless a precise localization is mandatory, the system should be able to deal with uncertainty or be based on higher-level (e.g., semantic) features, which can be sporadically found in the environment.

- Navigation must be often carried out without reliable localization. This implies that, again, it should rely on physical characteristics of the environment and on topological and semantic information. Additionally, the robots should be autonomous enough to deal with unattended local navigation (i.e. must be capable of reactive behavior and obstacle avoidance) both during automated navigation and teleoperation to guarantee safety (for itself and the people that may be in the environment), even in case of network failure.

- Communication must take into account the peculiar propagation pattern within this kind of environment, which includes slow and fast fadings and the need for covering distances of several kilometers. This means that it is usually necessary to rely on multi-hop communication and that, in the case of using a backbone network, careful planning is needed to determine where to place the repeaters.

  In addition, the network must be timing-aware to guarantee an adequate control task period and reduced jitter, which is paramount when the network is part of the control-loop or when there is a man in the middle.

- The robot team can use any of the distributed, decentralized and centralized paradigms, but the choice has implications both in terms of system architecture and network support, especially considering the bandwidth necessary for the implementation. Here, the centralized

solution is advocated, in which the network bears most of the weight but has the advantage of requiring less clever robots that are cheaper and longer lasting. This centralized solution is compatible with some distributed functionalities among the robots, such as obstacle avoidance, to make the system safer, mainly in the case where a robot is lost or disconnected from the base station.

V. FUTURE WORK

Naturally, this paper does not cover all aspects of such a difficult field, nor is our research complete. Work remains to be done in order to keep advancing and making intervention teams better. Our present goal is this: making robots more and more autonomous and allowing them to deal with even more challenging environments. When the floor is full of mud, as in the environment shown in Figure 35(a), for example, navigation and localization are especially complex. Besides the difficulties of traversing the terrain, which requires a platform capable of doing so (see Fig. 35(b)), the bigger problem consists in how to use the data provided by the sensors for a proper localization. The presence of a soil so extraordinarily irregular, slippery, and dynamic (the robot tends to plunge into the mud), makes it impossible to identify a clear floor frame to which transform the sensor readings in order to use standard localization or navigation techniques. Similarly, processing LIDAR information in presence of puddles and deep potholes to generate a feasible and safe navigation plan is remarkably difficult, as confirmed by the preliminary experiments we carried out in [Tardioli et al., 2018b]. Another finding that we believe deserves deeper research is the use of the RF signal itself as an ad-hoc landmark to help in localization in these environments. While some promising result on discrete robot localization are presented in [Seco et al., 2018], we are now refining the method for continuous localization. Moreover, we are experimenting with UAVs since we believe they can very useful in this type of interventions. Specifically, our interest focuses in hybrid teams capable of taking advantage of the much higher degree of mobility of these devices opposed to their limited battery life compared to ground robots. In our vision, ground robots can transport UAVs on board, to be used for punctual exploration of otherwise inaccessible areas when needed.

REFERENCES

[Alonso-Mora et al., 2018] Alonso-Mora, J., Montijano, E., Nägeli, T., Hilliges, O., Schwager, M., and Rus, D. (2018). Distributed multi-robot formation control in dynamic environments. *Autonomous Robots*.

[Bakambu, 2006] Bakambu, J. N. (2006). Integrated autonomous system for exploration and navigation in underground mines. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2308–2313.

[Brooker et al., 2007] Brooker, G., Hennessey, R., Lobsey, C., Bishop, M., and Widzyk-Capehart, E. (2007). Seeing through dust and water vapor: Millimeter wave radar sensors for mining applications. *Journal of Field Robotics*, 24(7):527–557.

[Chung, 2018] Chung, T. (2018). DARPA subterranean (SubT) challenge. *DARPA Challenge TTO, https://www.darpa.mil/news-events/2017-12-21*.

[Delogne, 1991] Delogne, P. (1991). EM propagation in tunnels. *IEEE Trans. on Antennas and Propagation*, 39(3):401–406.

[Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271.

[Dixon and Frew, 2009] Dixon, C. and Frew, E. W. (2009). Maintaining Optimal Communication Chains in Robotic Sensors Networks Using Mobility Control. *Mobile Networks and Applications*, 14(3):281–291.

[Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15.

[Dudley et al., 2007] Dudley, D., Lienard, M., Mahmoud, S., and Degauque, P. (2007). Wireless propagation in tunnels. *Antennas and Propagation Magazine, IEEE*, 49(2):11–26.

[Emslie et al., 1975] Emslie, A., Lagace, R., and Strong, P. (1975). Theory of the propagation of UHF radio waves in coal mine tunnels. *IEEE Trans. Antennas Propag.*, 23(2):192–205.

[Fox, 2003] Fox, D. (2003). Adapting the sample size in particle filters through kld-sampling. *The International Journal of Robotics Research*, 22(12):985–1003.

[Fox et al., 1997] Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1):23–33.

[Frese, 2010] Frese, U. (2010). Interview: Is slam solved? *Robotics and Autonomous Systems*, 24:255–257.

[Grisetti et al., 2007] Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46.

[Hsieh et al., 2008] Hsieh, M. A., Cowley, A., Kumar, V., and Taylor, C. J. (2008). Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*, 25(1-2):111–131.

[Jacobson et al., 2018] Jacobson, A., Zeng, F., Smith, D., Boswell, N., Peynot, T., and Milford, M. (2018). Semi-supervised slam: Leveraging low-cost sensors on underground autonomous vehicles for position tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3970–3976.

[Kapp, 2002] Kapp, S. (2002). 802.11: Leaving the wire behind. *IEEE Internet Computing*, 6:82–85.

[Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM Int. Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan.

[Lazaro and Castellanos, 2010] Lazaro, M. T. and Castellanos, J. A. (2010). Localization of probabilistic robot formations in SLAM. In *the 2010 IEEE International Conference on Robotics and Automation*, pages 3179–3184.

[Leingartner et al., 2016] Leingartner, M., Maurer, J., Ferrein, A., and Steinbauer, G. (2016). Evaluation of sensors and mapping approaches for disasters in tunnels. *Journal of Field Robotics*, 8(33):1037–1057.

[Lienard and Degauque, 1998] Lienard, M. and Degauque, P. (1998). Propagation in wide tunnels at 2 GHz: a statistical analysis. *Vehicular Technology, IEEE Transactions on*, 47(4):1322–1328.

[Liu and Oh, 2012] Liu, Q.-L. and Oh, D.-H. (2012). Performance evaluation of multi-hop communication based on a mobile multi-robot system in a subterranean laneway. *Journal of Information Processing Systems*, 8(3):471–482.

[Losch et al., 2018] Losch, R., Grehl, S., Donner, M., Buhl1, C., and Jung, B. (2018). Design of an autonomous robot for mapping, navigation, and manipulation in underground mines. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1406–1412.

[Loupos et al., 2018] Loupos, K., Doulamis, A. D., Stentoumis, C., Protopapadakis, E., Makantasis, K., Doulamis, N. D., Amditis, A., Chrobocinski, P., Victores, J., Montero, R., Menendez, E., Balaguer, C., Lopez, R., Cantero, M., Navarro, R., Roncaglia, A., Belsito, L., Camarinopoulos, S., Komodakis, N., and Singh, P. (2018). Autonomous robotic system for tunnel structural inspection and assessment. *International Journal of Intelligent Robotics and Applications*, 2(1):43–66.

[Martínez and Harbour, 2005] Martínez, J. M. and Harbour, M. G. (2005). RT-EP: A Fixed-Priority Real Time Communication Protocol over Standard Ethernet. In *the 10th International Conference on Reliable Software Technologies, Ada-Europe*, pages 180–195.

[Mascarich et al., 2018] Mascarich, F., Khattak, S., Papachristos, C., and Alexis, K. (2018). A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels. In *2018 IEEE Aerospace Conference*, pages 1–7.

[Minguez, 2005] Minguez, J. (2005). The obstacle-restriction method (orm) for robot obstacle avoidance in difficult environments. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, page 37063712.

[Minguez and Montano, 2004] Minguez, J. and Montano, L. (2004). Nearness diagram navigation (nd): Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation 20(1)*, pages 45–59.

[Mur-Artal and Tardós, 2017] Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robotics*, 33(5):1255–1262.

[Murphy et al., 2008] Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M. (2008). *Search and Rescue Robotics*, pages 1151–1173. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Nagatani et al., 2013] Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., and Kawatsuma, S. (2013). Emergency response to the nuclear accident at the Fukushima Daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63.

[Ng and Liew, 2007] Ng, P. C. and Liew, S. C. (2007). Throughput analysis of IEEE802.11 multi-hop ad hoc networks. *IEEE/ACM Transactions on networking*, 15(2):309–322.

[Nsasi and Vladimir, 2007] Nsasi, B. J. and Vladimir, P. (2007). Autonomous system for navigation and surveying in underground mines. *Journal of Field Robotics*, 24(10):829–847.

[Pasteau et al., 2016] Pasteau, F., Narayanan, V. K., Babel, M., and Chaumette, F. (2016). A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair. *Robotics and Autonomous Systems*, 75:28 – 40. Assistance and Service Robotics in a Human Environment.

[Prim, 1957] Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401.

[Rizzo et al., 2018] Rizzo, C., Cavestany, P., Chataigner, F., Soler, M., Moreno, G., Serrano, D., Lera, F., and Villarroel, J. L. (2018). Wireless propagation characterization of underground sewers towards autonomous inspection with drones. In *ROBOT 2017: Third Iberian Robotics Conference, VOL 2*, volume 694 of *Advances in Intelligent Systems and Computing*, pages 849–860. Springer International Publishing AG.

[Rizzo et al., 2014a] Rizzo, C., Kumar, V., Lera, F., and Villarroel, J. L. (2014a). RF odometry for localization in pipes based on periodic signal fadings. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4577–4583.

[Rizzo et al., 2013a] Rizzo, C., Lera, F., and Villarroel, J. L. (2013a). Transversal fading analysis in straight tunnels at 2.4 GHz. In *2013 13th Int. Conf. on ITS Telecommunications (ITST)*, pages 313–318.

[Rizzo et al., 2013b] Rizzo, C., Lera, F., and Villarroel, J. L. (2013b). Uhf and shf fading analysis using wavelets in tunnel environments. In *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, pages 1–6.

[Rizzo et al., 2014b] Rizzo, C., Lera, F., and Villarroel, J. L. (2014b). A methodology for localization in tunnels based on periodic RF signal fadings. In *2014 IEEE Military Communications Conference*, pages 317–324.

[Rizzo et al., 2016] Rizzo, C., Sicignano, D., Riazuelo, L., Tardioli, D., Lera, F., Villarroel, J. L., and Montano, L. (2016). Guaranteeing communication for robotic intervention in long tunnel scenarios. In *Robot 2015: Second Iberian Robotics Conference*, pages 691–703. Springer International Publishing.

[Rizzo et al., 2013c] Rizzo, C., Tardioli, D., Sicignano, D., Riazuelo, L., Villarroel, J. L., and Montano, L. (2013c). Signal-based deployment planning for robot teams in tunnel-like fading environments. *The International Journal of Robotics Research*, 32(12):1381–1397.

[Romeo and Montano, 2006] Romeo, A. and Montano, L. (2006). Environment understanding: Robust feature extraction from range sensor data. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3337–3343.

[Seco et al., 2018] Seco, T., Rizzo, C., Espelosín, J., and Villarroel, J. L. (2018). Discrete robot localization in tunnels. In Ollero, A., Sanfeliu, A., Montano, L., Lau, N., and Cardeira, C., editors, *ROBOT 2017: Third Iberian Robotics Conference*, pages 823–834, Cham. Springer International Publishing.

[Serafin and Grisetti, 2015] Serafin, J. and Grisetti, G. (2015). NICP: Dense normal based point cloud registration. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 742–749.

[Sicignano et al., 2013] Sicignano, D., Tardioli, D., Cabrero, S., and Villarroel, J. L. (2013). Real-time wireless multi-hop protocol in underground voice communication. *Ad Hoc Netw.*, 11(4):1484–1496.

[Sicignano et al., 2010] Sicignano, D., Tardioli, D., and Villarroel, J. L. (2010). *QoS over Real-Time Wireless Multi-hop Protocol*, pages 110–128. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Tardioli et al., 2010] Tardioli, D., Mosteo, A., Riazuelo, L., Villarroel, J., and Montano, L. (2010). Enforcing network connectivity in robot team missions. *The International Journal of Robotics Research*, 29(4):460–480.

[Tardioli et al., 2018a] Tardioli, D., Parasuraman, R., and Ögren, P. (2018a). Pound: A multi-master ROS node for reducing delay and jitter in wireless multi-robot networks. *Robotics and Autonomous Systems*, 111:73 – 87.

[Tardioli et al., 2018b] Tardioli, D., Riazuelo, L., Seco, T., Espelosín, J., Lalana, J., Villarroel, J. L., and Montano, L. (2018b). A robotized dumper for debris removal in tunnels under construction. In *Advances in Intelligent systems and computing 639, Springer Int. Publishing, Third Iberian Robotics Conference*, pages 126–139. Springer, Cham.

[Tardioli et al., 2016a] Tardioli, D., Sicignano, D., Riazuelo, L., Romeo, A., Villarroel, J. L., and Montano, L. (2016a). Robot teams for intervention in confined and structured environments. *Journal of Field Robotics*, 33(6):765–801.

[Tardioli et al., 2016b] Tardioli, D., Sicignano, D., Riazuelo, L., Romeo, A., Villarroel, J. L., and Montano, L. (2016b). Robot teams for intervention in confined and structured environments. *Journal of Field Robotics*, 33(6):765–801.

[Tardioli et al., 2012] Tardioli, D., Sicignano, D., Riazuelo, L., Villarroel, J., and Montano, L. (2012). Robot teams for exploration in underground environments. In *Workshop ROBOT11: Robótica Experimental, Sevilla (Spain)*, pages 205–212.

[Tardioli et al., 2015] Tardioli, D., Sicignano, D., and Villarroel, J. L. (2015). A wireless multi-hop protocol for real-time applications. *Computer Communications*, 55:4–21.

[Tardioli and Villarroel, 2007] Tardioli, D. and Villarroel, J. L. (2007). Real time communications over 802.11: RT-WMP. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1–11.

[Tardioli and Villarroel, 2012] Tardioli, D. and Villarroel, J. L. (2012). Routing wireless real-time traffic using minimum spanning trees. In *2012 International Conference on Computing, Networking and Communications (ICNC)*, pages 170–176.

[Tardioli and Villarroel, 2014] Tardioli, D. and Villarroel, J. L. (2014). Odometry-less localization in tunnel-like environments. In *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 65–72.

[Urcola et al., 2008] Urcola, P., Riazuelo, L., Lazaro, M., and Montano, L. (2008). Cooperative navigation using environment compliant robot formations. In *Int. conf. on Intelligent Robots and Systems, 2008. IEEE/RSJ*, pages 2789–2794.

[Yang et al., 2017] Yang, S., Song, Y., Kaess, M., and Scherer, S. (2017). Pop-up SLAM: semantic monocular plane SLAM for low-texture environments. *CoRR*, abs/1703.07334.

[Zhou et al., 2015] Zhou, C., Plass, T., Jacksha, R., and Waynert, J. A. (2015). RF propagation in mines and tunnels: Extensive measurements for vertically, horizontally, and cross-polarized signals in mines and tunnels. *IEEE Antennas and Propagation Magazine*, 57(4):88–102.

[Zlot and Bosse, 2014] Zlot, R. and Bosse, M. (2014). Efficient Large-Scale Three-Dimensional Mobile Mapping for Underground Mines. *Journal of Field Robotics*, 31:758–779.

[Özaslan et al., 2017] Özaslan, T., Loianno, G., Keller, J., Taylor, C. J., Kumar, V., Wozencraft, J. M., and Hood, T. (2017). Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs. *IEEE Robotics and Automation Letters*, 2(3):1740–1747.

photos/orbslam_durango_1.png

photos/orb_slam_trajectory.png

photos/orb_slam_points.png

Fig. 4: Two ActivMedia Pioneer P3AT robots autonomously avoiding a car inside the Manzanera tunnel, 600 m long, while establishing a video WIFI-based link with the base station located at the entrance of the tunnel. The robots were equipped with a SICK LMS 200 Laser Range-Finder, a Canon VC-C4 camera and a CISCO AIRONET 350 wireless device. The onboard computer was a VersaLogic VSBC-6 PC104+ (Pentium III 850 MHz) running Windows XP OS.
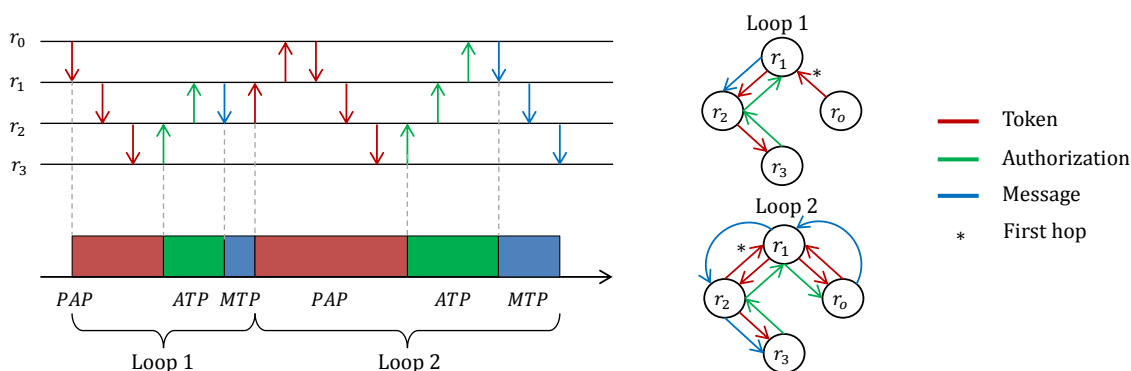
Fig. 5: An example of RT-WMP operations, phases and flows management. In (a) and (b) Priority Arbitration Phase (PAP), Authorization Transmission Phase (ATP) and Message Transmission Phase (MTP) are shown on two consecutive loops. The * symbol indicates the loop start. During the PAP (in red), nodes reach a consensus about which holds the Most Priority Message in the network. Subsequently, in the ATP (in green), an authorization to transmit is sent to the node that holds the MPM. Finally, in the MTP (in blue), this node sends the message to the destination. In (c), an example of how the protocol manages the periodic flows involved in the experiment.

Fig. 6: Interior of the Somport Tunnel. A small shelter can be seen on the right wall within the red square. When the shelters were found, the localization uncertainty along the longitudinal axis was sometimes reduced.
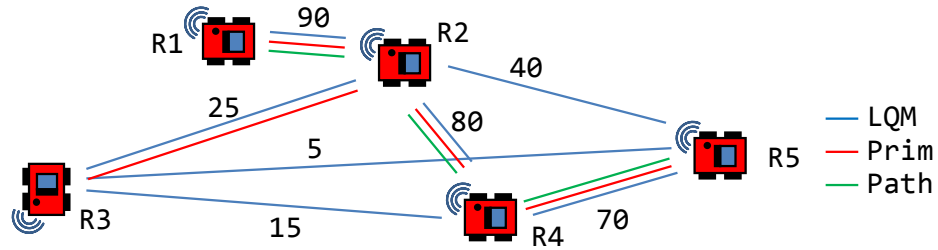
Fig. 7: The LQM contains the link quality between each pair of nodes (blue line). The Minimum Spanning Tree (red line) is computed over the graph, represented by the LQM. Then, the Dijkstra algorithm was used to compute the best path over the tree (green line).
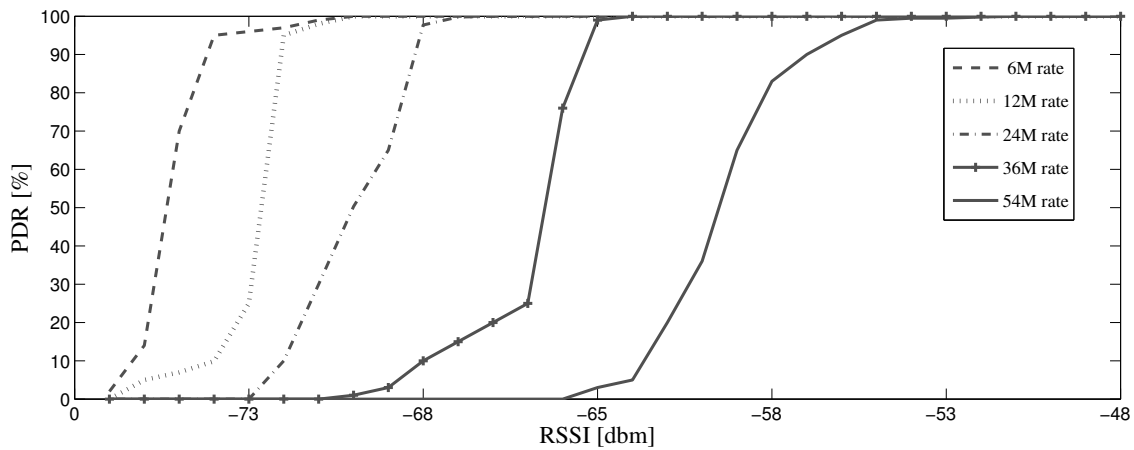
Fig. 8: Relationship between Packet Delivery Ratio (PDR) and RSSI obtained through real-world experiments in which a receiver measured the RSSI of the frames sent by a transmitter node. The RSSI was gradually reduced at the receiver side using RF attenuators. It is easy to identify, for each data rate, the RSSI that guarantees a PDR close to 100%.
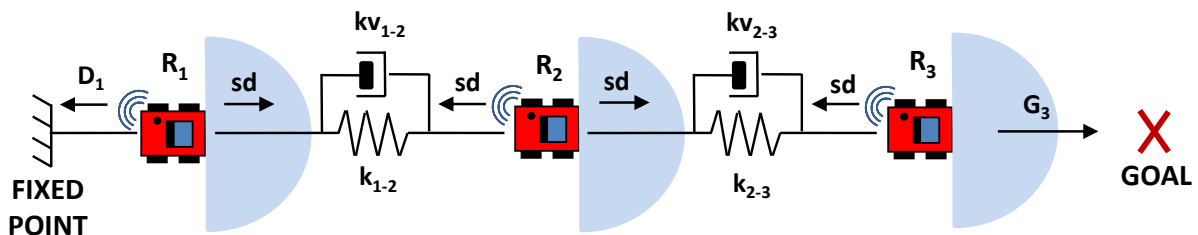
Fig. 9: Spring-damper system (SDS) for controlling the robot motion from the signal strength being *sd* the virtual forces connecting the robots, K_{i-j} and Kv_{i-j} the springs and the dampers that connect robot i and j, respectively. Force G_3 represent the pulling force of the goal while D_1 the force between the last robot and the fixed point.
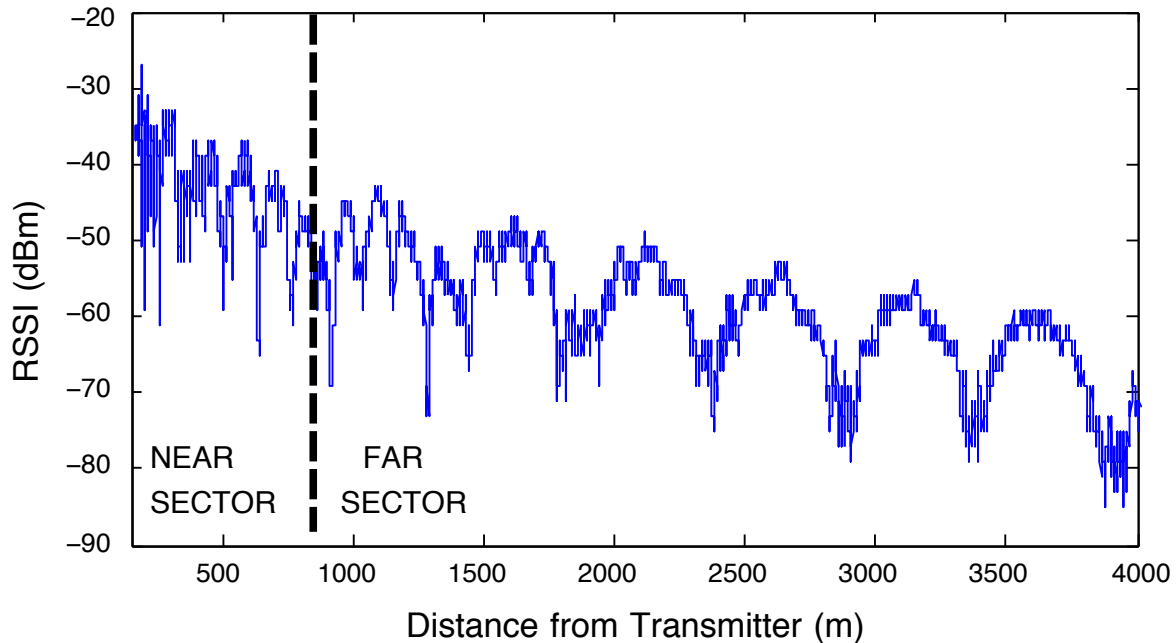
Fig. 10: Received signal strength (RSSI) at 2.4 GHz along a 4 km path inside the Somport Tunnel. Notice the long reach, for an emitter that was a standard off-the-shelf USB WIFI card+antenna combo suitable for no more than 100 m in an office environment. Note also the existence of deep fadings –512 m period– with RSSI in *valleys* too low for maintaining an adequate communications link.

Fig. 11: A robotized car and the Robucar platform used in the measuring campaign for verifying the reproducibility of the power scheme distribution in fading scenarios. The all-terrain vehicle was equipped, besides with twelve communication nodes (TP-LINK WN7200ND), with two SICK DSF60 odometers and a SICK LMS 200 LIDAR. The Robucar platform was equipped with two frontal (SICK LMS 200) and two rear (SICK LMS 291/200) LIDAR sensors. It also incorporates three wireless Alpha Network AWUS036NH adapters.
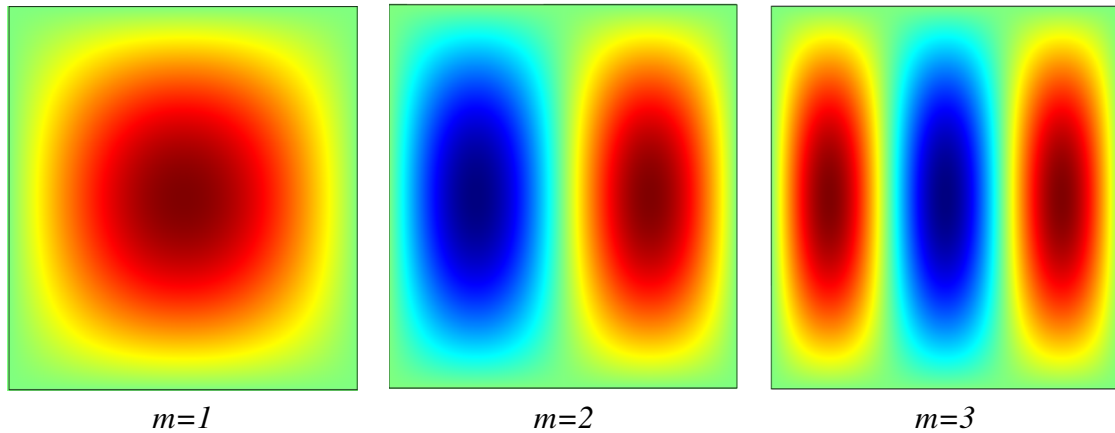
Fig. 12: Electric field distribution for the first three vertical modes across a rectangular tunnel section. The index m accounts for the number of power maxima across the tunnel width. Red-yellow shades depict positive values, while the blue shades show negative values. Green represents zero field.
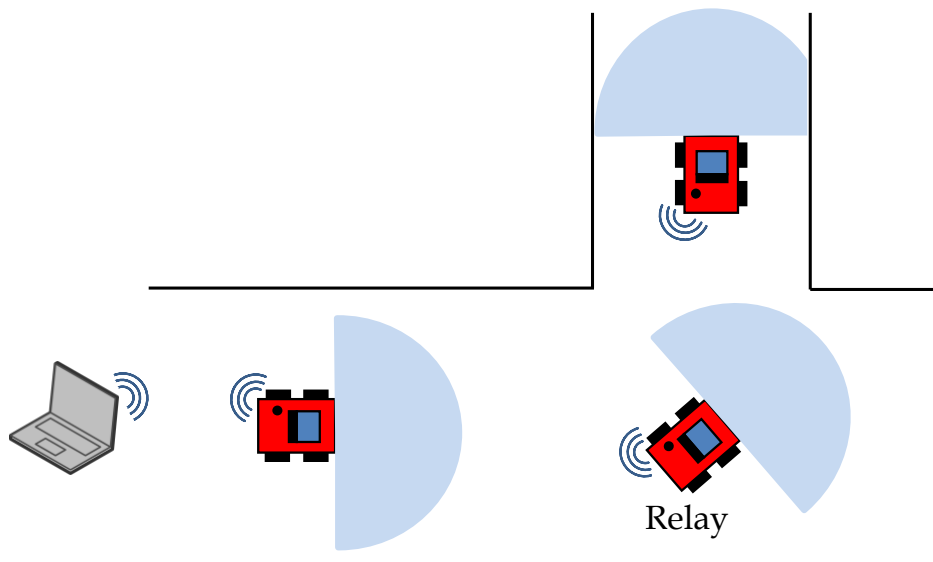
Fig. 13: The radio signal suffers from strong attenuation when the line-of-sight is lost. In order to guarantee the communication, a robot is positioned in the corner to maintain the line-of-sight with both the robot still in the corridor and the robot entering the lateral gallery, and is used as a relay.

(a)　　　　　　　　　　　　(b)

Fig. 14: Reflective panels used for relative localization (a). The leader robot moving toward the shelter in the lateral gallery, and the follower standing at the corner acting as a relay for the base station located 300 m behind this robot in the tunnel (b).

Fig. 15: The robot reaching the shelter at the end of a lateral gallery (120 meters long). The robot incorporates multimedia devices (microphone, speakers and an RGB camera).
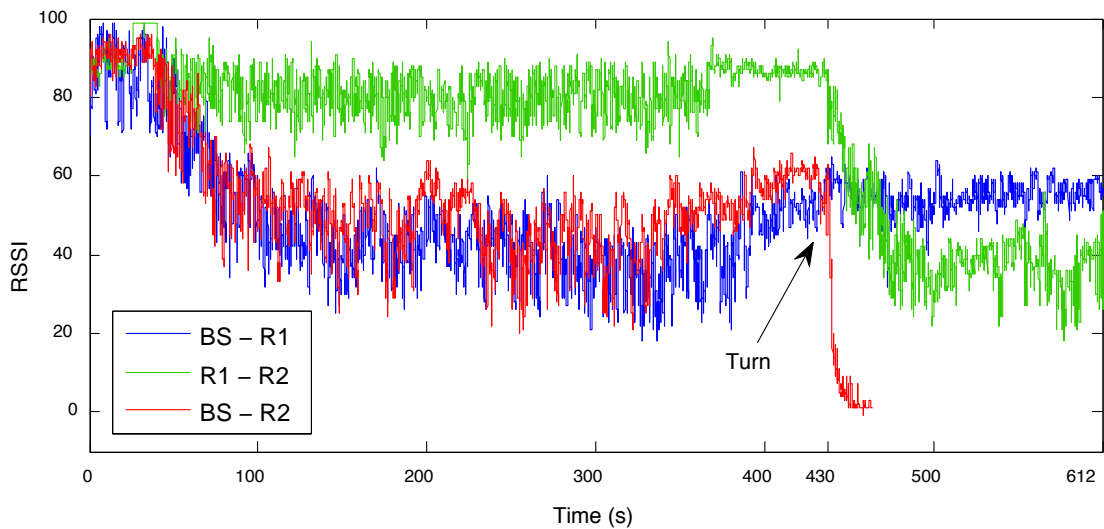
Fig. 16: RSSI registered by each pair of nodes during the experiment. The signal between the BS and the mobile robots $R1$ and $R2$ decrease as they move away together; the signal between them is, instead, constant until one of them stops at the corner and the other enters the gallery. In that moment (turn) the signal between BS and $R2$ falls abruptly and then is lost.
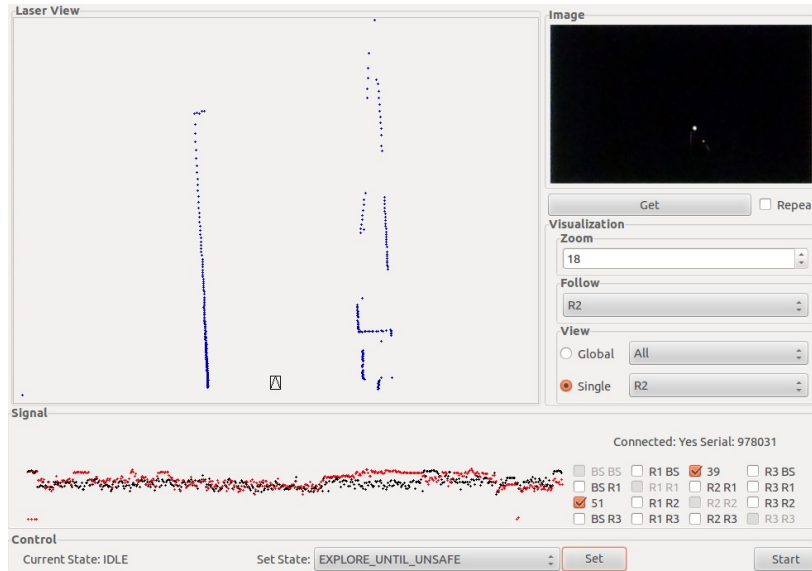
Fig. 17: User interface provided to the base station. It includes LIDAR data, radio signal and image visualization. In this case, the image is mostly black but it is possible to see a white spot which is the light entering the far end of the tunnel. It also provides information about the current state of the experiment.
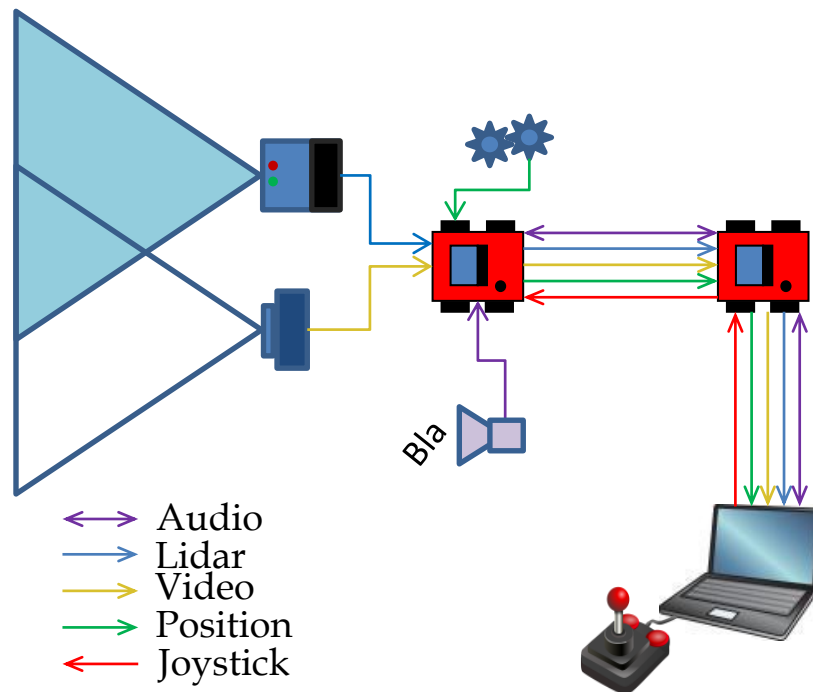
Fig. 18: Flows needed for communication and control. Audio flow was bi-directional, Joystick went from the base station to the leader robot while video from the leader robot to the base station. LIDAR and position flows were sent from each of the robots to the base station.
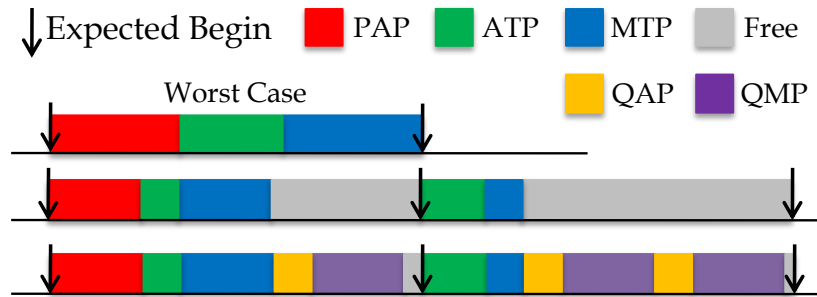
Fig. 19: RT-WMP phases after the introduction of the QoS extension. The QAP and QMP phases were added to the PAP, ATP and MTP of the original scheme. Multiple QAP and QMP phases could take place after the standard loop.
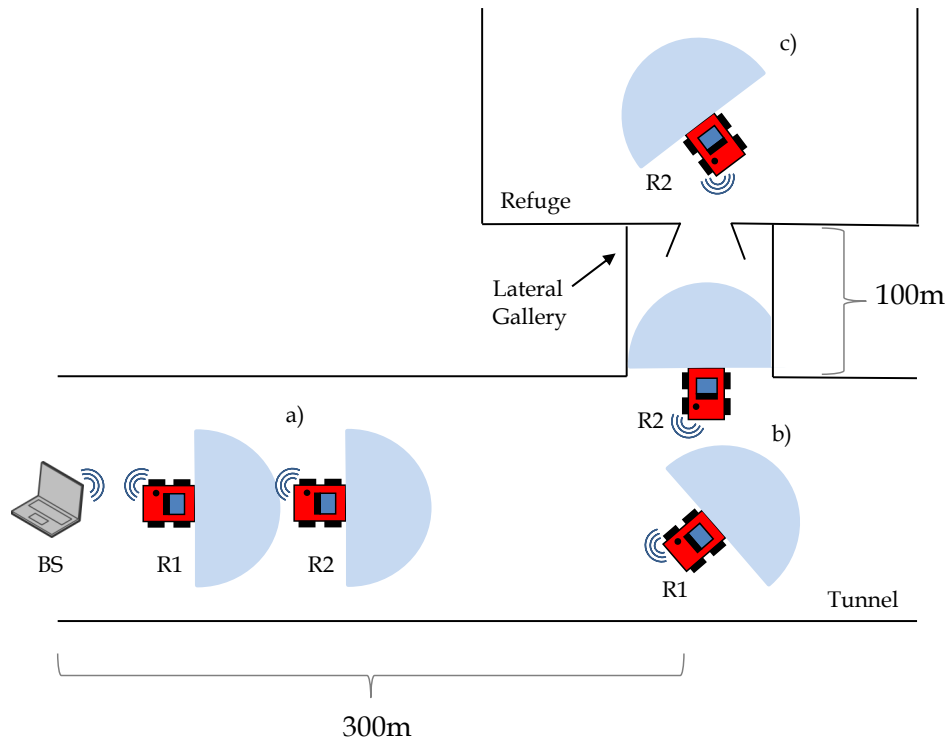
Fig. 20: Complete experiment. Robot leader, $R2$, must reach the refuge, maintaining connection the BS, by deploying a chain with $R1$. The robots are close to the base station and start to move toward the lateral gallery (a). After reaching the gallery, $R1$ stops in the corner and $R2$ moves toward the shelter (b). $R2$ reaches the shelter and communicate with the BS through $R1$ still at the corner (c).
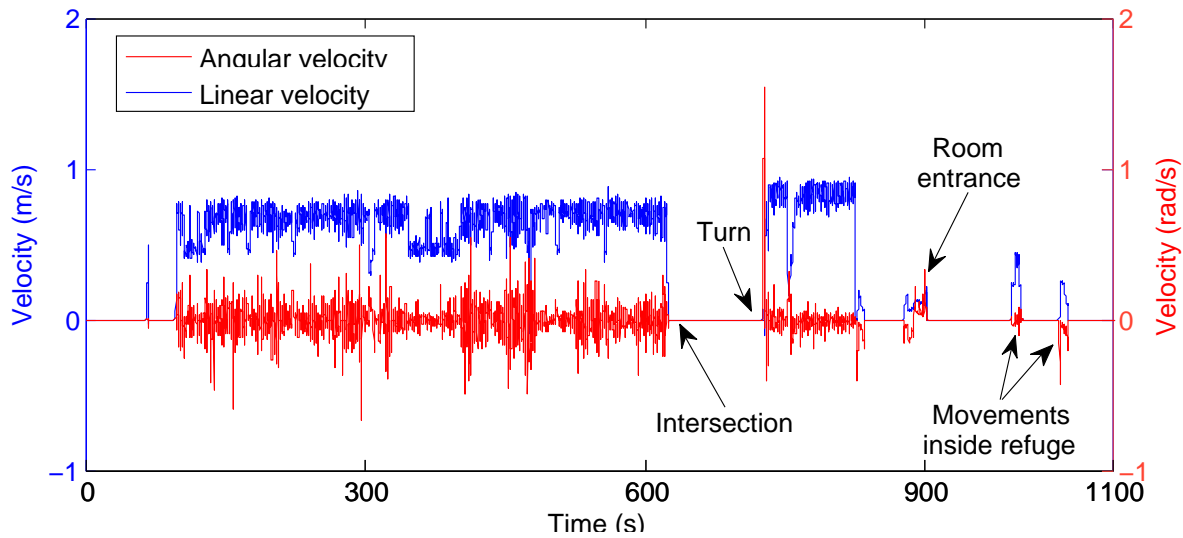
Fig. 21: Velocity of the leader robot during the experiment. It is possible to observe the different actions taken by the robot at its stopping at the intersection, its turn toward the gallery and the movement to enter and inside the shelter. Notice the noise in the measurement due to the uneven terrain.
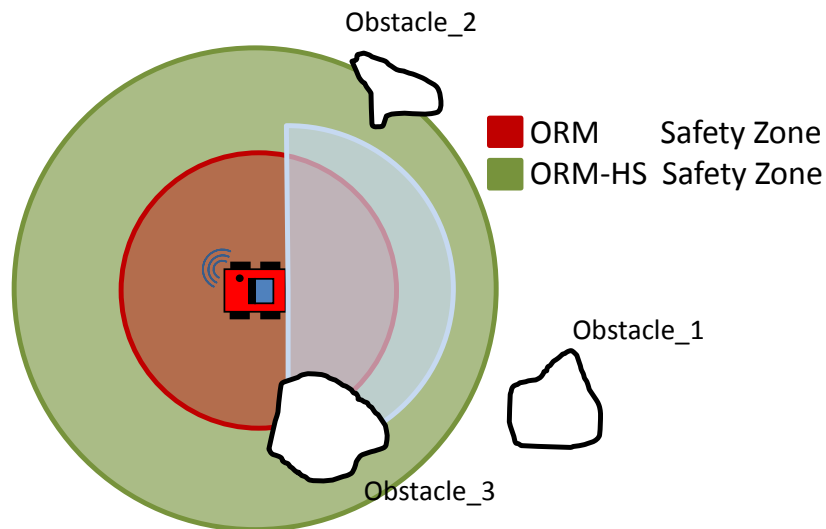
Fig. 22: The safety zones defined in the ORM-HS obstacle avoidance method. When an obstacle is within the ORM-HS safety zone, the maximum speed of the robot is reduced proportionally to the distance to the obstacle itself, until reaching the safety zone where the standard ORM algorithm is applied.
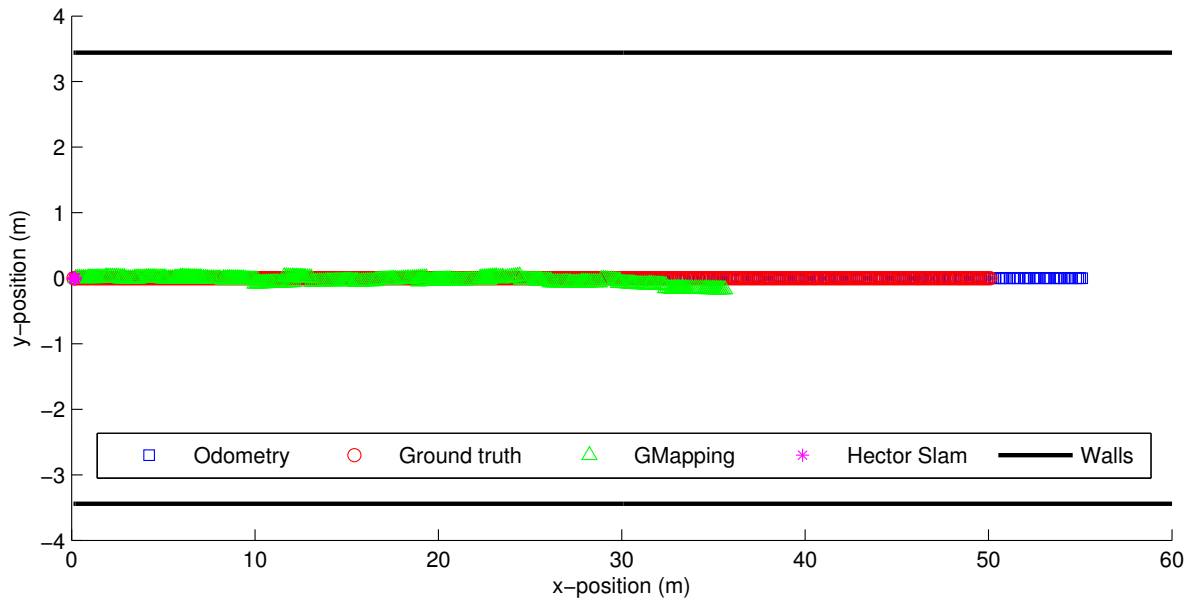
Fig. 23: Results of two scan-matching-based localization algorithm, odometry (which had a 10% error), and ground-truth in a simulated environment with completely smooth walls over a $50m$ stretch. Odometry provides a better result than scan-matching algorithms. Especially, Hector SLAM algorithm, which do not use odometry, reports no movement.
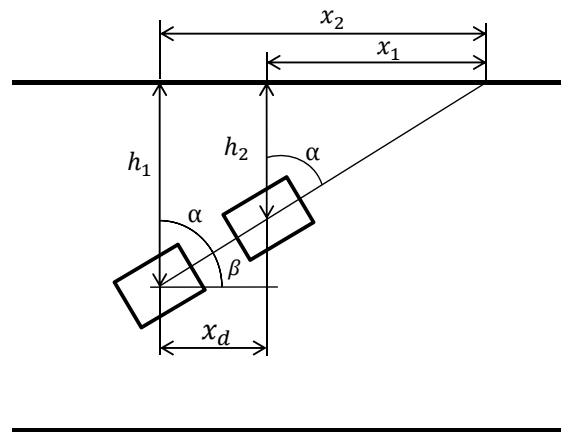
Fig. 24: Calculation of the $x-$axis displacement using a zig-zag trajectory. In each stretch, knowing the approaching angle, and computing the distances h_1 and h_2 it is trivial to obtain the linear displacement x_d.
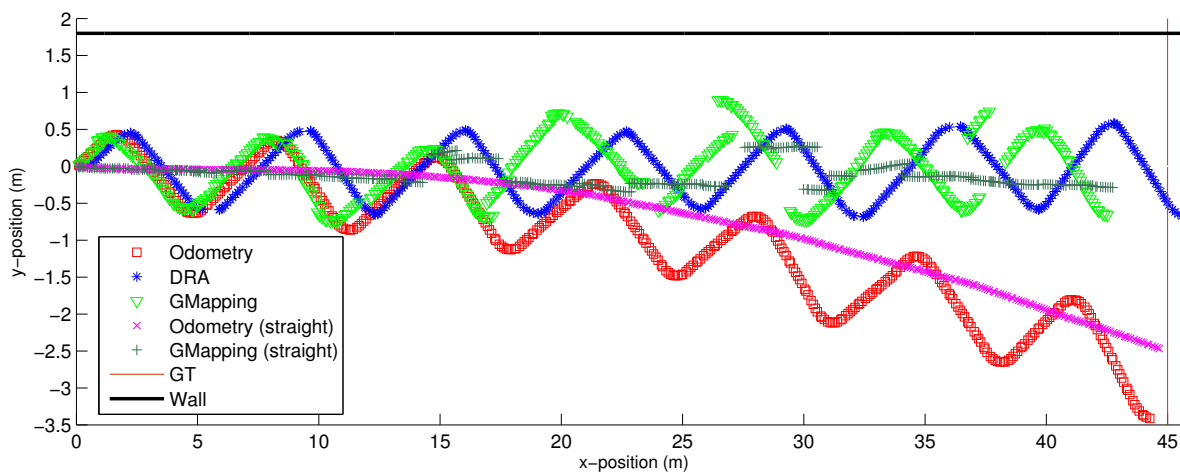
Fig. 25: Comparison between the DRA and other localization algorithms in a real-world experiment. The robot had to travel a stretch 45 m long. The figure shows the results of the experiments carried out over a zig-zag and a straight path. The GT line represents the final ground-truth position
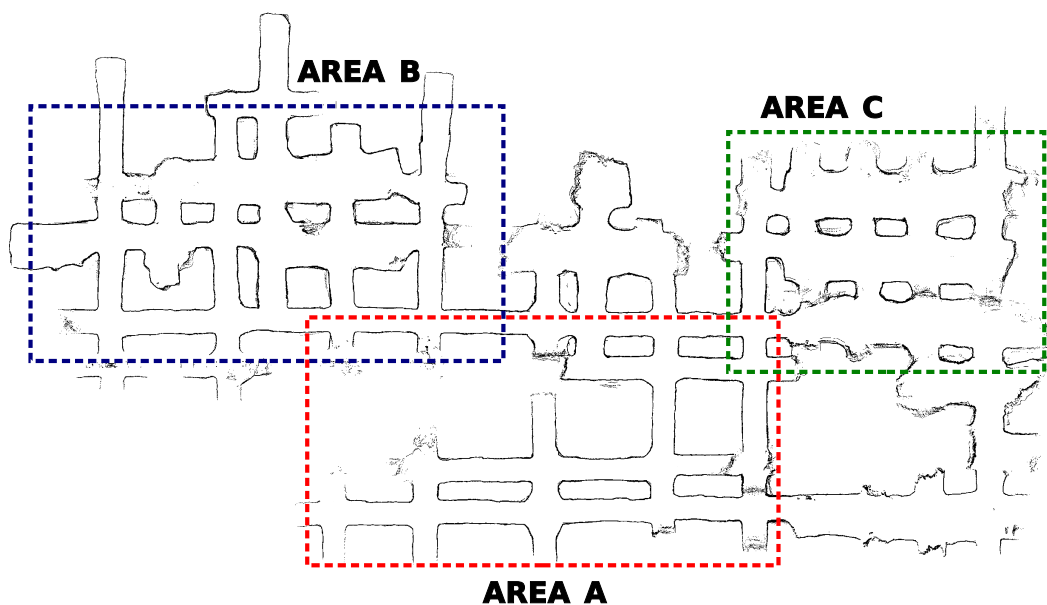
Fig. 26: Map of a portion of the Santa Marta mine (Spain) obtained in a previous mapping campaign used for verifying the system performance. The semantic features to be recognized were: *corridor*, *cross*, *left_hand*, *right_hand*, *left_turn*, *right_turn*, *end of corridor* and *T_intersection*. The area was divided into areas according to the different levels of exploitation.
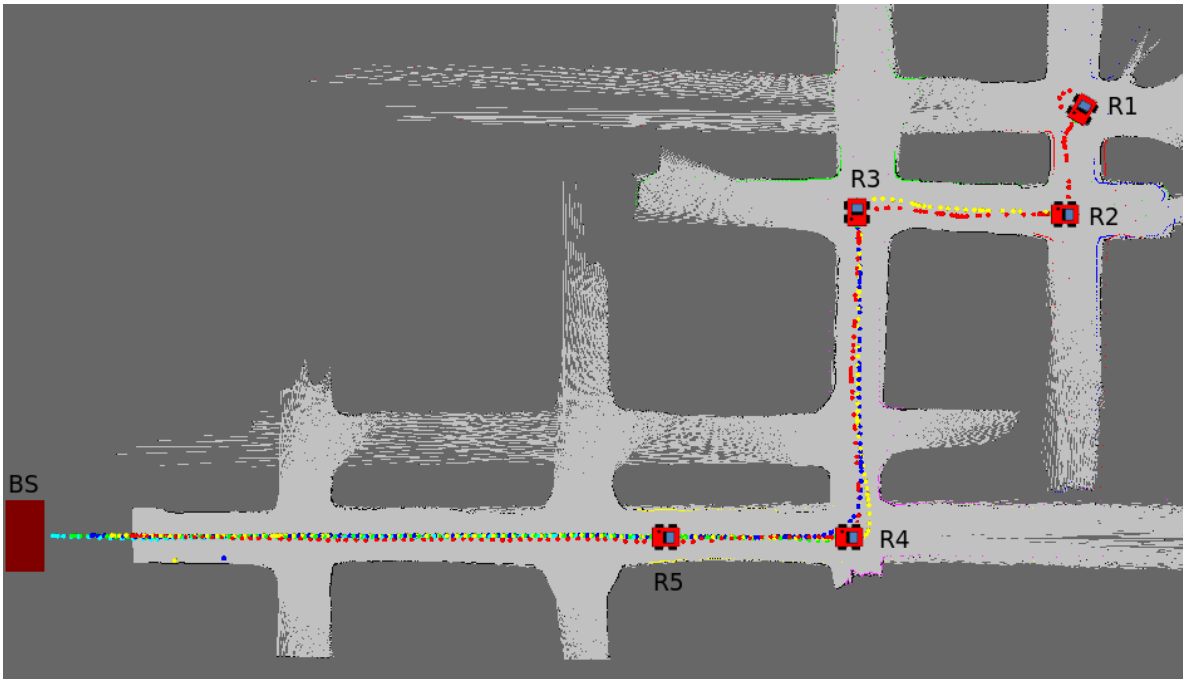
Fig. 27: Simulation of the whole intervention system with five robots and a base station in a simulated mine scenario. The plan was: 'pass 2nd cross, left_hand, 2nd cross, right_turn, 1st cross, reach next cross'.
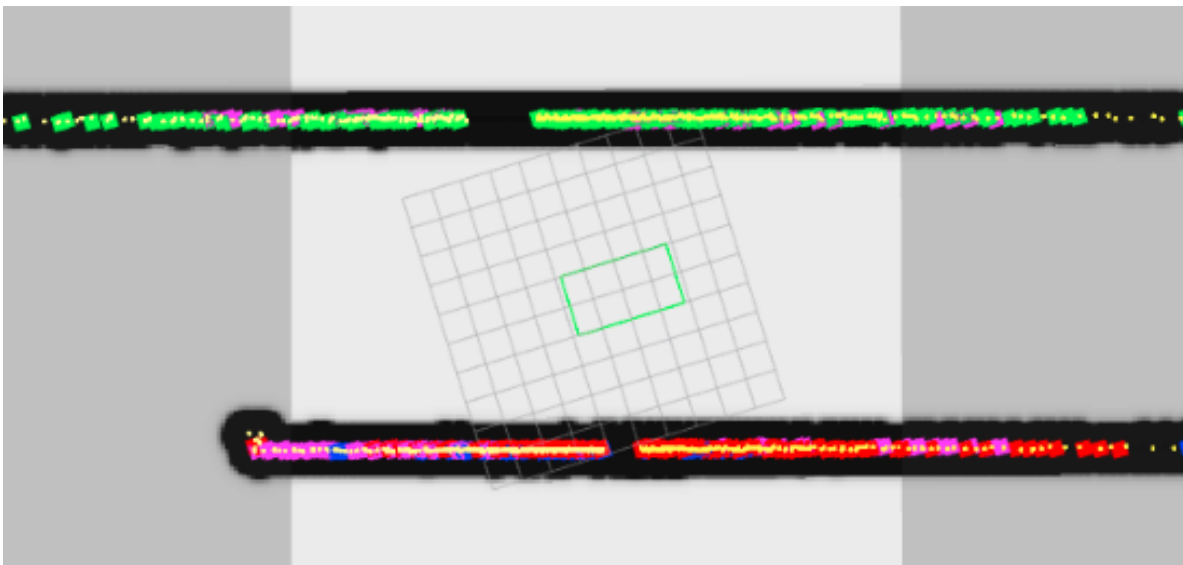

Fig. 28: Robot moving inside the tunnel. In yellow the raw LIDAR readings, in red and green the points belonging to the right and left walls identified by the algorithm and in violet and blue the other (shorter) lines detected.
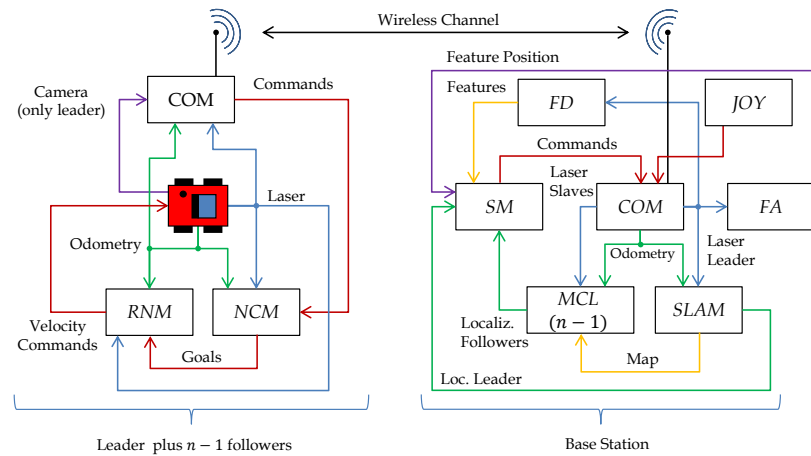
Fig. 29: Conceptual Architecture of the System. The different modules are: COmmunication Module (COM), Reactive Navigation Module (RNM), Navigation Control Module (NCM), Monte Carlo Localization (MCL), Simultaneous Localization And Mapping (SLAM), Feature Analyzer (FA), Feature Detector module (FD), State Machine module (SM), and teleoperation module (JOY).
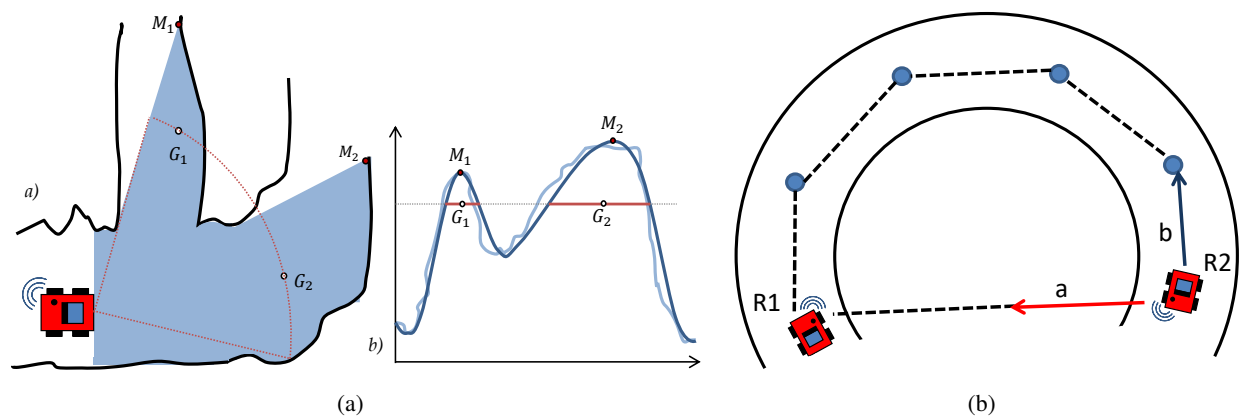

(a)   (b)

Fig. 30: Calculation of the goal for entering the lateral gallery (a). Trail of breadcrumbs for avoiding forces that would pull the followers toward the walls (b).
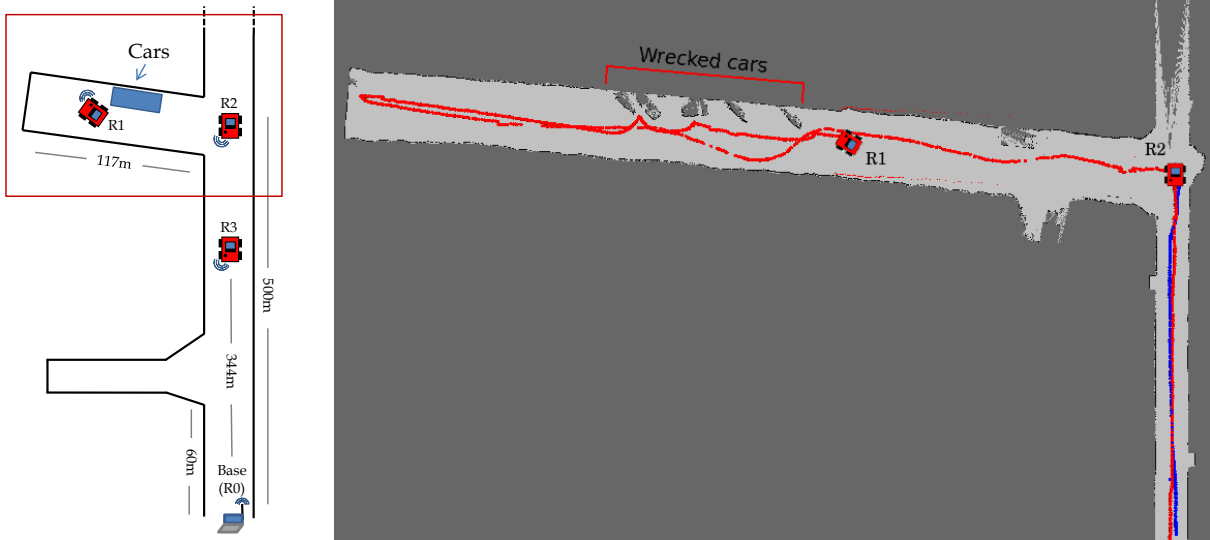
Fig. 31: Final configuration of the robots in the field experiment (left), and detail of the zone identified by the red rectangle (right), representing the leader robot, the relay robot in the corner, and the paths of each. The map shown was built on-the-fly at the base station with the data sent by the leader robot and provided to the followers' AMCL algorithms also running of the base station.

Fig. 32: Leader robot being teleoperated to explore the neighborhood of a wrecked car. The camera images and laser data were being used at the base station to control the robot.
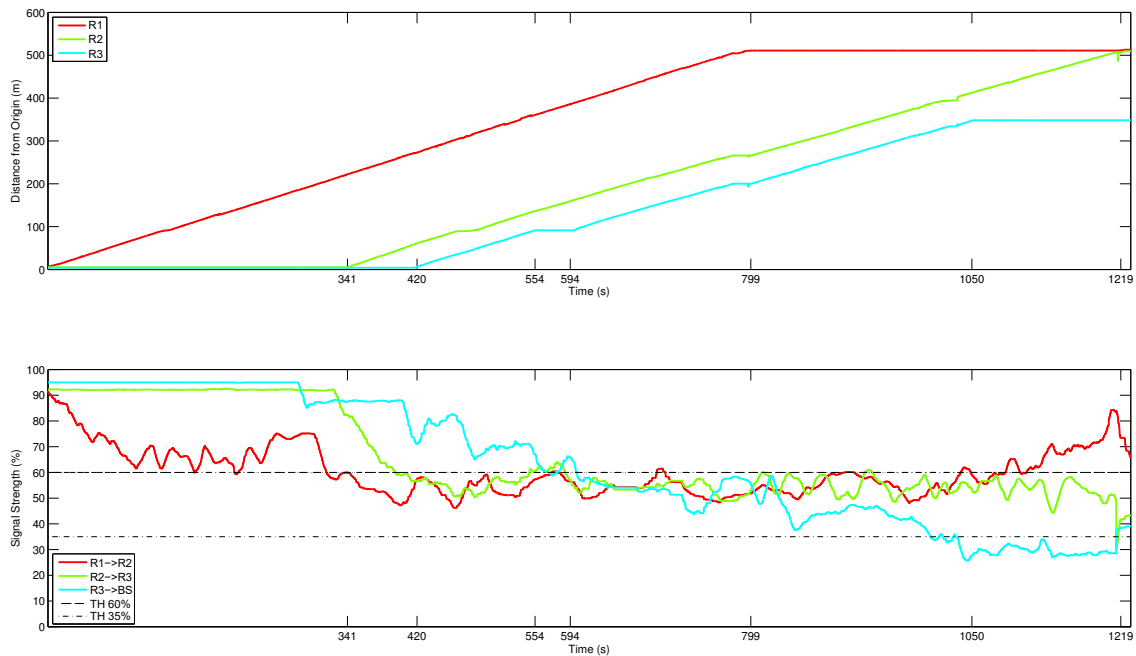

Fig. 33: Filtered RSSI among each pair of nodes (field experiment). The figure also shows the upper and lower (safety) thresholds and the timestamps in which each robot started moving or stopped.
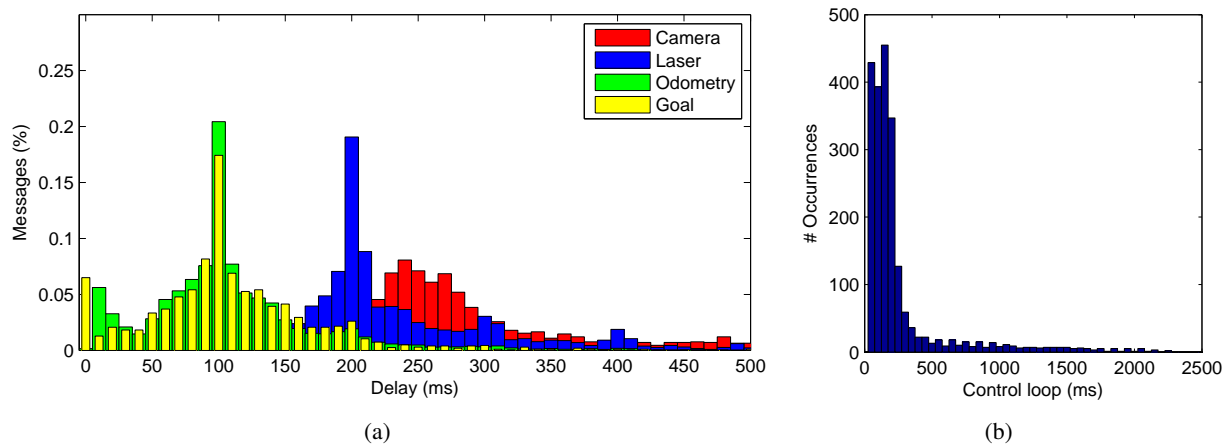
Fig. 34: Distribution of the inter-arrival times of the different flows (a) and control-loop period during the teleoperation (b). The figures show a set of Gaussian distributions around the expected periods and a Poisson distribution respectively.

Fig. 35: A real tunnel under construction (a). The robotized dumper used in our preliminary experiments (b). These environments use to be extremely challenging. Mud, water and potholes exceedingly complicates both the localization and the navigation of vehicles.