# Multi-robot Online Sensing Strategies for the Construction of Communication Maps

**Alberto Quattrini Li, Phani Krishna Penumarthi, Jacopo Banfi, Nicola Basilico, Jason M. O'Kane, Ioannis Rekleitis, Srihari Nelakuditi, and Francesco Amigoni**

**Abstract** This paper tackles the problem of constructing a communication map of a known environment using multiple robots. A communication map encodes information on whether two robots can communicate when they are at two arbitrary locations and plays a fundamental role for a multi-robot system deployment to reliably and effectively achieve a variety of tasks, such as environmental monitoring and exploration. Previous work on communication map building typically considered only scenarios with a fixed base station and designed offline methods, which did not exploit data collected online by the robots. This paper proposes Gaussian Process-based online methods to efficiently build a communication map with multiple robots. Such robots form a mesh network, where there is no fixed base station. Specifically, we provide two leader-follower online sensing strategies to coordinate and guide the robots while collecting data. Furthermore, we improve the performance and computational efficiency by exploiting prior communication models that can be built from the physical map of the environment. Extensive experimental results in simulation and with a team of TurtleBot 2 platforms validate the approach.

## 1 Introduction

Communication capabilities are fundamental for effective deployment and operation of multi-robot systems in several tasks – such as exploration (Banfi et al, 2016; Quattrini Li et al, 2016; Yamauchi, 1998), environmental monitoring (Dunbabin and Marques, 2012), and search and rescue (Murphy et al, 2016). In several multi-robot systems, some level of communication between robots is assumed, so that situational awareness and a high level of cooperation can be achieved (Amigoni et al, 2013). Recent work is explicitly considering communication in the multi-robot system design (Banfi et al, 2016; Gregory et al, 2015; Hollinger and Singh, 2012), so that the robots are not hindered too much by communication constraints (Tuna et al, 2013). To guarantee effective coordination throughout a task, robots need to know whether communication between two robots is possible from two arbitrary locations, before even moving there. Such information can be encoded in a *communication map*. Most of the papers designing communication-aware multi-robot systems share in common the assumption that robots have already a communication map. This, however, is not always a safe assumption, as a communication map is usually not available in practice (Liu and Cerpa, 2011). Other approaches assume a conservative communication model – such as limited range line-of-sight – which, as drawback, limits the capabilities of a multi-robot system. Having a reliable communication map is beneficial both as a standalone

A. Quattrini Li is with Department of Computer Science, Dartmouth College, Hanover NH, USA `alberto.quattrini.li@dartmouth.edu`.
P. K. Penumarthi, J. M. O'Kane, I. Rekleitis, and S. Nelakuditi are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA `penumarp@email.sc.edu`, `{jokane,yiannisr,srihari}@cse.sc.edu`.
J.Banfi is with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca NY, USA `jb2639@cornell.edu`.
N. Basilico is with the Department of Computer Science, University of Milan, Milano, Italy `nicola.basilico@unimi.it`.
F. Amigoni is with the Artificial Intelligence and Robotics Laboratory, Politecnico di Milano, 20133 Milano, Italy `francesco.amigoni@polimi.it`.

Fig. 1: TurtleBot 2 robots equipped with a WiFi dongle, deployed in the Engineering building at the University of South Carolina, building a communication map.
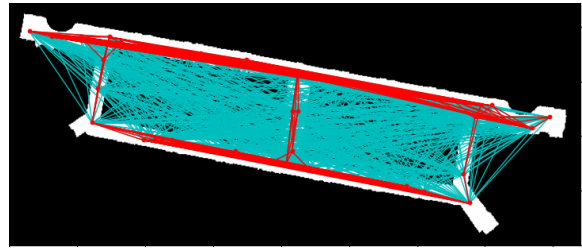


Fig. 2: Communication graph, in an environment where Turtlebot 2 robots have been deployed, built using a conservative communication model (edges in red) – i.e., limited range line-of-sight – and using the communication map built by our proposed method (edges in cyan).

task – for example to decide where to optimally place routers in an indoor environment – and to efficiently accomplish other robotic tasks so that, when choosing where to go, robots can account for communication constraints (Amigoni et al, 2017).

In this paper, we propose a system for the efficient online construction of communication maps by using a multi-robot system that collects communication quality measurements in the environment (see Figure 1 for the experimental setup used in this paper).

In the literature, typically, most of the work considers the problem of building a communication map with respect to a fixed router in the environment (Ferris et al, 2007; Mirowski et al, 2014). Other work designs some offline methods to collect signal strength measurements from a set of predefined locations (Hsieh et al, 2008; Riva et al, 2018). In many practical scenarios, a dense coverage is not feasible; for instance, when the time for the task is limited or when the environment is unknown.

Differently from the literature, in this paper, we consider a mesh network – a network where the nodes (robots) are not stationary and can communicate with each other in a range. Measurements are collected between pairs of robots as the mission progresses, allowing for online decisions on where to go. In particular, the main original contributions of this paper are:

– introducing a formal representation of communication maps based on Gaussian Processes (GPs);
– designing of two online strategies based on a leader-follower paradigm to decide where the robots should go to collect measurements;
– exploiting prior knowledge from wireless communication models to reduce the number of locations to be visited by the robots, so that the total traveled distance decreases;

– implementation of the proposed system in the Robot Operating System (ROS) framework;
– extensive experimental activities in simulation and with real robots.

The main motivation of this work is to improve coordination of multi-robot systems that can use such communication maps to reduce the hindrance of communication constraints. Figure 2 shows a communication graph, where edges encode whether robots can communicate between locations in an environment. Edges using a conservative communication model and utilizing the communication map built by our proposed method are shown in red and cyan, respectively. With our approach, the number of edges is more than 300% of those using a conservative communication model. This allows robots to accomplish tasks more efficiently as they have more freedom to plan to reach locations from where they can still maintain communication.

The construction and use of communication maps is particularly important in situations where communication infrastructures are not available or security issues prevent their use, leaving the adoption of ad hoc networks the only viable solution. Examples can be found in disaster or military scenarios (Ochoa and Santos, 2015).

This paper extends our preliminary results in (Banfi et al, 2017) and (Penumarthi et al, 2017). In particular, it includes more detailed descriptions of the methods in Section 4 and a more general formulation of the GP with an arbitrary mean function. In addition, an in-depth experimental analysis is presented in Section 5, adding experiments in simulation with the filtering technique, as well as experiments where we use different GP mean functions.

The structure of the paper is as follows. Section 2 presents the related work in communication map building. Section 3 formally describes the multi-robot communication map construction problem we address. Sec-

tion 4 presents the modeling, the sensing strategies, and the use of communication models to improve the process of building communication maps. Section 5 reports experimental results from several experiments in simulation and with real robots. Section 6 concludes the paper and discusses future research directions.

## 2 Related Work

The communication mapping problem has been typically studied assuming that a router is placed in a fixed position in the environment. Furthermore, many papers focus on the use of WiFi signal strength measurements for localization. Mirowski et al (2014) investigate different clustering-based methods to gather WiFi samples. Such WiFi samples can then be processed for finding WiFi signal fingerprints used for localization in an environment. Ferris et al (2007) use Gaussian Process Latent Variable Models for localizing a robot as well as reconstructing a topological connectivity graph from signal strength measurements. Similarly, Ladd et al (2005) use a Hidden Markov Model to track the position of a moving operator through signal strength measurements. Scholl et al (2012) build a hand-held system for mapping the radio signals by following a pre-specified pattern. The map is then used again for localization purposes. Kemppainen et al (2010) propose a method for a single robot to explore and build a model of a magnetic field in an environment. Such a model is then used by the robot to localize in the environment.

Some works provide methods for robust communication between robots. Im et al (2014) propose a robotic exploration system that exploits a prior communication model to predict disconnections and react accordingly. Fink et al (2013) design a system for determining robot trajectories to guarantee a communication network topology.

Other approaches exploit a single robot exploring an environment and localizing a radio source (Twigg et al, 2012) and multiple robots to map a stationary source without any coordination (Fink and Kumar, 2010). Hsieh et al (2008) propose some offline methods to compute efficient joint paths for small teams of robots, whose task is to collect signal strength measurements from a set of predefined locations. For the same problem, Riva et al (2018) derive some theoretical guarantees and propose greedy methods for an arbitrary number of robots.

A related body of work designs algorithms to explore different types of phenomena, such as position of obstacles (Yamauchi, 1998), water quality (Das et al, 2015; Manjanna et al, 2018), and gas leakage detection (Lilienthal et al, 2001). These works, while desirable, do not require a strict multi-robot coordination to

build a map of a phenomenon. A more abstract account of these problems is provided by methods for informative path planning, like for example Best et al (2018); Marchant et al (2014); Singh et al (2009); Zlot et al (2002). However, these methods cannot be directly applied to our problem since they usually assume that the measurements are performed by the single robots independently, while in our problem measurements are pairwise and require the coordinated activity of two robots.

In this paper, motivated by applications for building communication maps, we aim at tackling a realistic scenario with the following assumptions:

- There is not a fixed router, but all robots can communicate with each other, forming a mesh network. This makes the problem more challenging, as the robots need to coordinate with each other to collect measurements, at least, pairwise.
- The robots can perform online decisions on where to go to collect WiFi signal strength measurements. Because of the size of the state space, as shown in the next sections, it is not feasible for the robots to densely collect measurements over the environment. As such, strategies are necessary to perform online decision making.

In the following section, we formally state the problem addressed in this paper.

## 3 Problem Statement

We consider a known bounded planar environment with obstacles, where free space is denoted as $\mathcal{A} \subset \mathbb{R}^2$ and $\mathbf{p} \in \mathcal{A}$ denotes a location that can be occupied by a robot. There are $m$ mobile robots deployed in $\mathcal{A}$, starting from arbitrary initial locations $\mathbf{p}_1, \ldots, \mathbf{p}_m \in \mathcal{A}$. They can localize themselves within a global coordinate system, for example using a laser range finder or an RGB-d camera. In addition, they have an omni-directional WiFi transceiver (equal for all robots), to communicate with peers over the radio channel within the maximum communication range allowed by the device. The environment is represented by the robots as a two-dimensional occupancy grid, in which square cells with fixed size are marked either as free or occupied. The robots use such a representation to compute high-level paths between centers of arbitrary cells (that compose the set of discrete locations where the robots can be). The low-level controller of the each robot then accounts for kinematic constraints and smooths the trajectory as the robot executes the computed path.

The goal is to incrementally find a sequence of location pairs in $\mathcal{A}$ for the robots and to collect sig-

nal strength measurements between each pair of locations in order to build a communication map. Two metrics are considered to evaluate the performance of the system: traveled distance and quality of the communication map. Note that, as the robots move towards their next selected locations, measurements are collected along the paths and a partial communication map can be updated continuously.

Notice also that, in this paper, heading is not considered, as robots are equipped with omni-directional WiFi transceivers. An interesting future work is to equip the robot with directional WiFi transceivers to increase the communication range. In such a case, the goal is to find a sequence of pose pairs — i.e., heading needs to be considered.

More formally, a *communication map*, representing the information about communication links quality between ordered pairs of locations in $\mathcal{A}$, can be formalized as follows: a function $\hat{f} : \mathcal{A} \times \mathcal{A} = \mathcal{A}^2 \to \mathbb{R}_{\leq 0}$ estimating the received radio signal strength, RSSI $f : \mathcal{A}^2 \to \mathbb{R}_{\leq 0}$, in dBm, between any two locations $\mathbf{p}_i$ and $\mathbf{p}_j$. Such a function has a 4D input space, making the state space to explore large. The closer the values of $f$ and $\hat{f}$ (which are negative, see above) get to zero, the higher the actual and estimated, respectively, communication link reliability between $\mathbf{p}_i$ and $\mathbf{p}_j$. The range is typically between $-93$ dBm and $-10$ dBm when the transmitter and receiver are very far and very close, respectively. Our formulation can be easily adapted to other metrics used for estimating the quality of a communication channel. For sufficiently large environments, the availability of a communication link between two far locations can be excluded *a priori*, knowing the indicative range of the transceiver. Therefore, locations lying outside a maximum range $R_c$ can be excluded from the communication map. Let us call $\mathbf{x}_{ij} = (\mathbf{p}_i, \mathbf{p}_j)$ a pair of locations in the free space of $\mathcal{A}$ and $\hat{f}(\mathbf{x}_{ij})$ the estimate of the actual signal strength $f(\mathbf{x}_{ij})$ from $\mathbf{p}_i$ to $\mathbf{p}_j$. Communication links are not necessarily symmetric as shown experimentally by Heurtefeux and Valois (2012); thus, in general, $f(\mathbf{x}_{ij}) \neq f(\mathbf{x}_{ji})$. We build *static* communication maps in which the estimate of the signal strength between $\mathbf{p}_i$ and $\mathbf{p}_j$ is independent of time. In other words, we assume to work in environments that are static and where, as a consequence, signal strengths between locations do not change with time. The methods we propose can be generalized for dynamic environments with the introduction of detrimental effects (having uncertainty at $\mathbf{x}_{ij}$ increasing with time as long as no measurement is taken there) and penalizing the utility of taking measurements at those $\mathbf{x}_{ij}$ that expose a high variability among repeated measurements. The study of this extension is left for future works.
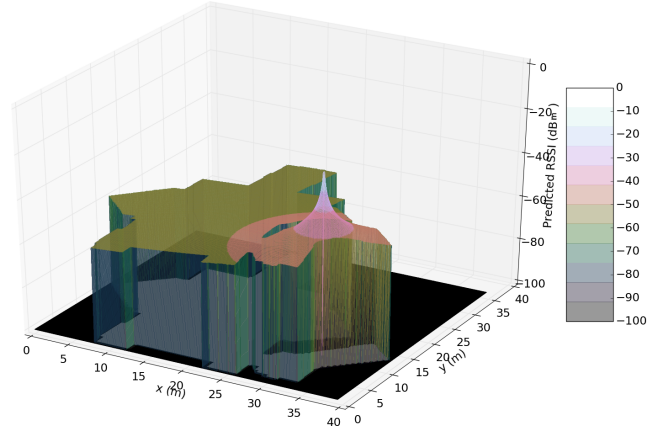


Fig. 3: Example of communication map, fixing the location of one of the two robots (where the peak is). Note that the signal strength is not plotted at locations in the environment inaccessible to the robots.

An example of a communication map instance is presented in Figure 3, by fixing the location $\mathbf{p}$ of one of the two robots (the transmitting robot in this case). Clearly, the closer to the transmitting robot, the higher the RSSI value. Note that there are infinite instances of such a representation of a communication map, one for every possible location $\mathbf{p}$ of the transmitting robot in the free space of the environment.

In general, we assume not to have any prior information on the communication map. However, as we will show in the next section, a communication map can be calculated as a prior from the physical map of the environment.

The aim of this paper is to design efficient online sensing strategies that produce a sequence of location pairs from which WiFi signal strength measurements can be collected in order to build a "good" communication map $\hat{f} \simeq f$ while limiting the number of acquired samples and, as a consequence, travelling the shortest possible distance. The similarity between $\hat{f}$ and $f$, namely the quality of the communication map, is calculated as the rooted mean square error between signal strength predicted by $\hat{f}$ and the actual measurement of $f$ over sampled pairs of locations.

## 4 A Gaussian Process-based Sensing System

In this section we present, first, how the communication map is represented in our approach; second, the strategies used to decide where the robots should go to collect measurements; third, the use of communication models as priors to improve the performance of the system in terms of computation time and traveled distance. Fig-

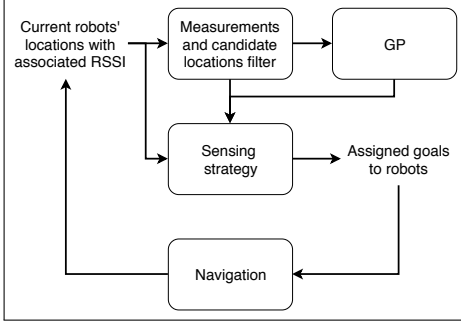ure 4 shows a depiction of the pipeline of the proposed system.



Fig. 4: Block diagram representing the proposed system for communication map building. Once the proposed system selects and assigns goals (locations) to the robots, the robots navigate towards them and, from the new locations, a new planning process restarts.

## 4.1 Gaussian Process-based Communication Map

Given the spatial correlation that radio signal strength displays, we use a GP (Rasmussen and Williams, 2006) to represent the communication map. Such a model can also be used to predict signal strengths in areas where measurements have not been collected yet, with an associated uncertainty. Specifically, $\hat{f}$ (defined in Section 3) can be calculated as a posterior distribution fitted over a set of noisy observations made by robots which navigate and coordinate in the environment to collect signal strength measurements. Assume that the robot team as a whole collected $q$ measurements over the environment. Let $\mathbf{Y} = [y^1, y^2, \ldots, y^q]^T$ and $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^q]^T$ be the set of those measurements and the set of the corresponding pairs of locations from where they have been collected, respectively; recall that $\mathbf{x}^i \in \mathcal{A}^2$. The signal strength observation $y^i = f(\mathbf{x}^i) + \epsilon$ is affected by additive sensing error, which is assumed to be i.i.d. with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. As covariance function $k(\mathbf{x}, \mathbf{x}')$, which expresses the spatial correlation between any two values of $f$, a radial basis kernel (RBF) is used, as done in the mainstream approach of GPs:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{2l^2}\right), \tag{1}$$

where the signal variance $\sigma_f^2$ and length scale $l^2$ are parameters that indicate the amplitude and the smoothness.

Let us call $K(\mathbf{X}_1, \mathbf{X}_2)$ the $q \times q$ matrix where $\mathbf{X}_1 = [\mathbf{x}_1^1, \ldots, \mathbf{x}_1^q]^T$ and $\mathbf{X}_2 = [\mathbf{x}_2^1, \ldots, \mathbf{x}_2^q]^T$ are the locations of two robots, $K_{ij} = k(\mathbf{x}_1^i, \mathbf{x}_2^j)$, and $I_q$ a $q \times q$ identity matrix. The correlation between the observed function values is represented by the following equation:

$$\text{cov}(\mathbf{Y}) = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I_q. \tag{2}$$

Here, we assume the GP to have a constant mean function, as typically done in literature (Rasmussen and Williams, 2006); therefore it is fully specified by the parameter vector $\theta = [\sigma_n^2, \sigma_f^2, l^2]^T$. In Section 5, we will show different mean functions. Such a parameter vector can be computed by finding the one that maximizes the observations log-likelihood, that is,

$$\theta^* = \arg\max_\theta \log P(\mathbf{Y} \mid \mathbf{X}, \theta), \tag{3}$$

where

$$\log p(\mathbf{Y} \mid \mathbf{X}, \theta) = \\ -\frac{1}{2}\left(\mathbf{Y}^\mathbf{T}\text{cov}(\mathbf{Y})^{-1}\mathbf{Y} - \log|\text{cov}(\mathbf{Y})| - q\log 2\pi\right). \tag{4}$$

To calculate an estimate of the signal strength, the optimal parameter vector $\theta^*$ is used in unobserved regions by evaluating the posterior. Specifically, called $\mathbf{W} = [\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^l]^T$ a set of arbitrary location pairs for which a query to get the corresponding signal strength estimate is performed, $P(f(\mathbf{W}) \mid \mathbf{X}, \mathbf{Y}) \sim \mathcal{N}(\mu_\mathbf{W}, \Sigma_\mathbf{W})$, where the mean vector is obtained as $\mu_\mathbf{W} = \mu(\mathbf{W}) + K(\mathbf{X}, \mathbf{W})^T K(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{Y} - \mu(\mathbf{X}))$ and represents the estimate $\hat{f}(\mathbf{W})$, while the covariance matrix is given by $\Sigma_\mathbf{W} = K(\mathbf{W}, \mathbf{W}) - K(\mathbf{X}, \mathbf{W})^T K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{X}, \mathbf{W})$. Note that the main diagonal of $\Sigma_\mathbf{W}$ is called *predictive variance* and quantifies the uncertainty of estimates in $\mathbf{W}$.

The GP provides a mechanism to integrate noisy readings collected in the environment into a posterior distribution of the signal strength. Such a posterior can be used to obtain link estimates with quantified uncertainty. To deploy such a mechanism in real multi-robot settings, two issues need to be tackled. The first one is to design a mission execution scheme according to which robots repeatedly coordinate, gather sensor data, share information, and update the communication map, maximizing Equation (4). The second problem is to design utility functions to optimize the online selection of joint data-gathering locations. Both problems are central in the definition of *sensing strategies*, presented in the next section.

## 4.2 Sensing Strategies

A signal strength measurement at the joint location $\mathbf{x}_{ij} = (\mathbf{p}_i, \mathbf{p}_j)$ is obtained by a robot at location $\mathbf{p}_i$

polling another robot at location $\mathbf{p}_j$. Given the 4D state space that needs to be explored ($\mathcal{A} \times \mathcal{A}$), it is not feasible to densely cover it. The key idea of the designed sensing strategies is to privilege data acquisition locations that are informative, namely that are expected to induce high reductions in the current communication map's uncertainty, thus limiting the distance traveled by robots and providing high-quality communication maps.

Furthermore, robots might differ in terms of computational capabilities (as often happening in heterogeneous multi-robot systems). This characteristic of the robotic system would add some constraints for the GP parameter estimation process. As such, we design strategies for the following two settings. In the *homogeneous* setting, we assume that any robot has an onboard computer with sufficient computational power to run the GP model. In the *heterogeneous* setting, only few robots can compute online a GP model, while others (cheap/basic platforms) just navigate and collect measurements.

To construct the communication map, robots need to collect *pairwise measurements*. This requires the presence of explicit coordination between at least two robots, otherwise we cannot guarantee that measurements are performed between informative locations of the environments. We present two strategies that employ a role-based *leader-follower* paradigm. In particular, the robots have the following roles: *leaders* are robots that maintain a communication map by iteratively updating the GP model according to the measurements acquired so far. They are also in charge of choosing the best locations to visit in coordination with the corresponding *followers*. In general, multiple leaders can be present in the robotic team. They can share information during the mission, as well as meet at the end of the mission to merge their measurements to build a final communication map. In our system, the leaders compute the computationally expensive GP models (complexity is cubic in the number of collected measurements, as illustrated in what follows), but distributed computation of GPs could be considered (Deisenroth and Ng, 2015).

Coordination between different groups of leader-followers is achieved by broadcasting or selectively relaying relevant information to be shared in a multi-hop fashion. In practice, the two strategies give rise to a "grouply distributed" multi-robot system. Each group of robots acts as an autonomous entity and benefits from episodic encounters with robots belonging to other groups, while, at the same time, each follower is subject to the directives of the corresponding leader. The computation is distributed over groups of robots that are in intermittent communication with the other robots of the same group (as detailed later). Both strategies

favor selection of locations in regions of $\mathcal{A}^2$ currently displaying high predictive variance and try to spread the robots. The strategy we propose for the homogeneous setting is called *Pairwise Mapping* (PM), while the one tailored for the heterogeneous setting is called *Region Mapping* (RM).

### 4.2.1 Pairwise Mapping in Homogeneous Settings

With the PM strategy, the team is divided in pairs of robots, where one robot acts as *leader* and the other one as *follower*. The leader iteratively drives itself and the follower to take measurements in the pairs of locations $\mathbf{p}_l^*, \mathbf{p}_f^* \in \mathcal{A}^2, |\mathbf{p}_l^* - \mathbf{p}_f^*| \leq R_c$ currently displaying a high predictive variance in the current communication map. While moving to such locations, the two robots poll other robots in the environment for additional measurements. The polling frequency is adapted from the estimated mission length, to limit the number of acquired samples. Coordination between different leader-follower pairs is achieved in two ways. First, each robot broadcasts the waypoints of its current path, along with those of its corresponding teammate. In this way, the other leaders know which regions of $\mathcal{A}^2$ can be excluded from their planning because already visited and sensed by other robots. Second, each robot maintains an updated collection of all the data gathered by the team by periodically asking its teammates to selectively relay the portion of the collected dataset not yet received by any other team member. This is done to make the most updated dataset available to each leader for training a new GP at replanning time.

In case two robots are not able to communicate from two target locations $\mathbf{p}_l^*, \mathbf{p}_f^*$, a recovery mechanism is adopted. Specifically, a pair of backup locations are selected and assigned to robots. The selection criterion — see below — guarantees the presence of a communication link. Denoted $r_l$ and $r_f$ the leader and follower robots, respectively, the PM strategy is formally defined by these steps:

(1) $r_l$ and $r_f$ are connected;

(2) $r_l$ selects the target locations, $\mathbf{p}_l^*, \mathbf{p}_f^*$, and backup locations $\mathbf{p}_l^b, \mathbf{p}_f^b$ (see Algorithm 1), informing $r_f$;

(3) $r_l$ and $r_f$ agree on a deadline $t_d$ to reach $\mathbf{p}_l^*$ and $\mathbf{p}_f^*$, calculated according to the length of the path between the current positions of the robots and their target locations and the robots' speed (see below);

(4) $r_l$ and $r_f$ move to $\mathbf{p}_l^*$ and $\mathbf{p}_f^*$, opportunistically exchanging the collected signal data and the undertaken path with other teammates, and possibly polling them to get additional measurements;

(5) if $r_l$ and $r_f$ are connected before $t_d$, go to Step (1); otherwise, go to the next step;

---

**Algorithm 1** PM planning algorithm for a pair $r_l, r_f$

---

**Input:** $\mathcal{D}$ (dataset), $\mathbf{p}_l, \mathbf{p}_f$ (current locations), $\mathcal{P}$ (teammate paths),
$\quad \mathcal{M}$ (environment map), $n_s^{\mathrm{PM}}$ (# samples), $d_{\min}$ (minimum distance)
$\quad$ tance)
**Output:** $\mathbf{p}_l^*, \mathbf{p}_f^*$ (target locations), $\mathbf{p}_l^b, \mathbf{p}_f^b$ (backup locations)
1: $\mathcal{J} \leftarrow$ extrapolateJointLocations$(\mathcal{P})$
2: $\mathcal{X} \leftarrow$ sampleLocations$(\mathcal{M}, n_s^{\mathrm{PM}}, \mathcal{J}, d_{\min})$
3: $\theta \leftarrow$ learnHyperParams$(\mathcal{D})$
4: $\mathbf{\Sigma}_{\mathcal{X}} \leftarrow$ predictUncertaintyGP$(\mathcal{X}, \mathcal{D}, \theta)$
5: $\mathbf{p}_1^*, \mathbf{p}_2^* \leftarrow \arg\max_{(\mathbf{p}_1, \mathbf{p}_2) \in \mathcal{X}} \{ \mathbf{\Sigma}_{\mathbf{xx}} + \mathbf{\Sigma}_{\mathbf{x'x'}} \} (\mathbf{x} = (\mathbf{p}_1, \mathbf{p}_2), \mathbf{x}' =$
$\quad (\mathbf{p}_2, \mathbf{p}_1))$
6: $\mathbf{p}_l^*, \mathbf{p}_f^* \leftarrow$ assignMinMaxDist$(\mathbf{p}_1^*, \mathbf{p}_2^*, \mathbf{p}_l, \mathbf{p}_f)$
7: $\mathbf{p}_l^b, \mathbf{p}_f^b \leftarrow$ backupDest$(\mathbf{p}_l^*, \mathbf{p}_f^*, \mathcal{M})$

---

(6) $r_l$ and $r_f$ set $\mathbf{p}_l^b$ and $\mathbf{p}_f^b$ as target locations; as soon as they regain connection, go to Step (1).

In particular, in Step (3), the value of the deadline $t_d$ can simply be chosen as the maximum estimated arrival time of the two robots, calculated knowing the length of the path and the speed of the robots, augmented by some tolerance value. In general, such a tolerance value depends on the complexity of the environment and on the robots' localization uncertainty. From some preliminary experiments, we selected to augment the maximum arrival time of $r_l$ and $r_f$ by 4% for both simulated and real robots.

Algorithm 1 describes how each leader chooses the next pairs of target and backup locations. The algorithm takes as input the collected signal data $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$, the current leader-follower locations $\mathbf{p}_l, \mathbf{p}_f$, the paths currently undertaken by the other leader-follower pairs $\mathcal{P}$, and a map of the physical environment $\mathcal{M}$. As additional parameters, the algorithm requires the number of samples $n_s^{\mathrm{PM}}$ to generate, and a value $d_{\min}$ used to exclude some location pairs from planning as they will be probably close to pairs already visited by other subteams. Initially (Lines 1-2), the algorithm samples candidate pairs of locations $\mathcal{X}$ where to send $r_l$ and $r_f$. Specifically, it first computes the set $\mathcal{J}$ of joint waypoints that the other leader-follower teams will traverse while moving to their target locations, assuming a constant speed along their path; then, it selects $n_s^{\mathrm{PM}}$ samples from $\mathcal{A}^2$ at distance not greater than $R_c$, pruning those that are not at least $d_{min}$ far apart (in travel distance) from each location pair in $\mathcal{J}$. In Line 3, a new GP is trained with the gathered data, and in Lines 4-5 the most uncertain pair $\mathbf{p}_1^*, \mathbf{p}_2^* \in \mathcal{X}$ is selected. The two locations are then assigned to the robots $r_l$ and $r_f$ to minimize the maximum traveled distance. This implicitly optimizes the energy consumption and produces $\mathbf{p}_l^*, \mathbf{p}_f^*$ (Line 6). Finally, in Line 7, backup locations $\mathbf{p}_l^b, \mathbf{p}_f^b$ are chosen as the pair of points closest to the target locations, and for which a safe communication link is guaranteed. We assume that two robots can

always communicate if within a given distance (see Section 5 for distance values we use in experiments).

### 4.2.2 Region Mapping in Heterogeneous Settings

Also the Region Mapping (RM) strategy is based on a leader-follower paradigm. Differently from the PM strategy, leader robots (the more expensive platforms with powerful computers) have an arbitrary but fixed number of followers $R_f = \{r_{f_1}, \ldots, r_{f_k}\}$. As in the previous strategy, leaders are in charge of maintaining the GP model. However, with this strategy, they iteratively drive the followers in *regions* $R^*$ with high predictive variance. Once a region has been selected, the leader moves to its center $\mathbf{p}_c$. At the same time, the followers move towards safe positions $\mathcal{S}_f = \{\mathbf{p}_1^s, \ldots, \mathbf{p}_k^s\}$, where they can acknowledge that the leader has reached its goal. Then, the followers move along pre-computed paths $\mathcal{P}_f = \{p_1, \ldots, p_k\}$ that can significantly reduce the prediction uncertainty in the region of $\mathcal{A}^2$ centered in $\mathbf{p}_c$. If a follower $r_{f_i}$ cannot communicate with the leader when it gets to the end of its path, it will move to its backup location chosen from a pre-computed set $\mathcal{B}_f = \{\mathbf{p}_1^b, \ldots, \mathbf{p}_k^b\}$. The coordination among teams is achieved by letting the leaders choose regions to visit whose centers are sufficiently far apart. As with the PM strategy, robots can also acquire additional measurements, while selectively relaying the gathered dataset. The rationale behind this strategy is to possibly avoid re-visiting the same region, by lowering down the uncertainty around a fixed leader position, rather than taking only sparse measurements. Formally, leader $r_l$ and followers $R_f$ act as follows:

(1) $r_l$ is initially connected to each follower in $R_f$;
(2) $r_l$ decides the region $R^*$ to explore next, computing $\mathbf{p}_c, \mathcal{P}_f, \mathcal{S}_f, \mathcal{B}_f$ (see Algorithm 2);
(3) $r_l$ moves to $\mathbf{p}_c$, while the followers move to $\mathcal{S}_f$;
(4) followers follow their paths in $\mathcal{P}_f$, $r_l$ remains still;
(5) followers regain connection with $r_l$ either at end of their paths, or by going to their backup locations in $\mathcal{B}_f$; go to Step (1).

Before executing Step (4), once the leader reaches $\mathbf{p}_c$, it waits until all followers communicate with the leader. In this way, excessive delays of some of them can be handled by reorganizing the paths of the remaining ones.

Algorithm 2 describes the RM strategy. The algorithm takes as input the collected signal data $\mathcal{D}$, the current leader and followers locations $\mathbf{p}_l, \mathbf{P}_f$, the centers of the regions being visited by other groups $\mathbf{P}_c$ as known by $r_l$, a map of the physical environment $\mathcal{M}$, the number of regions to sample $n_s^{\mathrm{RM}}$, a parameter

$d_{\min}$ used for coordinating the spreading of the different groups, and the number of waypoints to visit in the new region $n_w$. As with the previous strategy, a suitable choice of $d_{\min}$ avoids that two different subteams of robots with a close replanning time decide to start mapping the same region of the environment.

In Line 1, $n_s^{\mathrm{RM}}$ candidate circular regions $\mathcal{R}$ with diameter $2R_c$ are selected. In Line 2, a new GP is trained with the gathered data, and in Lines 3-4 the region $R^* \in \mathcal{R}$ with the highest mean predictive variance is selected. In Line 5, a set $\mathcal{W}$ of $n_w$ waypoints to visit is selected from a fine-grained discretization of $\mathcal{A} \cap R^*$ according to the following method: iteratively choose the point $\mathbf{p}_w$ displaying the highest sum of predictive variance when paired with $\mathbf{p}_c$ and sufficiently far apart from the already chosen waypoints (this distance threshold could also be the same $d_{\min}$). Notice that this spacing is required because it would be inefficient to choose a waypoint very close to another one, since the uncertainty of the former will be already reduced by visiting the latter. In Line 6, the initial safe destinations for the followers $\mathcal{S}_f$ are selected as follows: first, followers are iteratively assigned to the closest locations in $\mathcal{W}$ guaranteeing a safe communication link to $\mathbf{p}_c$. $\mathcal{W}$ is updated accordingly, so that, at the end of this first phase, $\mathcal{W} \cap \mathcal{S}_f = \emptyset$. Then, if any follower remains unassigned, its corresponding safe location is chosen as the closest location from its current location in $\mathcal{A}$ guaranteeing a safe communication link. In Line 7, the remaining waypoints are assigned to the robots with the aim of minimizing the maximum followers' traveled distance. Finally, in Line 8, the backup locations are computed as in the PM strategy.

Note that, in Line 7, the objective of minimizing the bottleneck traveled distance gives rise to the Multiple Traveling Salesman Path Problem, which is NP-hard (Rekleitis et al, 2008). This problem can be formalized in terms of a simple Mixed Integer Linear Program (MILP) as follows. With a slight notation over-

load, let $f$ be a generic follower robot, and $V(f)$ be the set $\mathcal{W} \cup \{\mathbf{p}_f^s\}$. We define three sets of variables: $x_{ij}^f$, with $i, j \in V(f)$, is a binary variable taking value 1 iff in the path of $f$ a visit to $i$ is followed by a visit to $j$; $y_i^f$, with $i \in \mathcal{W}$, is a binary variable taking value 1 iff waypoint $i$ is visited by $f$; $u_i^f$, with $i \in \mathcal{W}$, is a continuous variable representing the possible position of waypoint $i$ in the path of $f$. A continuous variable $b$ is also defined to represent the objective function value. The MILP model reads as follows:

$$\min b \quad \text{s.t.} \tag{5}$$

$$\sum_{j \in V(f)} x_{\mathbf{p}_f^s j}^f = \sum_{i \in V(f)} x_{i \mathbf{p}_f^s}^f \quad \forall f \in R_f \tag{6}$$

$$\sum_{j \in V(f)} x_{\mathbf{p}_f^s j}^f = 1 \quad \forall f \in R_f \tag{7}$$

$$\sum_{i \in V(f)} x_{iw}^f = \sum_{j \in V(f)} x_{wj}^f \quad \forall f \in R_f, \forall w \in \mathcal{W} \tag{8}$$

$$\sum_{j \in V(f)} x_{wj}^f = y_w^f \quad \forall f \in R_f, \forall w \in \mathcal{W} \tag{9}$$

$$\sum_{f \in R_f} y_w^f = 1 \quad \forall w \in \mathcal{W} \tag{10}$$

$$b \geq \sum_{i \in V(f)} \sum_{j \in \mathcal{W}} d_{ij} x_{ij}^f \quad \forall f \in R_f \tag{11}$$

$$2 \leq u_w^f \leq |\mathcal{W}| + 1 \quad \forall f \in R_f, \forall w \in \mathcal{W} \tag{12}$$

$$u_i^f - u_j^f + 1 \leq |\mathcal{W}|(1 - x_{ij}^f) \quad \forall f \in R_f, \forall i, j \in \mathcal{W}. \tag{13}$$

Constraints (6) and (7) enforce, for each follower, a path that starts and ends at the corresponding $\mathbf{p}_f^s$. Constraints (8) and (9) guarantee path consistency. Constraints (10) impose that each waypoint must be visited by exactly one follower. Constraints (11) bind the objective function to the maximum traveled distance (notice that starting points are excluded from the inner summation, as the problem does not require building a tour). Finally, Constraints (12) and (13) avoid the presence of subtours. The time required to solve this model to optimality rapidly grows with the size of the input. However, preliminary experiments showed that it is usually better to keep the size of $\mathcal{W}$ relatively small (i.e., only a few points), as robots also periodically obtain measurements while moving along their planned paths, as in the PM strategy (Section 4.2.1).

Both proposed methods consider all possible locations uniformly sampled from the free space of the environment, as no prior knowledge about the communication map is assumed. This could result in an overhead in terms of traveled distance, as well as computing time for updating the GP. However, in the networking literature, researchers proposed some communication models for indoor environments that could give a prior to guide the modeling and the incremental sensing process. In the following, we present some of such communication models and their use in our proposed system.

---

**Algorithm 2** RM planning algorithm for a group $r_l, R_f$

---

**Input:** $\mathcal{D}$ (dataset), $\mathbf{p}_l$, $\mathbf{P}_f$ (current followers locations), $\mathbf{P}_c$ (other region centers), $\mathcal{M}$ (environment map), $n_s^{\mathrm{RM}}$ (# region samples), $d_{\min}$ (minimum distance), $n_w$ (# waypoints)

**Output:** $\mathbf{p}_c$ (new region center), $\mathcal{P}_f$ (followers' paths), $\mathcal{S}_f$ (initial locations), $\mathcal{B}_f$ (backup locations)

1: $\mathcal{R} \leftarrow \texttt{sampleRegions}(\mathcal{M}, n_s^{\mathrm{RM}}, \mathbf{P}_c, d_{\min})$
2: $\theta \leftarrow \texttt{learnHyperParams}(\mathcal{D})$
3: $\overline{\Sigma}[\mathcal{R}] \leftarrow \texttt{meanVarianceRegionsGP}(\mathcal{R}, \mathcal{D}, \theta)$
4: $R^* \leftarrow \underset{R \in \mathcal{R}}{\arg\max} \, \overline{\Sigma}[R]$
5: $\mathcal{W} \leftarrow \texttt{getWaypoints}(R^*, n_w, \mathcal{D}, \theta)$
6: $\mathcal{S}_f \leftarrow \texttt{initialDests}(\mathbf{P}_f, \mathcal{W}, \mathcal{M})$
7: $\mathcal{P}_f \leftarrow \texttt{assignMinMaxDistPaths}(\mathcal{S}_f, \mathcal{W}, \mathcal{M})$
8: $\mathcal{B}_f \leftarrow \texttt{backupDest}(\mathcal{P}_f, \mathcal{M})$

## 4.3 Communication Model-Based Filtering

In this section, first, we present communication models from the networking literature, with an evaluation of their fidelity; second, we show how such models can be used in our approach to filter signal strength measurements polled by the robots when updating the communication map and to select locations where robots should go.

### 4.3.1 Prior from Communication Models

In general, it is hard to directly estimate the RSSI value knowing the map of an environment, because, for example, the compositions of the obstacles are not known. However, in the communication literature, some models — such as Free Space, Two-ray, Ten-ray, Wall Attenuation Factor, and Multi Wall Attenuation (Goldsmith, 2005) — have been shown to have relatively good accuracy in estimating the signal's power loss during propagation (also known as *path-loss*)[1]. These models vary in terms of computational complexity and accuracy. Furthermore, each model usually has several parameters and quantifying them accurately is hard, as they depend on the specific physical environment. The challenge in using such models is to optimize the related parameters to adapt the models to the signal strength measurements taken from the actual environment (Bahl and Padmanabhan, 2000).

    We select and evaluate four communication path-loss models with varying complexity that have been tested in indoor environments. In the following, we report the equations for the different models[2], referring to transmitter $\mathbf{p}_i$ and receiver $\mathbf{p}_j$.

*Distance Model (DIST)* A (free space) distance-based path-loss model assumes that the signal is passing through vacuum; the path-loss observed by the signal depends only on the Euclidean distance between locations of transmitter and receiver (Goldsmith, 2005; Rappaport, 1996):

$$L_{\mathrm{dist}}(\mathbf{p}_i, \mathbf{p}_j) = -10 \log_{10} \left[ \frac{\sqrt{G_L}\lambda}{4\pi d(\mathbf{p}_i, \mathbf{p}_j)} \right]^2 , \qquad (14)$$

where $G_L$ is the product of transmitter and receiver antenna gains[3]; $\lambda$ is the wavelength of the transmitted

---

[1] It should be noted that all these path-loss models are independent of the used communication frequency, i.e., they do not restrict the analysis to WiFi/LTE, etc.

[2] We slightly change the notation of the original papers to make notation uniform and consistent with that used in this paper and to highlight variables and parameters.

[3] Gain is defined in terms of the antenna's capability to send/receive signals in a direction.

signal (m); and $d()$ is the Euclidean distance between two locations (m).

*Wall Attenuation Factor Model (WAF)* An empirical model (Bahl and Padmanabhan, 2000), which assumes the physical map of the environment to be available beforehand, as path-loss is influenced by the number of walls between transmitter and receiver, in addition to the Euclidean distance:

$$L_{\mathrm{waf}}(\mathbf{p}_i, \mathbf{p}_j) = 10n \log_{10} \frac{d(\mathbf{p}_i, \mathbf{p}_j)}{d_0} +$$
$$\begin{cases} w(\mathbf{p}_i, \mathbf{p}_j) \times \mathrm{WAF} & \text{if } w(\mathbf{p}_i, \mathbf{p}_j) < C \\ C \times \mathrm{WAF} & \text{otherwise}, \end{cases} \qquad (15)$$

where $n$ indicates the rate of change in path-loss (dBm); $d_0$ is a reference distance (m), $w(\mathbf{p}_i, \mathbf{p}_j)$ is the number of walls on a straight line between transmitter and receiver, $C$ is an empirical constant – i.e., the maximum number of walls that can make a difference in path-loss; and WAF is a constant factor specific to the type of each wall (dBm).

*Multi-Wall Model (MWM)* Another empirical model (Zvanovec et al, 2003) based on the following equation:

$$L_{\mathrm{mwm}}(\mathbf{p}_i, \mathbf{p}_j) = L_{\mathrm{FSL}}(\mathbf{p}_i, \mathbf{p}_j) + \sum_{l=1}^{N} k_l w_l(\mathbf{p}_i, \mathbf{p}_j) + k_f \mathfrak{f}(\mathbf{p}_i, \mathbf{p}_j) , \qquad (16)$$

where $L_{\mathrm{FSL}}(\mathbf{p}_i, \mathbf{p}_j) = L_0 + 10n \log(d(\mathbf{p}_i, \mathbf{p}_j))$, models a free space path-loss model (dBm), with $L_0$ being the path loss at a reference distance (dBm); $w_l()$ is the number of walls of $l^{\mathrm{th}}$ type between transmitter and receiver, $k_l$ is a parameter for the attenuation affecting the signal for a wall of type $l$ (dBm); $\mathfrak{f}(\mathbf{p}_i, \mathbf{p}_j)$ is the number of floors between transmitter and receiver, and $k_f$ is the attenuation parameter observed by signal due to the type of that floor (dBm).

*ITU Radio Communication Model (ITU)* An empirical model, used by the IEEE 802.15 Working Group for Wireless Personal Area Networks (Hernandez et al, 2012), for testing the proposed channel model of a signal propagating in an arbitrary environment:

$$L_{\mathrm{itu}}(\mathbf{p}_i, \mathbf{p}_j) = 20 \log_{10} f + n \log_{10} d(\mathbf{p}_i, \mathbf{p}_j) + k_f \mathfrak{f}((\mathbf{p}_i, \mathbf{p}_j)) - 28 , \qquad (17)$$

where $n$ is the distance power loss coefficient (dBm); $f$ is the communication frequency (MHz); $d()$ is the distance between two locations (in m); $k_f$ is the floor penetration loss factor (dBm); $\mathfrak{f}()$ is the number of floors between two locations.

Each parameter of the above models should be fine-tuned according to the specific environment. Values for parameters are heuristically suggested in the related papers, usually for a communication signal at 2.4 GHz (WiFi) for different scenarios, including indoor office environments. In this paper, we start from those and fine tune them according to the characteristics of the hardware on the Turtlebot 2 robots used for experiments.

Considering the transmitting power $T_{\text{power}}$ (dBm), the RSSI between transmitter and receiver can be then calculated as:

$$\text{RSSI}(\mathbf{p}_i, \mathbf{p}_j) = T_{\text{power}} - L_{()}(\mathbf{p}_i, \mathbf{p}_j). \tag{18}$$

As a physical map of the environment is available in our setting as a grid map, a communication map can be computed as a prior for every location reachable by a receiver robot, given a fixed location for a transmitter robot. Note that, as some locations are inaccessible to the robots — e.g., because of doors — and grid maps are pre-built by the robots, the number of walls is an estimate of the actual number of walls. Specifically, every change from free space cell to occupied cell along the line segment connecting $\mathbf{p}_i$ and $\mathbf{p}_j$ in the grid is counted as one wall. Grid maps are preprocessed in such a way that small objects are removed from them and thus not counted as walls. Generating a communication map as prior using the above models shows that locations closer to the fixed robot observe higher RSSI values, while distant locations have lower values. While this trend seems to be common, the RSSI values from these priors are different; for instance, WAF model seems to return values that are smaller than those of other models.

The accuracy of the communication models is empirically evaluated. In particular, the error is calculated as the difference between the measurements actually collected by two Turtlebot 2 robots, described in Section 5.3, and the corresponding RSSI values estimated by the communication models. We conducted 6 different preliminary experiments in the Engineering building of the University of South Carolina[4]. Each experiment involved one robot fixed at an arbitrary location, and the other robot following a precomputed path. Each robot measures the WiFi signal strength at 10 Hz along with its position in the map. The physical environment used for these experiments is depicted in Figure 12(a).

Table 1 shows mean (and standard deviation) error calculated for the 6 different experiments. An accuracy error of $<20$ dBm is comparable to what is shown,

for example in (Bahl and Padmanabhan, 2000). It is worth mentioning that we observed a change of 8 dBm to 10 dBm in the RSSI value while changing the height of the antenna (by a few centimeters) on the robot. As such models display a relatively low error, the use of such communication models as prior, shown in the next section, is justified.

Also note that we included (in the last row of Table 1) our GP model built on the measurements taken and validated its modeling power for the signal strength finding that its error is the lowest among the different models.

### 4.3.2 Use of Communication Models as Prior

First of all, the communication models described in the previous section can be directly integrated in the GP as mean function to possibly improve the quality of the predictions, making the prediction process slightly more computationally expensive than using a zero-mean function. In the experimental section, we discuss the impact of varying the mean function on the overall performance.

Second, such priors can be used to guide the robots on where to take measurements. In particular, for a given location of a leader (either for the PM or for the RM algorithm), we design an algorithm to generate a set of locations that can be provided as goals to the followers and to filter measurements in the GP training dataset. The main idea is to choose locations that are highly informative, namely those which present some change in the field, as constant values can be easily approximated by the underlying assumption of the GPs, namely smoothness of the modeled phenomenon.

Mathematically speaking, the RSSI slope is estimated at each location by the first order derivative of the communication map built as prior from the communication models. The change of slope between neighbor locations in the map can be calculated by the second order derivative. If the slope change between two locations crosses a threshold $(\tau)$, that location is selected as one of the possible goals of the moving robot (follower). This idea is explained in detail in Algorithm 3. Specifically, the inputs are the chosen communication model comm_model (see Section 4.3.1), the physical map $\mathcal{A}$ of the environment to be used for creating a communication map as prior, and a specific location $(x_1, y_1)$ in the environment for the leader. Lines 1-3 create a communication map as prior according to one of the communication models and the environment $\mathcal{A}$ for all possible locations where the robots can go. According to such a communication map, specific goals are calculated in Line 4. In particular, the first and second derivatives

---

[4] All experiments were conducted at night time, so the interference due to moving objects/humans is minimal, except for the people performing the experiment.

Table 1: Errors (dBm) between the calculated RSSI values with the different models and the actual robot measurements, for 6 experiments performed varying locations of a robot acting as a transmitter; while another moving robot follows a fixed path and collects data. The values are comparable to what is shown, for example, in (Bahl and Padmanabhan, 2000).

| Path-loss Model | Experiment 1 | | Experiment 2 | | Experiment 3 | | Experiment 4 | | Experiment 5 | | Experiment 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mean | stdev | mean | stdev | mean | stdev | mean | stdev | mean | stdev | mean | stdev |
| Distance | 8.09 | 4.98 | 9.40 | 7.27 | 9.40 | 7.28 | 17.56 | 9.27 | 10.89 | 8.54 | 6.14 | 5.62 |
| WAF | 12.02 | 10.24 | 15.33 | 12.08 | 15.34 | 12.09 | 27.29 | 10.87 | 20.06 | 11.85 | 17.47 | 7.04 |
| MWM | 7.98 | 5.01 | 9.37 | 7.26 | 9.38 | 7.27 | 17.66 | 9.29 | 11.15 | 8.58 | 4.03 | 3.66 |
| ITU | 11.33 | 7.84 | 15.49 | 10.28 | 17.79 | 7.09 | 29.02 | 11.74 | 20.51 | 9.66 | 15.40 | 6.43 |
| GP | 2.92 | 2.43 | 3.14 | 2.48 | 3.26 | 2.38 | 2.90 | 2.30 | 3.18 | 2.30 | 2.91 | 2.09 |

of such a communication map are computed in Lines 8-10 and Lines 11-13, respectively. The locations that display a second derivative high value are selected as candidate locations where to go (Lines 14-17).

---

**Algorithm 3** Goals for Moving Robot to Pick Observations

---

**Input:** $\mathcal{A}$ (physical map of the environment), $\mathbf{p}_i$ (possible locations that robots can occupy), comm_model, $(x_1, y_1) \in \mathcal{A}$ (considered location of leader), $\tau$ (threshold)
**Output:** List of candidate goals for the moving robot $\{(x_i^d, y_i^d)\}, \forall i \in goals$
1: **for** $(x_2, y_2) \in \mathbf{p}_i - (x_1, y_1)$ **do**
2: $\quad rssi_{(x_1, y_1)}^{(x_2, y_2)} \leftarrow calculate\_rssi(\mathcal{A}, x_1, y_1, x_2, y_2, \text{comm\_model})$; $\triangleright$ Using appropriate Equation from Equations (14), (15), (16), (17)
3: **end for**
4: $goals \leftarrow get\_goals(rssi, x_1, y_1, \mathbf{p}_i)$;
5: **return** $goals$
6: **procedure** GET_GOALS$(rssi, x_1, y_1, \mathbf{p}_i)$
7: $\quad goals \leftarrow \{\}$ $\triangleright$ Return variable
8: $\quad$ **for** $(x_2, y_2) \in \mathbf{p}_i$ **do** $\triangleright$ Calculate the slope of RSSI
9: $\quad\quad rssi\_slope_{(x_1, y_1)}^{(x_i, y_i)} \leftarrow calculate\_slope(x_1, y_1, x_2, y_2, rssi)$
10: $\quad$ **end for**
11: $\quad$ **for** $(x_2, y_2) \in \mathbf{p}_i$ **do** $\triangleright$ Calculate the slope of rate of change of RSSI
12: $\quad\quad rssi\_change\_slope_{(x_1, y_1)}^{(x_i, y_i)} \leftarrow calculate\_slope(x_1, y_1, x_2, y_2, rssi\_slope)$
13: $\quad$ **end for**
14: $\quad$ **for** $(x_2, y_2) \in \mathbf{p}_i$ **do**
15: $\quad\quad$ **if** $rssi\_change\_slope_{(x_1, y_1)}^{(x_i, y_i)} > \tau$ **then**
16: $\quad\quad\quad goals.add((x_i, y_i))$
17: $\quad\quad$ **end if**
18: $\quad$ **end for**
19: $\quad$ **return** $goals$
20: **end procedure**

---

Additionally, locations generated from the algorithm can be used to filter signal strength measurements polled by the robots. Remind that the polling frequency between two moving robots can be set arbitrarily. In practice, high frequencies allow robots to collect large datasets; at the same time, it would require large computational efforts for the GP parameter estimation. Note that maximizing Equation (4) takes $O(n^3)$, where $n$ denotes the number of samples: as a consequence, their number should be limited to a set of few but significant ones. If measurements are taken close to the locations found by Algorithm 3 within a given range, they are used
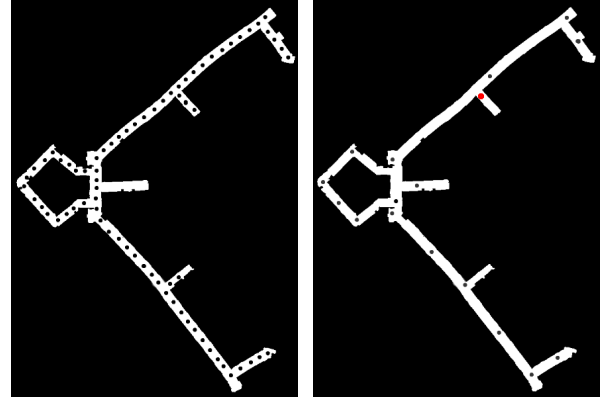


(a) Candidate locations uniformly spaced  (b) Candidate locations filtered

Fig. 5: Candidate locations uniformly spaced in the environment (a) without the use of a communication model and candidate locations (black) considered for followers for a given location of the leader in red (b).

for the GP update. It is also important to note that $\tau$ needs to be generated heuristically. When $\tau$ is high, the algorithm might not find a sufficient number of candidate locations. As such, the communication map cannot be built accurately. Lower $\tau$ could result in too many goals for the robots, making the GP computation process too expensive. From our preliminary experiments, $\tau = 11$ ($\pm 10\%$; measure unit dBm/m$^2$) is a reasonable value for all the four models. This provided, for example, in the environment of the Engineering building at the University of South Carolina between 30-120 goals for various communication models, among 212 possible locations that were uniformly sampled in the tested environments (see Figure 5). Note that, to account for inaccuracies of the prior model, locations within a given radius of those generated by Algorithm 3 are added to the list of candidate locations.

The proposed method to select locations is integrated in the PM and RM strategies by filtering the

locations considered for the follower. Then, PM and RM algorithms choose the set of goals according to the strategies described in Section 4.2.

Using the data collected during our experiments described in the previous section — i.e., one robot is fixed at a location, while the other one moves along a fixed, known path — we evaluate the GP built with all measurements and the GP built only with filtered measurements by the proposed method. In general, the GP with filtered data maintains low variance in GP predictions and low Root Mean Square Error (RMSE). At the same time, the GP training time reduces by 50%, compared to the GP with all data. For example, the RMSE between the observed data and the predicted values by the GPs, in one of the experiments, is 11.01 for the GP with all data and 11.03 for the GP with filtered data. These results validate the use of such a filtering approach. Further results are presented in the next section.

## 5 Experimental Evaluations

In this section, we evaluate our online sensing strategies discussing their performance obtained both in simulation and with real robots.

### 5.1 Experimental Setup

To ensure repeatability under controlled conditions, the system has been first evaluated with a realistic 2D simulator, Stage (Vaughan, 2008). We also deployed our strategies on a team of real Turtlebot 2 robots (Figure 1) to assess their validity under real-world conditions. Robots' control stacks have been implemented in ROS (Quigley et al, 2009), while tasks based on Gaussian Processes exploit the GPy framework (GPy, since 2012).

In our experiments we considered teams of 2, 4, and (limited to simulations) 6 robots for which we perform comparisons among the PM and RM strategies introduced in Section 4.2 and a baseline strategy RAND. The latter is an uninformed strategy where each robot moves randomly while polling teammates when they occasionally fall within its range. As a consequence, robots following the RAND strategy do not require to maintain a GP for selecting goal locations. The consequent time savings results in potentially collecting amounts of data samples much larger than with the PM or RM strategies. Thus, to ensure a fair comparison and focus on the quality of the collected data, we set polling periods seeking a computational effort that is balanced (ensuring that all the strategies roughly collect the same

amount of data samples) and affordable in real-time (preventing situations where the time needed for training GPs dominates the time devoted to all the other operations). From a preliminary empirical evaluation we draw the following values as suitable ones: for 2 robots, 3 s for PM and RM, 3.5 s for RAND; for 4 robots, 5 s for PM and RM, 10 s for RAND; for 6 robots, 10 s for PM and RM, 18 s for RAND. (Notice how the polling period of RAND is set to a higher value compared to PM and RM due to the extra temporal costs these last ones incur for training the GPs.) In addition, for the PM and RM strategies, in determining backup locations (see Section 4.2) we assume that two robots can always communicate if within a distance of $R_c/3$, or $R_c/2$ if in line-of-sight.

In summary, an experimental configuration is defined along the following dimensions:

- the environment where robots are deployed;
- the number of robots: in simulations $\{2, 4, 6\}$ while with real robots $\{2, 4\}$;
- the sensing strategy: RAND, PM, or RM (with 1 or 2 leaders);
- whether the method for filtering measurements based on the communication models is enabled (Section 4.3.2);
- the GP mean function: zero-mean, DIST (Equation (14)), or WAF (Equation (15)).

The metrics used to evaluate the strategies consider the quality of the GPs that would be obtained by merging all the collected data in a rendez-vous of the whole team every 5 min. In particular, one measure of the quality is given by the Rooted Mean Squared Error (RMSE) at time $t$ on a given test set (10,000 measurements collected randomly for simulations; measurements collected following some predefined trajectories as described in Section 4.3.1 for real robots):

$$\text{RMSE}_t = \sqrt{\frac{\sum_{\mathbf{x}^i \in \mathcal{A}^2}(\hat{f}^t(\mathbf{x}^i) - f(\mathbf{x}^i))^2}{|\mathcal{A}^2|}}. \qquad (19)$$

In addition, the average *predictive standard deviation* — namely, the squared root of the predictive variance — of the predictions is considered. The latter metric is fundamental for an online scenario, as ground truth is typically not available. The cost of constructing a communication map is assessed in terms of traveled distance and computing time to train the GP.

### 5.2 Simulations

In the simulation experiments, the communication channel is simulated using the signal propagation model proposed by Bahl and Padmanabhan (2000) — i.e.,
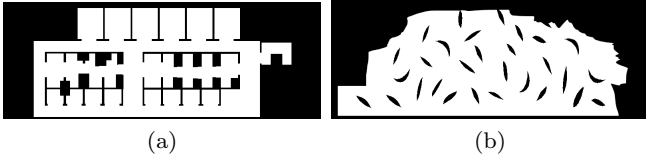
Fig. 6: Simulation environments (Office, left, and Cluttered, right), approximate size 80 m × 30 m.

Equation (15) and Equation (18). Such a communication model has been used in other simulators, such as MRESim (Spirin et al, 2014) and USARSim (Carpin et al, 2006), to test robot exploration strategies. In addition, USARSim has been used in the RoboCup Rescue Simulation Virtual Robot Competition in search and rescue scenarios. Note that in our real robot experiments, considering environments where only robots are present, the communication models were able to capture the general trend of the signal strength. The parameters are empirically chosen to achieve distance and signal strength comparable to those observable with a Turtlebot 2 robot. In particular, $T_{\text{power}} = -38$ dBm, $d_0 = 1$ m, $n = 2.3$, WAF $= 3.37$, $C = 5$. A bidirectional communication link is available between any two simulated robots if $\text{RSSI}(\mathbf{p}_i, \mathbf{p}_j) \geq -93$ dBm in both directions. The indicative communication range $R_c$ is set to 50 m. A white noise $\epsilon \sim \mathcal{N}(0,1)$ has been added to the signal to obtain more realistic measurements.

Given that WAF communication model is used, we adopt it also as the prior of our filtering method. The number of pairs of locations obtained is around 150 over around 2000 pairs for both environments. Furthermore, we show results by using a GP with zero mean, a mean function that uses DIST (Equation (14)), and one that uses WAF (Equation (15)).

We select two representative environments of realistic size depicted in Figure 6. "Office" is a portion of the "sdr_site_b" environment from the Radish repository (Howard and Roy, 2003), representing an indoor environment with corridors and rooms. "Cluttered" is inspired from the "grass" environment of the MRESim repository (Spirin et al, 2014). It is unstructured with many obstacles around.

We denote with RM-N the RM strategy where each group is composed of N robots and we consider the presence of at most 2 leaders. The following parameters are chosen from preliminary experiments: $n_s^{\text{PM}} = 10000$, $n_s^{\text{RM}} = 100$ for all the team sizes, $d_{\text{min}} = 25, 20, 15$ m for 2, 4, and 6 robots, respectively. The $n_w$ parameter is chosen to allow a fair coverage of a region, compatibly with a mission duration set to 30 minutes and accounting for the number of robots. In particular, $n_w = 3|R_f|, 2|R_f|, 1|R_f|$

for 2, 4, and 6 robots, respectively. Five runs are executed for each experimental configuration.

Figure 7 reports the results obtained in the Office environment, using a zero mean GP. Looking at the prediction performance for 2 robots (Figure 7(a)-(b)), it can be observed that all the strategies are able to significantly lower down both the RMSE and the predictive standard deviation. PM performs slightly better than RM. In addition, PM has a comparable performance with RAND in terms of RMSE and predictive standard deviation – there is no statistical significant difference between the two. This means that, when the state space of the spatial phenomenon to learn is large and the number of available robots is low, the use of a complex and informed strategy to select data samples might not introduce significant improvements in the map's quality.

On the other side, benefits can be observed with respect to efficiency. The traveled distance for the proposed strategies (Figure 7(d)) is consistently lower than the one from RAND. The computation time (Figure 7(c)) is comparable for the proposed strategies without any filtering and RAND, while it is significantly lower when the filtering technique is enabled (PM-waf and RM-2-waf). This means that our sensing strategies move robots around in an efficient way.

Figure 7(e)-(h) and Figure 7(i)-(l) show results for 4 and 6 robots, respectively. In these two settings, PM obtains better results compared to RAND, with RM still not performing as well as PM overall. For the RMSE, the advantage offered by PM is slight, but often statistically significant (e.g., $p$-value=0.004064 in one-way ANOVA at 30 min between PM and RAND for 4 robots). The predictive standard deviation decreases using PM around 15 minutes, which is consistently maintained until the end of the mission. Such improvement is statistically significant (e.g., $p$-value=0.00005615 at 30 min between PM and RAND for 6 robots). The results obtained by RM suggest that it could not be convenient to spend too much time on lowering down the uncertainty of a single region. However, we argue that, by setting the $n_w$ parameter equal to 1, we could obtain a performance similar to that of PM even for 2 and 4 robots. The relatively comparable performance of RAND in terms of map quality (RMSE and predictive standard deviation) comes from the fact that this strategy uniformly samples the spatial phenomenon (recall that robots keep polling each other). However, also in such scenarios, RAND has a remarkable downside in the distance it requires the robots to travel, making it a very inefficient sampling strategy.

While the quality of the communication map is slightly worse when the proposed methods run the filtering method
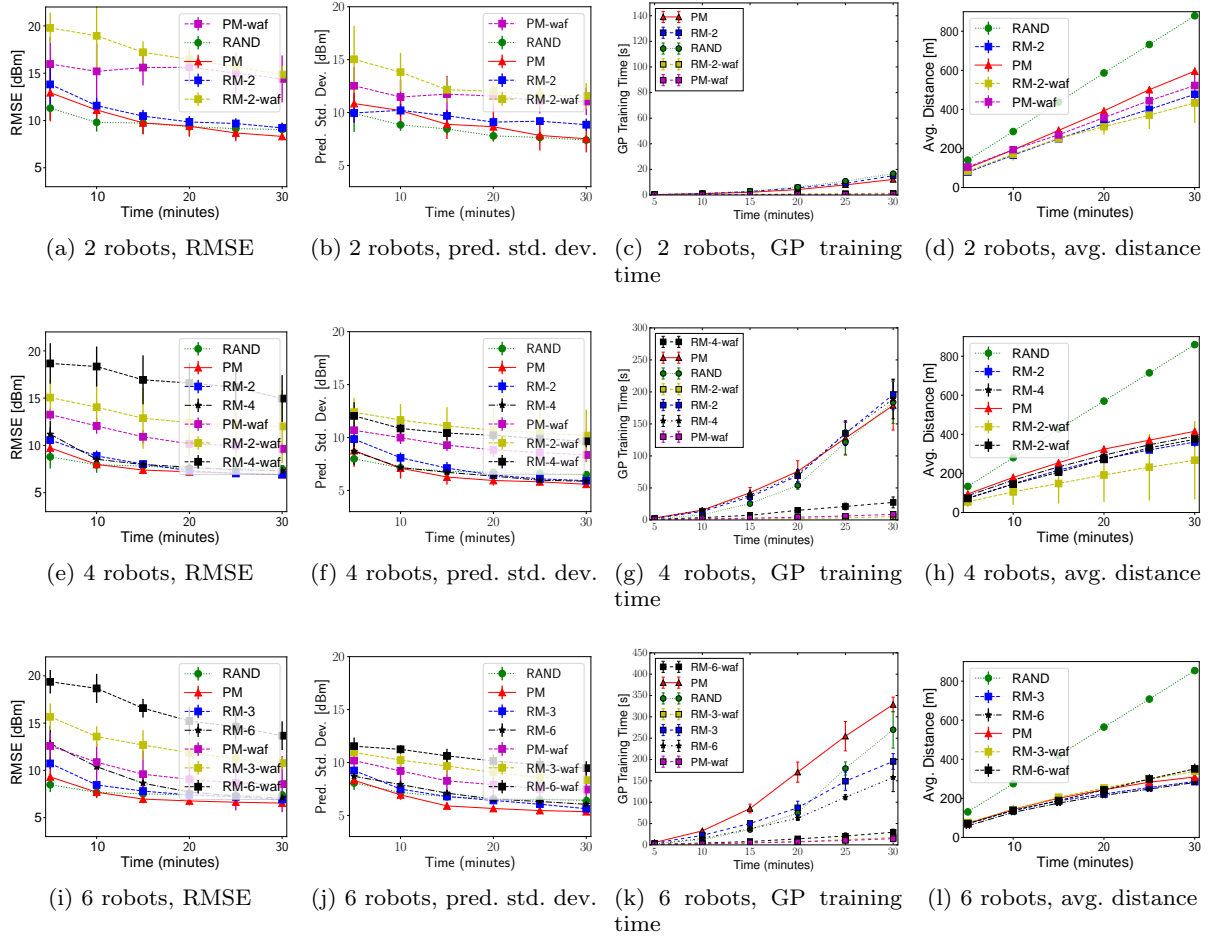
Fig. 7: Simulation experiments, Office environment, zero-mean GP.

of Section 4.3.2, the computation time and the traveled distance are lower, making use of filtering preferable when computing power is limited and time is a hard constraint.

Figure 8 shows results for the proposed strategies with the filtering method, where the threshold value is varied to obtain different numbers of locations. The number of locations obtained are $\sim 1600, \sim 600, \sim 300, \sim 100$, for no filtering, $\tau = 0$, $\tau = 5$, $\tau = 11$, respectively. As expected, the less aggressive on filtering, the higher the quality of the communication map in terms of RMSE. At the same time, the GP training time increases considerably, resulting into a training time close to that of the proposed strategies without any filtering. The samples used to build the GP models without filtering is around the 90% of the samples used with RAND, while the samples used to build the GP models with filtering is around the 50% of the samples used with RAND.
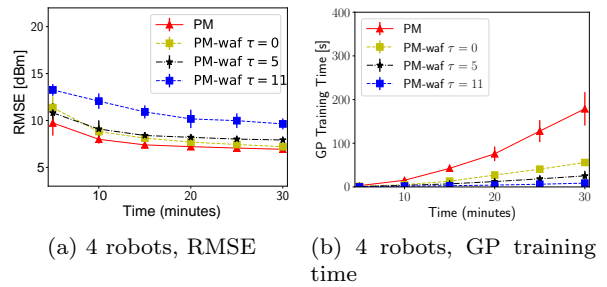


Fig. 8: RMSE and GP training time for 4 robots in the Office environment, varying the threshold value $\tau$ in Algorithm 3 to get different number of candidate locations.

By using different GP mean functions, the benefit is visible in the quality of the communication map for all the methods (Figure 9 shows results with 6 robots). In particular, also our proposed strategies using filtering method obtain comparable RMSE and predictive

standard deviation. There is a slight increase in the GP training time because of the extra term in the GP calculations for the mean function.

The results for the Cluttered environment are very similar to those of Office, and all the above considerations still hold — see Figure 10 for results with 6 robots.

Finally, Figure 11 shows two instances of communication maps built for a source located at the center of the simulation environments for 6 robots using the PM strategy. Intuitively, the communication maps capture well how the signal strength changes over the environment: it is maximum close to the transmitter robot (at the center) and then it decreases with distance, keeping higher in locations in line-of-sight with the transmitter robot (for example, see the ridge in the communication map in Figure 11(a)).

5.3 Experiments with Real Robots

We also deployed and tested our algorithms on a team of two and four TurtleBot 2 robots, in the Swearingen Engineering Center at the University of South Carolina. Each platform is equipped with an RGB-d camera (Microsoft Kinect) and an on-board laptop with an integrated WiFi card. The maps used for localization are built in a setup phase, where a single robot is manually driven around the environment to collect readings, processed by the ROS gmapping package (Grisetti et al, 2007).

Two of the environments used are depicted in Figure 12. They have different characteristics: the first one (shown in Figure 12(a)) is characterized by long corridors with some intersecting short corridors and one small loop. Note that between the two long corridors, there is an outdoor space, which the robots cannot access. The other environment (in Figure 12(b)) has instead corridors surrounding small office rooms.

Starting with the parameters for the algorithms from the simulation, we performed first some initial experiments to fine tune them to account for real world challenges. For example, the timeout $t_d$ needs to be increased due to locomotion noise which, in the real world, is clearly non-negligible.

Some preliminary experiments showed that the MWM communication model provides a good number of candidate locations and has good accuracy (see Section 4.3.1) for the environments considered. As such MWM is used to generate candidate locations using Algorithm 3. PM, RM, and their variants with MWM as communication prior are compared against RAND.

For each of the two physical environments of Figure 12, two and four Turtlebot 2 robots are used to verify the performance of both the proposed and the baseline strategies, for a 20 min duration.

The trends of the quantitative results are slightly different to those obtained in simulation as the robots travel comparable distances with the different strategies – Figures 13 and 14 show results for two robots on the map shown in Figure 12(a). This can be explained by the fact that, differently from simulations, the environment has narrower corridors which has two implications: first, the corridor structure of the environment guides the robots; second, such narrow corridors, together with motion noise, increases the probability of collisions. The values of uncertainty and RMSE are slightly higher compared to the simulation and they show also some increase during the mission: besides the higher complexity of the signal, the main reason lies on the fact that the robots travel much less distance (e.g., 200 m with real robots vs. 800 m in simulation) and as such they collect a lower number of samples. However, the GP model represents well the measurements with a RMSE and a predictive standard deviation comparable to those of simulation.

Figure 15 shows the results for 4 robots in the Lab-Corridors environment. The results for the Office-Corridors environment of Figure 12(b) are very similar to those of Lab-Corridors, and all the above considerations still hold.

Using communication models as mean function for the GP does not provide as much benefit as in simulation: the RMSE is slightly higher when using DIST and WAF as mean functions, compared to a zero-mean function. However, GP models with DIST and WAF mean functions seem to provide a slightly more reliable prediction, as shown by the predictive standard deviation (see Figures 13(d), 13(e), and 13(f)).

The communication map that is built by GP looks consistent with the obstacles; e.g., Figures 16 and 17 show a 2D communication map, by fixing one location, in the two maps. Since we experience some noise in localization, the associated WiFi strength measurements turned out being not accurate in a limited number of times. Further, although our approach computes the utility function not only on the basis of the selected destination locations, but also of the paths followed by the robots, few times the robots interfered with the motion of others, especially using RAND and the PM strategy, which have a lower coordination level compared to RM. Including the communication model in the GP mean function reduces the absolute value in terms of standard deviation, at the cost of slightly increasing the training time.
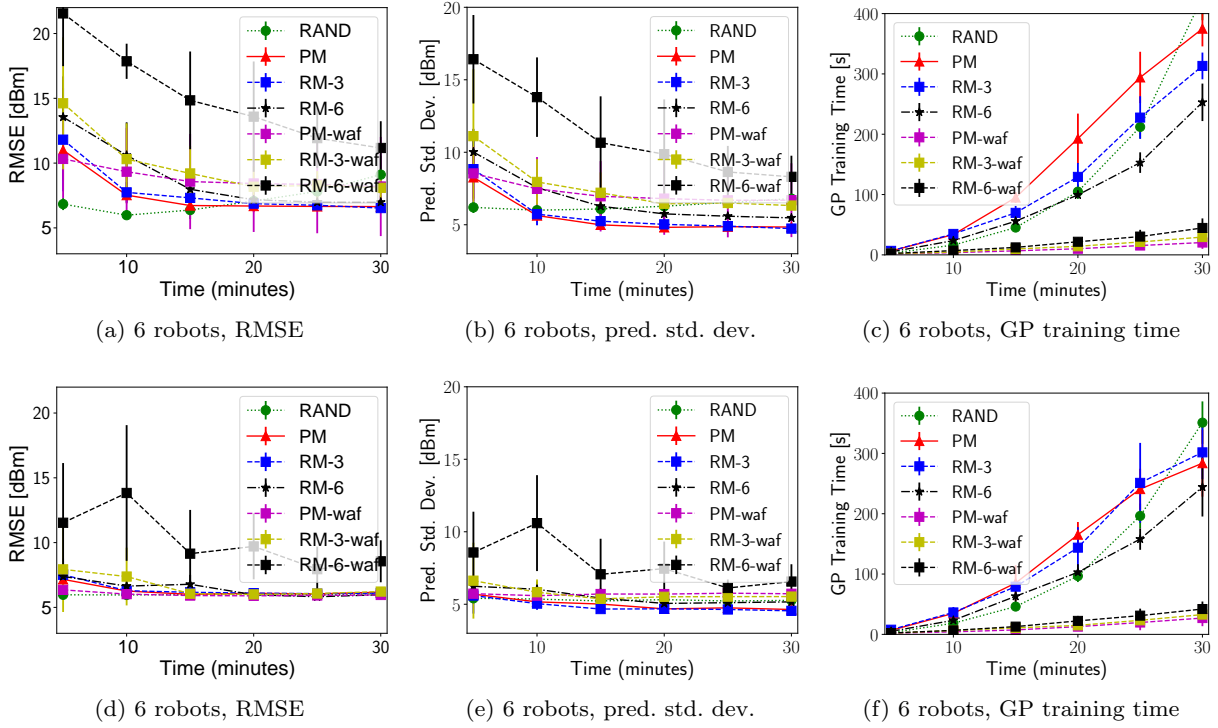
(a) 6 robots, RMSE      (b) 6 robots, pred. std. dev.      (c) 6 robots, GP training time

(d) 6 robots, RMSE      (e) 6 robots, pred. std. dev.      (f) 6 robots, GP training time

Fig. 9: Simulation experiments, Office environment, first row: GP with DIST mean function, second row: GP with WAF mean function.
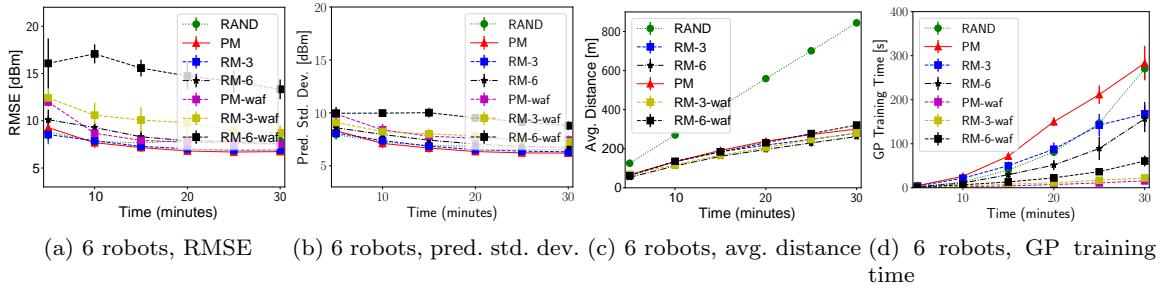


(a) 6 robots, RMSE    (b) 6 robots, pred. std. dev. (c) 6 robots, avg. distance    (d) 6 robots, GP training time

Fig. 10: Simulation experiments, Cluttered environment, zero-mean GP, 6 robots.

## 5.4 Discussion

Some general insights are highlighted here from the results obtained by our experimental activity, as well as some lessons learned that are starting points for our future work.

In general, when increasing the number of robots and when the main constraint is determined by the traveled distance, our coordinated sensing strategies are beneficial to avoid any overlap in measurements. Furthermore, if computational resources are very limited, including filtering methods makes the proposed system real-time. Using a communication model as a mean function of the GP provides benefit in terms of model-

ing, however slightly increasing the computational demand. In the real world, while the general highlights still hold, the communication channel is much more complex to be modeled, especially considering dynamicity. Moreover, the noise in locomotion and sensing needs to be taken into account also during the planning phase to avoid robots to be stuck and prevent wrong measurement associations. However, also in real world, our proposed sensing strategies provide good results.
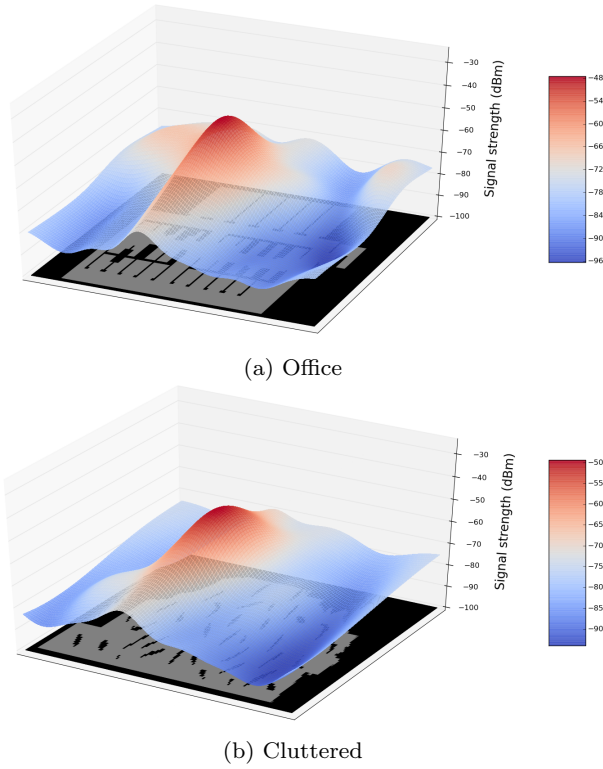
(a) Office



(b) Cluttered

Fig. 11: Maps for a source located in the center of the simulation environments (6 robots, PM strategy).



(a) Lab-Corridors
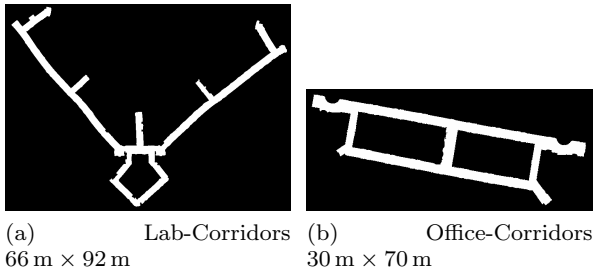66 m × 92 m

(b) Office-Corridors
30 m × 70 m

Fig. 12: Two environments, portions of third floor at Swearingen Engineering Center, University of South Carolina.

## 6 Conclusion

In this work, utilizing a Gaussian Process representation of the WiFi signal strength distribution, we designed and tested multi-robot online sensing strategies for mapping the quality of WiFi communication links between pairs of locations in a known environment. Such sensing strategies are fit for homogeneous or heterogeneous teams of robots. In addition, we introduced the use of communication models as prior to improve overall performance and computational efficiency. Experiments in simulation and with real robots show how our distributed coordination strategies can effectively

and efficiently perform communication map building task.

Future work will extend the approach to explicitly consider temporal variations and will employ more complex communication models that account for multiple paths, e.g., (Lindhé and Johansson, 2013). In addition, a method that allows long-term monitoring of WiFi signal strength is in our plans. Furthermore, it will be interesting to have dynamic teams and more synchronous coordination among different groups of robots, also in order to plan for sequences of pairwise measurements. The task of building communication maps will be integrated together with other missions robots might have, such as exploration (e.g., see the preliminary multi-robot system for exploration reported in (Amigoni et al, 2018), which uses communication maps built with an approach similar to that presented in this paper) or environmental monitoring (Manjanna et al, 2017). Extrapolating this work to outdoor, under-water, and aerial environments opens several interesting research directions.

## References

Amigoni F, Basilico N, Quattrini Li A (2013) How much worth is coordination of mobile robots for exploration in search and rescue? In: Chen X, Stone P, Sucar LE, Zant T (eds) RoboCup 2012: Robot Soccer World Cup XVI, no. 7500 in Lecture Notes in Computer Science, Springer, p 106–117

Amigoni F, Banfi J, Basilico N (2017) Multirobot exploration of communication-restricted environments: A survey. IEEE Intell Syst 32(6):48–57

Amigoni F, Banfi J, Basilico N, Rekleitis I, Quattrini Li A (2018) Online update of communication maps for exploring multirobot systems under connectivity constraints. In: Proc. Distributed Autonomous Robotic Systems (DARS)

Bahl P, Padmanabhan VN (2000) RADAR: an in-building RF-based user location and tracking system. In: Proc. IEEE International Conference on Computer Communications (INFOCOM), vol 2, pp 775–784

Banfi J, Quattrini Li A, Basilico N, Rekleitis I, Amigoni F (2016) Asynchronous multirobot exploration under recurrent connectivity constraints. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp 5491–5498

Banfi J, Quattrini Li A, Basilico N, Rekleitis I, Amigoni F (2017) Multirobot online construction of communication maps. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp 2577–2583
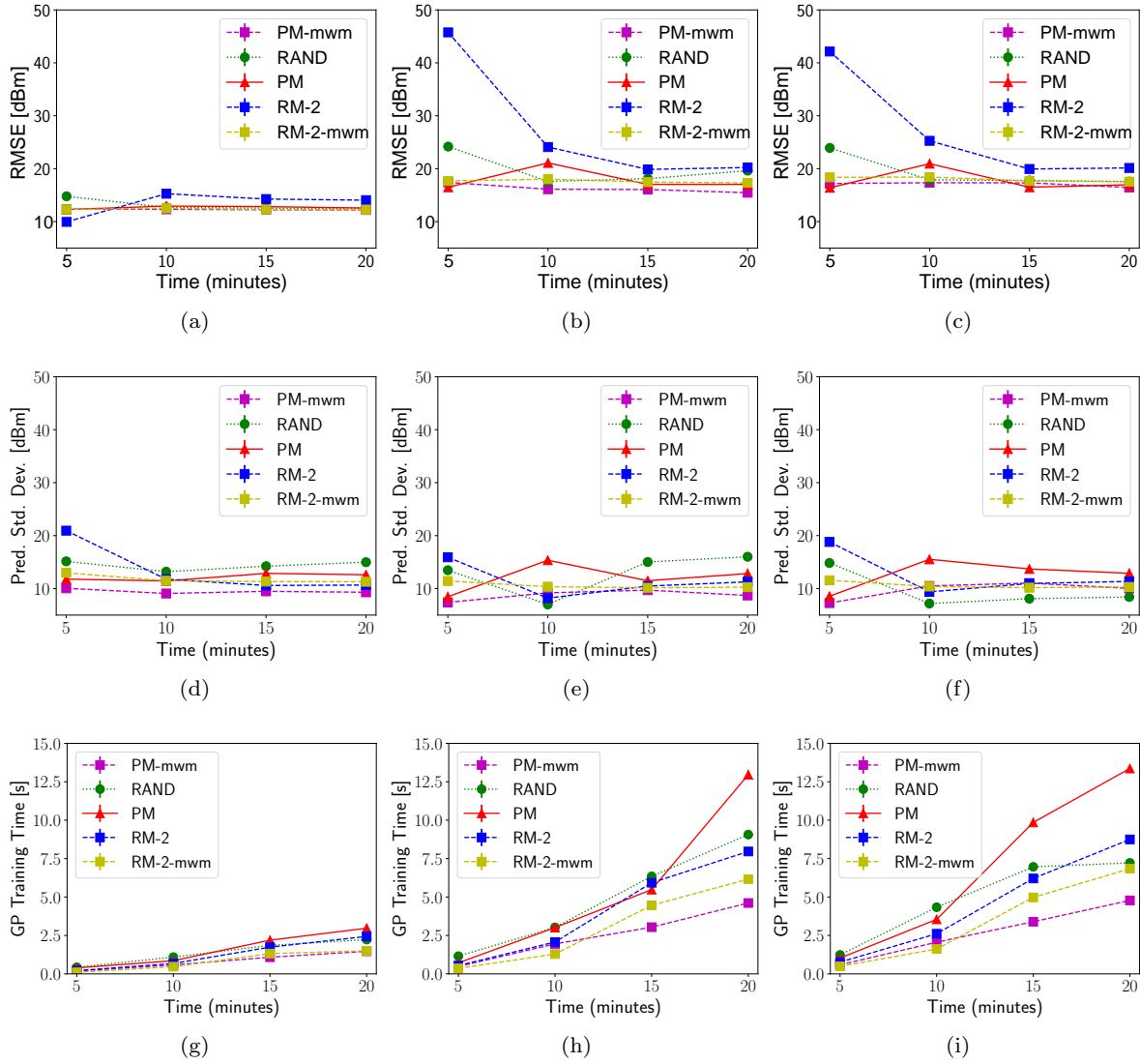
Fig. 13: Results with two Turtlebot 2 robots in Lab-Corridors, where the first column identifies the use of the GP with zero mean, the second one, the use of the GP with DIST mean function, the third one, with MWM mean function.
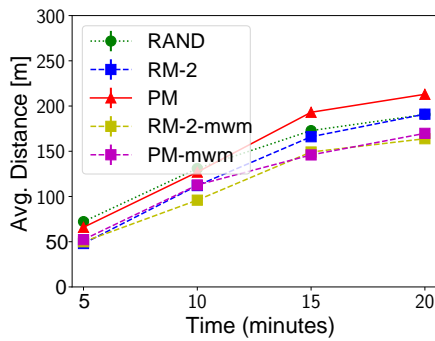


Fig. 14: Distance traveled with two Turtlebot 2 robots in Lab-Corridors.

Best G, Cliff O, Patten T, Mettu R, Fitch R (2018) Dec-MCTS: Decentralized planning for multi-robot active perception. The International Journal of Robotics Research To appear.

Carpin S, Stoyanov T, Nevatia Y (2006) Quantitative assessments of USARSim accuracy. In: Proc. of Per-MIS

Das J, Py F, Harvey JBJ, Ryan JP, Gellene A, Graham R, Caron DA, Rajan K, Sukhatme GS (2015) Data-driven robotic sampling for marine ecosystem monitoring. The International Journal of Robotics Research 34(12):1435–1452

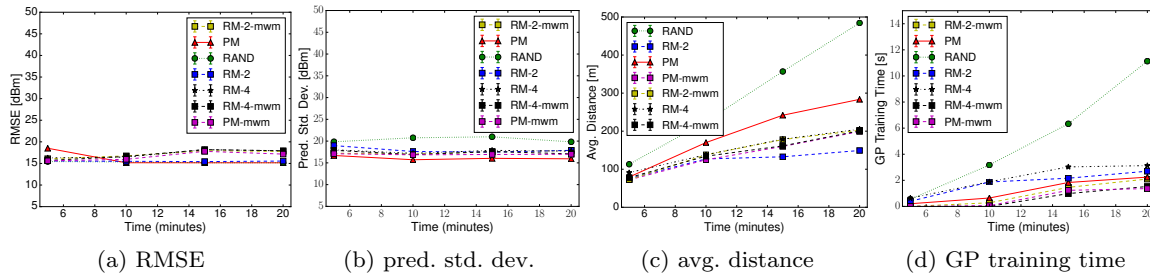Deisenroth M, Ng J (2015) Distributed gaussian processes. In: Proc. International Conference on Inter-

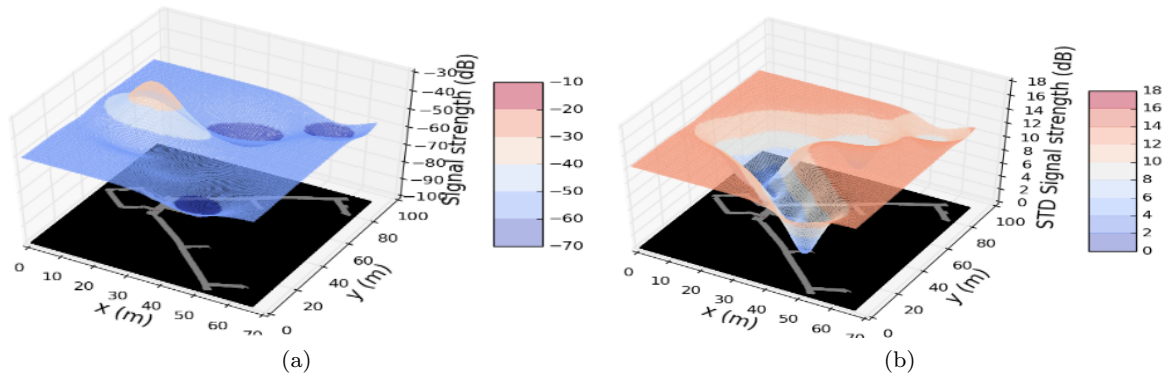Fig. 15: Lab-Corridors environment, zero-mean GP, 4 robots.



Fig. 16: An instance of a communication map (a) and the corresponding predictive standard deviation (b) built by two Turtlebot 2 robots in Lab-Corridors.

national Conference on Machine Learning ICML, pp 1481–1490

Dunbabin M, Marques L (2012) Robots for environmental monitoring: Significant advancements and applications. IEEE Robotics Automation Magazine 19(1):24–39

Ferris B, Fox D, Lawrence ND (2007) WiFi-SLAM using Gaussian Process Latent Variable Models. In: Proc. International Joint Conference on Artificial Intelligence (IJCAI), pp 2480–2485

Fink J, Kumar V (2010) Online methods for radio signal mapping with mobile robots. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp 1940–1945

Fink J, Ribeiro A, Kumar V (2013) Robust control of mobility and communications in autonomous robot teams. IEEE Access 1:290–309

Goldsmith A (2005) Wireless Communications. Cambridge University Press

GPy (since 2012) GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy

Gregory J, Fink JR, Stump E, Twigg JN, Rogers JG, Baran D, Fung N, Young S (2015) Application of multi-robot systems to disaster-relief scenarios with limited communication. In: Proc. International Conference on Field and Service Robotics (FSR), pp 639–653

Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Transactions on Robotics 23(1):34–46

Hernandez M, Li HB, Dotlić I, Miura R (2012) Reference channel models for proposals evaluation to TG8. In: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), pp 1–20

Heurtefeux K, Valois F (2012) Is RSSI a good choice for localization in wireless sensor network? In: Proc. IEEE International Conference on Advanced Information Networking and Applications, pp 732–739

Hollinger GA, Singh S (2012) Multirobot coordination with periodic connectivity: Theory and experiments. IEEE Transactions on Robotics 28(4):967–973

Howard A, Roy N (2003) The robotics data set repository (Radish). http://radish.sourceforge.net/

Hsieh MA, Cowley A, Kumar V, Taylor CJ (2008) Maintaining network connectivity and performance in robot teams: Research articles. Journal of Field Robotics 25(1-2):111–131

Im HJ, Lee CE, Cho YJ (2014) Radio mapping scheme using collective intelligent robots for teleoperation in
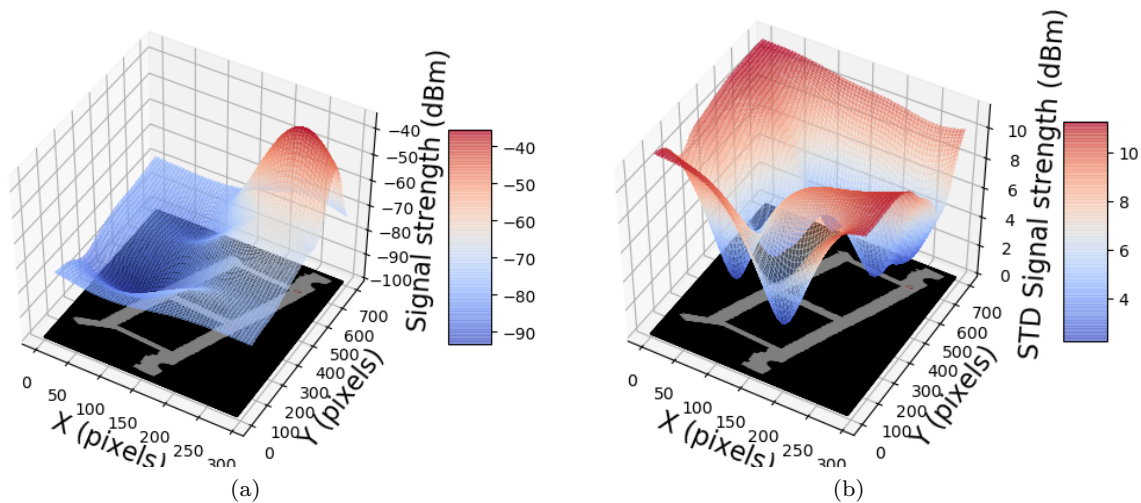
Fig. 17: An instance of a communication map (a) and the corresponding predictive standard deviation (b) built by two Turtlebot 2 robots in Office-Corridors.

unstructured environments. In: Proc. IEEE International Symposium on Robot and Human Interactive Communication, pp 856–861

Kemppainen A, Haverinen J, Vallivaara I, Röning J (2010) Near-optimal SLAM exploration in Gaussian processes. In: IEEE Conference on Multisensor Fusion and Integration, pp 7–13

Ladd AM, Bekris KE, Rudys A, Kavraki LE, Wallach DS (2005) Robotics-based location sensing using Wireless Ethernet. Wireless Networks 11(1-2):189–204

Lilienthal A, Zell A, Wandel M, Weimar U (2001) Sensing odour sources in indoor environments without a constant airflow by a mobile robot. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), vol 4, pp 4005–4010

Lindhé M, Johansson KH (2013) Exploiting multipath fading with a mobile robot. The International Journal of Robotics Research 32(12):1363–1380

Liu T, Cerpa AE (2011) Foresee (4C): Wireless link prediction using link features. In: Proc. Conference on Information Processing in Sensor Networks IPSN, pp 294–305

Manjanna S, Hansen J, Quattrini Li A, Rekleitis I, Dudek G (2017) Collaborative sampling using heterogeneous marine robots driven by visual cues. In: Proc. Conference on Computer and Robot Vision

Manjanna S, Quattrini Li A, Smith R, Rekleitis I, Dudek G (2018) Heterogeneous multi-robot system for exploration and strategic water sampling. In: Proc. IEEE International Conference on Robotics and Automation (ICRA)

Marchant R, Ramos F, Sanner S (2014) Sequential Bayesian optimisation for spatial-temporal monitoring. In: Proc. Conference on Uncertainty in Artificial Intelligence (UAI), pp 553–562

Mirowski P, Ho TK, Whiting P (2014) Building optimal radio-frequency signal maps. In: Proc. International Conference on Pattern Recognition, pp 978–983

Murphy RR, Tadokoro S, Kleiner A (2016) Disaster robotics. In: Springer Handbook of Robotics, pp 1577–1604

Ochoa S, Santos R (2015) Human-centric wireless sensor networks to improve information availability during urban search and rescue activities. Information Fusion 22:71–84

Penumarthi PK, Quattrini Li A, Banfi J, Basilico N, Amigoni F, O'Kane J, Rekleitis I, Nelakuditi S (2017) Multirobot exploration for building communication maps with prior from communication models. In: Proc. International Symposium on Multirobot Systems (MRS), pp 90–96

Quattrini Li A, Cipolleschi R, Giusto M, Amigoni F (2016) A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings. Autonomous Robots 40(4):581–597

Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Berger E, Wheeler R, Ng A (2009) ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software

Rappaport TS (1996) Wireless communications: principles and practice. Prentice Hall PTR New Jersey

Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. MIT Press

Rekleitis I, New A, Rankin E, Choset H (2008) Efficient boustrophedon multi-robot coverage: an algorithmic approach. Annals of Mathematics and Artificial Intelligence 52(2):109–142

Riva A, Banfi J, Fanton C, Basilico N, Amigoni F (2018) A journey among pairs of vertices: Computing robots' paths for performing joint measurements. In: Proc. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), pp 229–237

Scholl PM, Kohlbrecher S, Sachidananda V, Laerhoven KV (2012) Fast indoor radio-map building for RSSI-based localization systems. In: Proc. International Conference on Networked Sensing (INSS), pp 1–2

Singh A, Krause A, Guestrin C, Kaiser W (2009) Efficient informative sensing using multiple robots. Journal of Artificial Intelligence Research 34:707–755

Spirin V, de Hoog J, Visser A, Cameron S (2014) MRESim, a multi-robot exploration simulator for the rescue simulation league. In: Proc. RoboCup, pp 106–117

Tuna G, Gulez K, Gungor VC (2013) The effects of exploration strategies and communication models on the performance of cooperative exploration. Ad Hoc Networks 11(7):1931–1941

Twigg JN, Fink JR, Paul LY, Sadler BM (2012) RSS gradient-assisted frontier exploration and radio source localization. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp 889–895

Vaughan R (2008) Massively multiple robot simulations in stage. Swarm Intelligence 2(2-4):189–208

Yamauchi B (1998) Frontier-based exploration using multiple robots. In: Proc. International Conference on Autonomous Agents, pp 47–53

Zlot R, Stentz A, Dias MB, Thayer S (2002) Multi-robot exploration controlled by a market economy. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp 3016–3023

Zvanovec S, Pechac P, Klepal M (2003) Wireless LAN networks design: Site survey or propagation modeling? Radio Engineering 12(4):42–49