Ph.D. Thesis Proposal

Distributed Exploration with Robotic Networks: Queue Stability and Collaborative Localization

Lillian Clark
Ming Hsieh Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA 90089
lilliamc@usc.edu

Committee Members
Prof. Bhaskar Krishnamachari (Chair)
Prof. Konstantinos Psounis
Prof. Michael Neely
Prof. Gaurav Sukhatme
Dr. Joseph Galante

Qualification Exam Date: May 7th, 2021, 9:15-11:00am

Thesis statement: Distributed exploration algorithms for robotic networks (1) should explicitly optimize for low-complexity anchorless localization performance and mapping accuracy and (2) can allow intermittent connectivity in order to improve exploration efficiency without sacrificing timely data transfer.

1 Introduction

Robotic exploration is an integral component of future space missions, with applications including characterizing lunar regolith disturbance on the surface [1, 2] and mapping subsurface tunnels formed by cooled flowing lava which may be favorable for human activities [3, 4]. Networked multi-robot systems are well-suited for these harsh, remote environments because of the low volume, mass, and development costs of small robots and the inherent redundancy of teams of robots [5, 6, 7]. Research on robotic networks is relevant for many other applications where spatially distributed measurements are useful and robots operate under uncertainty, including underwater monitoring [8], precision agriculture [9], disaster mitigation [10], and subterranean exploration [11, 12].

Collaboration among robots presents advantages in three dimensions: networking, positioning, and task performance. Robots can forward information to extend the effective exploration range under connectivity constraints [13, 14]. They can use inter-robot detection and/or ranging to improve localization [15, 16]. Through coordination, robots can improve task-specific efficiency [17, 18].

While multi-robot exploration, connectivity-maintenance, and relative localization are each well-explored research areas individually, (1) the applicability of previous methods to communication-limited environments where connectivity must be sacrificed to perform exploration is yet to be fully explored, and (2) their joint consideration, i.e. exploration algorithms which explicitly account for data transfer and relative localization performance, has received little attention.

In this proposal, we focus on the application of mapping unknown environments with multiple robots who share this information with a stationary data sink. We look into three key problems in the field of robotic network exploration. Can a distributed exploration algorithm make online decisions which meet a constraint on the queuing delay of new information, ensuring data is transferred in a timely manner? How should the robotic network move to minimize geometry-dependent localization uncertainty while exploring? Is it feasible for a network of computationally-constrained robots equipped with inter-robot ranging devices to localize in a shared, stationary frame based on only these measurements and their odometry? As initial steps towards addressing these problems, we have developed (1) a queue-stabilizing framework for networked multi-robot exploration, and (2) a low-complexity algorithm for trilateration and mapping with robotic networks.

In our first study, we explore the design of exploration strategies for applications where intermittent connectivity is intentional. Consider a lunar lander and several scouting robots deployed to map the surrounding site. Assuming the robots act autonomously, it may be beneficial to sacrifice instantaneous connectivity in order to explore areas beyond line of sight or out of range. However, the lunar lander must receive all data in order for the data to be transferred to Earth. As robots may randomly fail, it is crucial for this data transfer to happen in a timely manner. We present a novel distributed controller that balances exploring with strategically restoring connectivity to a stationary data sink from time to time to prevent data loss. The configuration of robots affects both the network connectivity and the accuracy of relative localization. Strict constraints on these metrics limit exploration, so we take a more flexible approach to managing these multiple aobjectives. We introduce a Lyapunov function which depends on the amount of untransferred data, the time this data has been waiting in the queue, and metrics on connectivity and localizability, which

are functions of the network configuration and can be estimated locally. From this we derive a distributed, online controller which balances maximizing new information and stabilizing this Lyapunov function. Previously, myopic/greedy algorithms have been overlooked in this setting since periodic, rendezvous-based, and deployment-based approaches all require optimizing over possible paths [19]. Our controller provides the distinct advantage that computations are over a single time step, yet we can make formal claims about the time-average behavior. We analyze a simple form of our queue-stabilizing controller and prove that the average queuing delay is bounded, and each robot is guaranteed to recover connectivity when feasible. We then demonstrate in simulation that our controller can reduce localization uncertainty and improve coverage over state of the art approaches.

In our second study, we focus on localization in infrastructure-denied environments i.e. settings where GPS is unavailable and reference points like stationary beacons do not exist. We consider a network of four or more robots which measure and communicate position-related data in order to act as a cooperative positioning system. We present Trilateration for Exploration and Mapping (TEAM), a novel algorithm for low-complexity localization and mapping which leverages commercially-available ultra-wideband (UWB) radios on board the robots to provide range estimates with centimeter accuracy and perform anchor-less localization in a shared, stationary frame. Via coordinated mobility, ad hoc multi-hop communication, and the ability to trilaterate a position estimate, we demonstrate that four or more resource-constrained robots can develop accurate maps of unknown environments. Our contribution is the design and experimental evaluation of TEAM which (1) requires an order of magnitude less computational complexity than a Rao-Blackwellized Particle Filter [20, 21]; (2) reduces the necessary sample rate of LiDAR measurements by an order of magnitude; (3) can reduce the maximum localization error by 50%; and (4) achieves up to a 28% increase in map accuracy in feature-deprived environments and comparable map accuracy in other settings. These computational, power consumption, and performance advantages are obtained at the cost of overall task efficiency, as the robots are periodically static in order to trilaterate. We demonstrate TEAM in several simulation environments and evaluate its performance on a network of Turtlebot3 Burger mobile robots.

In these two works we have proposed an exploration algorithm which accounts for localization performance and data transfer requirements and a localization and mapping algorithm which leverages collaboration to reduce the complexity of localizing. Building on this, we propose integrating the two in order to evaluate our queue-stabilizing framework experimentally on hardware. We also propose three directions for studying how increased collaboration can make the integrated system more scalable and robust: (1) sharing queue states may improve overall exploration efficiency, (2) sharing inter-robot ranging measurements may improve localization accuracy, and (3) collaborating to efficiently select instantaneous anchors may improve UWB channel utility and localization accuracy. We would also like to explore how the multipath components of UWB signals can be used to assist localization and mapping, which is particularly of interest for exploring dusty environments where light-based sensors are rendered unusable over time.

The rest of this thesis proposal is organized as follows. In Sec. 2 we present a review of the relevant literature and preliminaries for the relevant methodologies. In Sec. 3 and 4 we discuss the details of our two studies and highlight theoretical, numerical, and experimental results. In Sec. 5 we propose several directions for future work, and conclude in Sec. 6 with a proposed timeline and summary of the contributions of this thesis.

2 Related work

We will divide our discussion of the relevant prior work into four categories: multi-robot exploration, localization and mapping in multi-robot systems and network localizability, connectivity management in multi-robot teams and queuing networks, and distributed allocation of tasks and wireless medium access.

2.1 Exploration

Exploration is defined as the act of moving through an unknown environment while building a map that can be used for subsequent navigation, and exploration strategies decide, given what is known about the world, where to move to gain as much new information as possible. A seminal work by Yamauchi on robotic exploration introduces the concept of *frontiers*, regions on the border between space known to be open an unexplored space, and describes methods for detecting and navigating to these frontiers [22]. This approach extends easily to multiple robots who share their map representations [23]. In the last decade, advancements in approximations of mutual information for ranging sensors [24] has led to the development of exploration approaches that seek to directly maximize mutual information [25, 26]. Information-theoretic exploration strategies decide, given a probabilistic sensor model, where to move to minimize uncertainty in the map. The map is typically represented as an occupancy grid, where cells are either occupied or free with some probability [27].

Information gathering, a closely related problem to exploration, can be viewed as sequential decision processes in which actions are chosen to maximize an objective function, and the decision is typically solved myopically by maximizing over a limited time horizon [28]. Rather than selecting a frontier and navigating towards it using a path-planning algorithm, this approach chooses the next finite number of positions which will be the most informative. From a decision-theoretic standpoint, the sequential decision-making problem of exploration in as instance of a partially observable Markov decision process (POMDP). The difficulty in expressing model parameters for the exploration problem as a continous POMDP, much less scaling algorithms which solve the problem [29], makes their use challenging for real-world applications [30]. Reinforcement learning is a popular approach to solving large sequential decision-making problems by letting the learner optimize its performance from experience, and has been successfully applied to robotic exploration [30]. Reinforcement learning with multiple robots introduces new challenges for scalability, as the number of states can grow exponentially with the number of robots [31].

When exploring with multiple robots, several questions arise. How should they coordinate to improve exploration efficiency [18]? Can they make these decisions in a distributed manner [26, 32]? How and what information should they communicate with each other [28]? These questions are largely concerned with improving exploration efficiency and generally result in systems in which robots head in opposite directions - intuitively this is the fastest way to cover a space. As we will see in the next two sections, this behavior can be at odds with other objectives. The question of distributing frontier goals, for example, is also closely related to literature on distributed task assignment and allocation which we discuss further in section 2.3.

2.2 Localization and Mapping

While exploring, robots need to address two critical problems: mapping the environment and finding its location within the map. Simultaneous localization and mapping (SLAM) algorithms seek to localize a robot within a previously unexplored environment while constructing a consistent and incremental map. The interdependence of localization and mapping raises the complexity

of the problem, but is well studied. Less well-studied is the interdependence of exploration and localization [33], although the choice of movement does affect how well the robot can localize with respect to the map.

Various solutions have been proposed for SLAM, and while most focus on the single-robot case, there is also a large body of work on multi-robot SLAM [34]. Two of the most important and well-studied questions in multi-robot SLAM are how to represent and share local maps, and how to merge multiple maps with the appropriate transformations. To determine the relative positions between robots, many approaches to SLAM infer this data from the maps themselves. Some approaches incorporate inter-robot detection to improve relative position estimates. This inter-robot detection can improve the localization accuracy [35, 36], however as the network scales this introduces the robot disambiguation problem. Few approaches use inter-robot ranging to improve on absolute position systems, where two robots can directly measure the distance between themselves [37].

We have previously discussed localizing with respect to maps, but localizing with respect to landmarks is another relevant research domain. Fixed elements in the environment in known locations such as visual markers, motion-capture areas, and wireless beacons can allow accurate localization and are commonly used for robots operating in indoor environments, like factories and warehouses [38, 39]. Recently, ultra-wideband (UWB) wireless radios have emerged as high-accuracy localization solutions, with commercially-available products able to offer centimeter accuracy [40]. One significant advantage of UWB is that it offers the ability for low-complexity trilateration [41]. A recent survey by Shule *et al.* on UWB localization for collaborative multi-robot systems highlights several trends [42]. Most existing work relies on a set of fixed anchors with known positions, and uses Kalman filters or least squares estimators for tracking robots [43]. Fusing UWB ranging data from these stationary anchors with other sensors is a popular research direction [42, 44]. Few works consider UWB transceivers onboard the robots for relative positioning. As a method for inter-robot ranging, UWB has recently been demonstrated for autonomous docking [45], formation flying [46], and leader-follower [47, 48].

The localization problem is not unique to robotic applications, and in particular there is a significant body of research on localization in wireless sensor network, where resources are often more limited [49, 50]. In the network localization problem, nodes determine their locations by measuring the distance to their neighbors [51]. It is an instance of the more general problem of mapping n points to a low-dimensional space consistent with measurements between them (sometimes called *dissimilarities*), which has many applications. In information visualization, the related problem of displaying complex data in 2D is commonly called graph realization, graph drawing, or graph embedding [52]. In cyberphysical systems, determining the physical locations of devices is typically called sensor network localization [53]. In machine learning, embedding datasets which exist in high-dimensional space into low-dimensional manifolds can be highly useful for things like semantic analysis, and is commonly called manifold learning [54]. Many approaches exist to solving the network localization problem, including principal coordinates analysis [55], multidimensional scaling [56, 57], manifold learning algorithms like Isomap and Locally Linear Embedding [58], iterative approaches like stress majorization [59], and distributed approaches to local multilateration [60, 61].

The network localization problem is, given a network graph and distance measurements between neighboring nodes, determine a configuration of all nodes in two (or three) dimensions which is consistent with the given measurements [51]. The network localization solvability problem asks whether there is exactly one solution, and is closely related to the rigidity of the graph. *Rigidity* is a property of a graph which means that the edges present are enough to constrain the shape of the graph. For multi-robot systems, the accuracy of localization, exploration, and mapping can benefit from rigid formations [62]. In [62, 63, 64], the authors design controllers which increase or constrain localizability, deriving results from rigidity theory. However, even rigid graphs can

experience ambiguities which lead to challenges in robotic network localization [65, 66]. These ambiguities occur, for example, when a network of four nodes exist and the first three are co-linear. While their configuration could be rigid, it is impossible to determine whether the fourth is on one side of the line or the other. The Cramer-Rao bound, an alternate metric, provides a lower bound on the covariance of any unbiased location estimator and has been shown to accurately predict tracking performance [67]. The geometric dilution of precision (GDOP), commonly used to describe how localization error depends on GPS satellite positions, is the ratio of location estimate variance to signal variance and is closely related to the Cramer-Rao bound [68]. Some of these metrics, which depend on the geometry of of the network and can be modeled via the underlying graph, are very closely to metrics used to evaluate connectivity in the next section.

2.3 Medium Access and Role Assignment

In the previous section, we discussed UWB positioning as a popular approach to localization. Commercial, anchor-based systems mainly use Time Difference of Arrival (TDOA), a technique which requires nanosecond clock synchronization between anchors, limiting scalability and making distributed operation challenging [69, 70]. Two-way ranging (TWR) is a more flexible ranging solution, but introduces the key challenge of access control to a shared medium. A medium access control (MAC) mechanism must exist to coordinate measurements across all devices. The most common protocols include Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), and Carrier Sense Multiple Access with Collision Avoidance [71]. Previous work has considered MAC protocols specific to UWB networks [72] and protocols suitable for robotic networks [69, 73]. TDMA is often chosen as it allows high channel utilization, especially when scheduling over multi-hop networks allows the re-use of time slots, but the mobility of robots presents significant challenges for TDMA scheduling. Initial efforts toward this challenge aim to maximize channel utilization [69], but it is important to acknowledge that this problem is closely related to that of anchor selection. This is because the scheduling mechanism dictates which range measurements will be the most recent.

When a network localization problem has nodes with known locations, the problem of selecting anchors or reference nodes has been studied. Yang [74] proposed the idea of estimating the quality of trilateration to select anchors, while Cao *et al.* [69] recently extended this to the case where no nodes have known locations. Selecting anchors can be considered an instance of the general class of role allocation problems. Campbell and Wu [75] survey the literature on multi-agent role allocation and present an overview of explicitly and implicitly defined roles. They define the difference between a *task*, a piece of work an agent performs, and *role*, a description of an agent's responsibilities. In multi-robot task allocation, market-based approaches are an attractive compromise between centralized and distributed approaches [76], and have precedence in multi-robot exploration [32].

2.4 Connectivity and Queues

Many of the previously listed approaches for multi-robot exploration, localization, and mapping rely on communication between robots, and therefore connectivity. Connectivity-aware planning has been studied for the single robot and multi-robot settings [77, 78]. Control strategies to maintain connectivity in multi-robot systems have been studied [79, 14, 80, 81]. These works have created a strong precedent for the use of graph theoretic concepts when discussing communication. We consider three specific metrics: two-terminal network reliability, k-hop-connectivity, and algebraic connectivity. Consider a communication graph where vertices represent robots and edges represent the probability of successful communication. The two-terminal network reliability is the probability

that at least one successful path exists between two vertices [82]. This is known to be NP-hard to compute for a general graph [83], but methods exist to compute it exactly [84, 85]. The closely related k-hop-connectivity of two vertices is the weighted number of paths of k hops or fewer which connect the first to the second, where weights indicate the probability of success along the path [14]. Both of these metrics will increase as the number or reliability of the paths which connect two robots increases. For a given communication graph, it is possible to describe the adjacency matrix whose entries represent pairwise connections. From this we can compute the weighted Laplacian, whose second-smallest eigenvalue is known as the Fiedler value. The Fiedler value is considered an important value in graph theory, and quantifies the algebraic connectivity of the graph [86].

Maintaining connectivity at all times can limit the network of robots in certain applications [81]. Strategies for application where intermittent connectivity is sufficient or even intentional are less well-researched, but several approaches are explored in the literature. A recent survey by Amigoni *et al.* provides an overview of communication-restricted multi-robot exploration [87]. Some take a periodic approach [19], others are rendezvous-based [88], and others are role-based and assign connectivity and exploration as distinct tasks [89, 90, 91]. Deployment-based strategies exist which allow robots to lose connectivity en route to a predetermined configuration [10]. Multi-objective approaches exist and allow the robots to trade between connectivity and other metrics [92].

When robots do not have constant connectivity at all times, it makes sense to model their store of local data as a queue. As they generate or collect data, this information waits to be transferred. We can borrow concepts from queuing theory to study the nature of these queues, and from control theory to study their stability, i.e. whether or not they will grow unbounded [93]. For example, a Lyapunov function is a scalar function used to analyze stability, and Lyapunov drift is the change in this function over one time step. Minimizing Lyapunov drift to stabilize a queueing system was first presented by Tassiulas and Ephremides, in an algorithm called *max-weight scheduling* [94, 95]. In brief, this algorithm observes the current channel conditions and makes a decision on which queue to serve based on a sort of memory of past decisions which is captured in the queue states. It provides a very simple yet powerful way of designing near-optimal controllers in stochastic settings. In [96], Neely introduces the *virtual queue* terminology which generalizes this approach to constrained optimization via the *drift-plus-penalty* approach. Briefly, this algorithm seeks to minimize a penalty function while also minimizing Lyapunov drift to ensure the time-average expectation of a bounded metric is constrained.

While these tools for designing queue-stabilizing controllers are often used to solve problems like routing in static networks [97], applying them to mobility control in robotic networks is still an open research area. Previous work has considered robot allocation and deployment [98] and robotic message ferrying [99] between stationary data sources and sinks. In this work we are interested in modeling multi-robot exploration as a queuing network, in which exploration decisions affect whether the robots gain new data (increasing their queue) and/or can communicate with a data sink (emptying their queue). To the best of our knowledge, this is a novel way of modeling multi-robot exploration.

3 TEAM: Trilateration for Exploration and Mapping

In this study, we leverage UWB relative positioning devices on board four or more robots to design a low-complexity localization and mapping algorithm which leverages collaborative localization and coordinated mobility. An high-level overview of the approach is shown in Fig. 1. This work extends the idea of a cooperative positioning system first presented by Kurazume and Hirose [100, 36]. We take a similar approach to localization, but introduce the performance gains offered

by UWB and study the challenges UWB presents, namely medium access. Additionally, our work introduces explicity inter-robot communication and considers the application of mapping unknown environments.

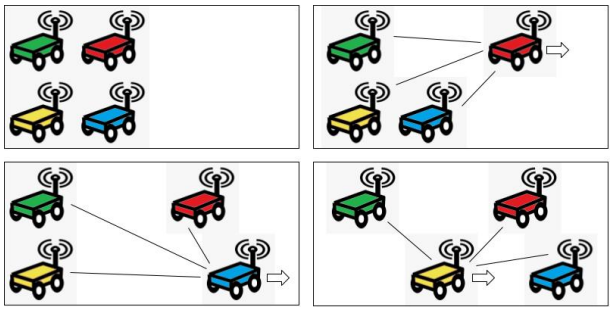

Figure 1: TEAM leverages collaborative localization and coordinated mobility. Top left: robots initiate a shared coordinate frame. Top right: the first robot moves and maps while using its neighbors to trilaterate. Bottom left and right: subsequent robots take turns moving and mapping.

3.1 Algorithm

TEAM is described in Algorithm 1. After initializing position estimates, each robot iterates through lines 2-12 wherein they map their environment and exchange their coordinates and maps. During a robot's TDMA window (see Fig. 3), the robot drives autonomously, initiates two-way ranging, and trilaterates. The following subsections present each component of the algorithm.

Algorithm 1: TEAM running on robot i

```
1:  coords, neighborCoords = Initialization()
2:  while True do
3:      neighborCoords = RadioReceive()
4:      if currentTime is in TDMAwindow_i then
5:          odom = DriveAutonomously()
6:          neighborDistances = UWBRange()
7:          coords = TrilateratePosition(odom, neighborCoords, neighborDistances)
8:      end if
9:      scan = LiDARScan()
10:     map = UpdateMap(coords, scan, map)
11:     RadioTransmit(coords, map)
12: end while
```

3.1.1 Initialization

Initialization occurs in line 1. If initial positions are unknown, TEAM can determine initial positions given that each robot has a known and unique identifier. Robot 0's position defines the map origin and the forward direction. Robot 1's position defines the y axis and robot 2's position defines the positive direction of the x axis. Robot 3 and any additional robots can then trilaterate as discussed below. If initial orientations are unknown, each robot can move a known distance in its local forward direction and then trilaterate to determine orientation. If initial positions are known,

TEAM leverages this data to perform sensor auto-calibration, estimating the sensor offset from the truth and using that bias to correct future measurements.

3.1.2 Trilateration

Robots receive position estimates from their neighbors via RadioReceive in line 3, and collect range measurements via UWBRange in line 6. Given the positions and distances from the three closest instantaneous anchors, TrilateratePosition (line 7) determines a position estimate as shown in Figure 2. The grey rings represent uncertainty in the two-way ranging estimate, and the estimated position is the centroid of the curved triangle formed by their intersection points. If fewer than three anchors are within the communication range, TrilateratePosition relies on the robot's previous position estimate and odometry to determine an updated position estimate. For a robotic network of at least five robots, TEAM can trilaterate in three dimensions.

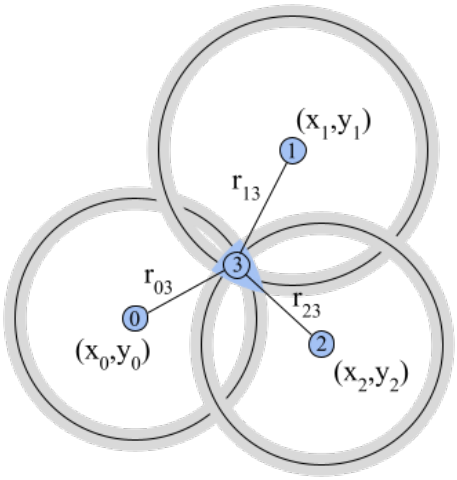

Figure 2: Robot 3 trilateration, using the received position estimates of robots 0, 1, and 2 as anchors.

To prevent UWB signal interference, TEAM uses a time-division multiple access (TDMA) protocol such that lines 5-7 are only executed during a specified window as illustrated in Fig. 3. We selected TDMA because it offers high channel utility and has extensions suitable for large teams of cooperating robots [43, 73].

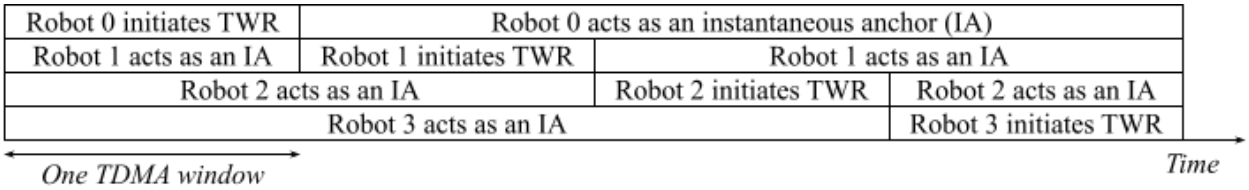

Figure 3: Time-division multiple access scheme for 4 robots performing TEAM. Each robot initializes two-way ranging (TWR) during its TDMA window, and this cycle repeats.

3.1.3 Mobility

TEAM is suitable for both autonomous exploration and teleoperation. In this work we are focused on the resource-constrained setting and elect to use simple drive control (line 5) in which robots

are given a target direction and perform autonomous collision avoidance based on LiDAR data.

To improve positioning accuracy, robots are stationary outside of their TDMA window. Note that with four robots, this scheme takes four times as long to cover a sum total distance when compared to all robots driving simultaneously. This highlights an important system design trade-off between prioritizing quick or accurate map coverage. For larger networks, the three robots required to act as instantaneous anchors are a smaller fraction of the network and the reduction in sum total distance covered is less significant.

3.1.4 Mapping

Each time the LiDAR takes a measurement (line 9), this data is associated with the current position estimate. The occupancy grid map representation is then updated (line 10) with the LiDAR data. The resulting map is shared with the data sink via RadioTransmit in line 11. To reduce the burden on the resource-constrained robots, the data sink is responsible for merging the maps received from all sources. Using the map merging algorithm from [101], the data sink is able to perform feature mapping and calculate the appropriate frame transforms to merge all received maps. This process does not require synchronization between robots, and the data sink can receive a map as long as any multi-hop communication path is available, as discussed below.

3.1.5 Communication

Under the assumption that all robots are within communication range, the publish/subscribe paradigm provided by the Robot Operating System (ROS) is sufficient for sharing positioning and map data (lines 3 and 11). However, lava tubes are characterized by branches which challenge connectivity [3]. Previous work exploring tunnels has considered a variety of approaches including data tethers and droppable network nodes [11]. This work builds on the approach of [102] with mobile network nodes which can forward or re-route data. The robots and data sink form an ad hoc mesh network, in which robots can act as relays to the data sink when needed, using Optimized Link State Routing (OLSR) to determine network neighbors and appropriate routes [103].

3.2 Complexity

In TEAM, selecting the three nearest anchors from the anchors within range can be done in $O(NlogN)$ where N is the size of the network. Calculating the location estimate is then a constant number of operations and several closed-form and approximate algorithms exist [41]. Associating scan data with a location estimate and updating the map can also be done in constant time, as new scans are integrated independent of the size of the existing map.

We compare TEAM with an improved Rao-Blackwellized Particle Filter (referred to here as SLAM), a Monte Carlo localization algorithm which is available via the GMapping library [20, 21]. Following the complexity analysis in [20], SLAM introduces complexity $O(P)$ each time the location estimate is updated, where P is the number of particles. This computation is associated with computing the proposal distribution, computing the particle weights, and testing if a resample is required. SLAM also introduces complexity $O(P)$ for each map update, and complexity $O(PM)$ each time a resample occurs, where M is the size of the map. For an optimized system, the number of particles required is typically between 8 and 60 [20], depending on the size and features of the environment. Thus, TEAM can provide up to a 60x reduction in computational complexity. Table 1 contains the average runtimes for relevant computations on our hardware.

Computation	SLAM, P=60	SLAM, P=1	TEAM
Trilaterate	-	-	0.001 s
Update map	0.40 s	0.1 s	0.1 s
Process scan	2.25 s	0.05 s	< 0.001 s

Table 1: Runtime comparison

3.3 Simulation Experiments

3.3.1 Setup

We use the Gazebo robotics simulator to conduct experiments in various environments. We implemented UWB ranging in simulation according to the following model:

$$d_{pozyx} = \begin{cases} d_{true} + N(\mu = 0, \sigma = 10cm), \text{ if } 1_{LOS} \\ \text{None, otherwise} \end{cases} \tag{1}$$

where d_{true} is the true distance between two robots and 1_{LOS} is an indicator function which evaluates to true if and only if line of sight is available.

Table 2 presents relevant parameters used in the simulation experiments. We assume UWB measurements have zero mean and a standard deviation of 10cm [40]. Each call to UWBRange returns the average of 10 independent measurements. To ensure synchronization between the LiDAR scans and UWB ranging measurements, a delay of more than 100 ms between the two sensors prevents TEAM from updating the map with this scan data. This timeout was empirically selected.

Note that with time-division mobility in the drive module, our approach generates maps at a slower pace. In simulation, we choose to not model the UWB interference constraint and allow the robots to range simultaneously and therefore drive simultaneously. This allows us to compare SLAM and TEAM across the same time scale.

Parameter	Value
Number of robots	4
Number of UWB measurements averaged	10
Std dev of UWB measeurements	10 cm [40]
LiDAR/Trilateration synchronization timeout	100 ms
Map publish rate	1 Hz
Position estimate publish rate	10 Hz
Max drive speed	0.22 m/s
360° LiDAR sample rate	5 Hz
360° LiDAR resolution	1°
360° LiDAR standard deviation	0.01
RBPF number of particles	50
Odometry variance	0.01 mm

Table 2: Relevant parameters for simulation experiments

3.3.2 Results

The first environment we consider challenges SLAM but is well-suited for TEAM: a long, featureless, obstacle-free corridor. This environment is difficult for a particle filter because the particle

distribution becomes spread as the robots navigate along the hallway which lacks discernible features for localization [20]. The results of this experiment are shown in Figure 4; they indicate that TEAM can significantly improve map accuracy in featureless environments.

The second environment we consider challenges TEAM but is well-suited for SLAM: an environment with lava tube-like characteristics, including branches that prevent line of sight. The Gazebo model for this environment comes from [104]. TEAM relies heavily on odometry data in this environment, and the results of this experiment are shown in Fig. 5 and strengthen our claim that TEAM results in accurate maps. We demonstrate the performance in the presence of noisy odometry data in Sec. 3.4.2.

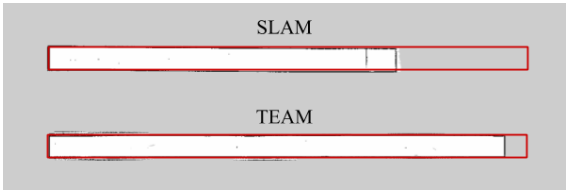

Figure 4: The top image resulted from SLAM with 50 particles, the bottom image resulted from TEAM, each after 25 simulated minutes. The true environment dimensions are overlaid in red.

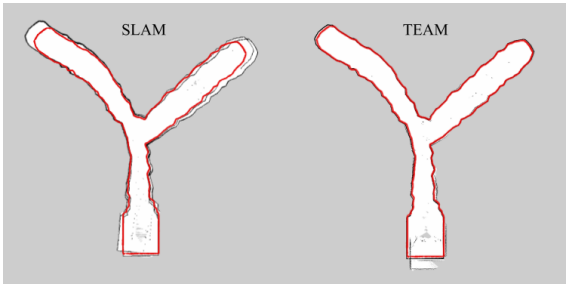

Figure 5: The left image resulted from SLAM with 50 particles, the right image resulted from TEAM. The true environment dimensions are overlaid in red.

One significant advantage of TEAM is that it decouples localization from LiDAR sensor measurements. This allows us to reduce the frequency of LiDAR scanning in order to save power; this is useful for robots that have severe energy constraints or memory limitations. Fig. 6 shows the result of an experiment in which scans are throttled to 0.1Hz from 5Hz used in the previous experiments. This shows that reduced LiDAR frequency significantly deteriorates the performance of SLAM, while leading to topologically correct albeit sparse maps with TEAM.

As discussed in Sec. 3.2, the algorithmic complexity of SLAM is a function of the number of particles used to capture the belief distribution of the location estimate. For TEAM, the algorithmic complexity is comparable to SLAM with a single particle. In Fig. 7 we illustrate the performance of SLAM with 1 particle and TEAM side by side, to highlight the difference in map accuracy despite similarly reduced computation time.

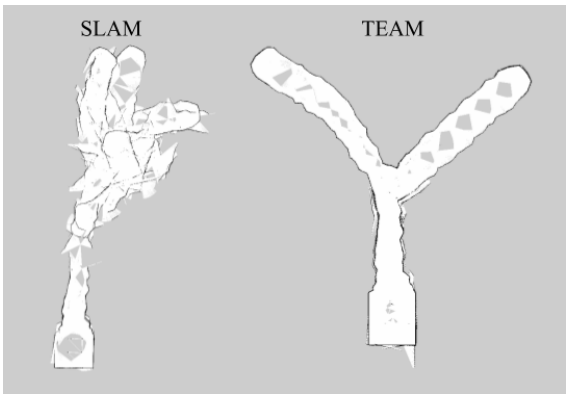

Figure 6: The left image resulted from SLAM with 50 particles and scans throttled to the low frequency of 0.1Hz. The right image resulted from TEAM with the same low LiDAR sample rate.

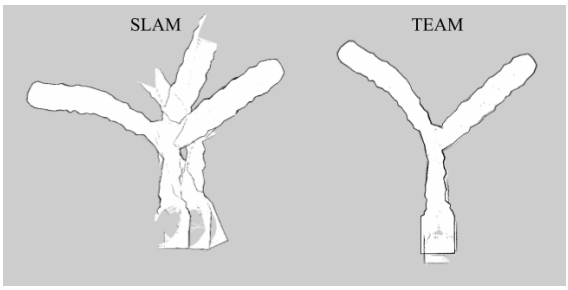

Figure 7: The left image resulted from SLAM with 1 particle. The right image resulted from TEAM, which is computationally comparable.

While the merged map presents a clear picture of the capabilities of TEAM, it is also useful to consider the localization error of the robots over time. For SLAM, the localization uncertainty varies as the belief distribution changes over time. In feature-deprived environments like the one shown in Fig. 4, this error can grow up to 25m. Fig. 8 compares the localization error as a function of time during exploration of the lava tube-like environment. We observe that errors in TEAM are less than 3m and these errors are infrequent, however errors in SLAM can grow up to 6m and are higher on average.

It is similarly useful to quantify the accuracy of the merged map created by each algorithm over time. We measure the absolute pixel error of map image files which were manually aligned with and compared to the image file of the true environment. We plot the map accuracy over time and show that while SLAM and TEAM both result in decreasing map error, TEAM achieves a lower final absolute pixel error (Fig. 9). We observe that TEAM achieves a 28% decrease in error relative to SLAM for the environment depicted in Fig. 4, and a 27% decrease in error for the environment depicted in Fig. 5.

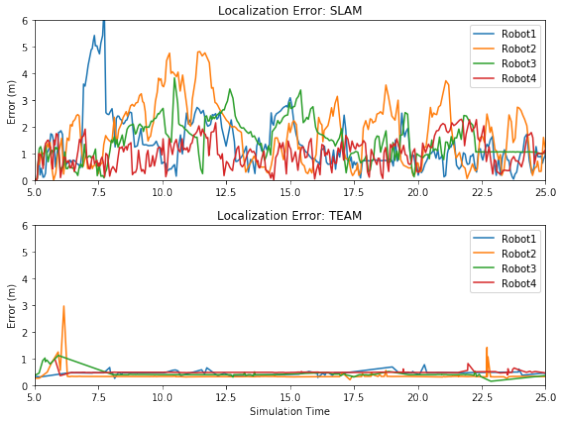

Figure 8: These graphs show the localization error over time for SLAM and TEAM for the environment depicted in Fig. 5.

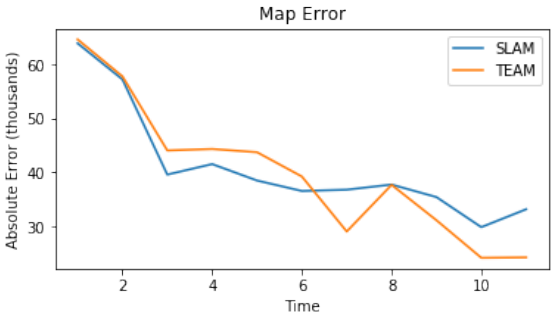

Figure 9: This graph shows the map error as a function of time for the environment depicted in Fig. 5.

3.4 Testbed Experiments

3.4.1 Setup

Our testbed is comprised of four Turtlebot3 Burgers, each equipped with a RaspberryPi 3B+ running Raspbian, an OpenCR1.0 control board, a 360 Laser Distance Sensor, and differential drive. We have extended each platform with a Pozyx UWB Creator series Anchor, and a USB Wireless Adapter Mideatek RT5370N with 2dBi antenna. The wireless adaptor allows each robot to designate one wireless interface for internet connectivity, and one wireless interface for joining our ad hoc mesh network, TurtleNet. The robots and data sink, a PC running Ubuntu 16.04, all implement Optimized Link State Routing (OLSR), a proactive routing protocol for mobile ad hoc mesh networks [103, 105]. We provide a tutorial for mesh networking Turtlebot3 Burgers [106].

Table 3 presents parameters used in the testbed experiments in addition to those previously listed in Table 2. The size of the TDMA window should be larger than the time to collect 10 ranging measurements (0.5 sec) and smaller than the time it would take a robot to drive beyond the maximum communication range (90 sec). Delays between the robots and the data sink do not affect the quality of the resulting merged map, as it is processed asynchronously; however, delays or dropped packets between robots affect the location estimate accuracy. We introduce a 0.3 second buffer period at the end of each TDMA window during which the robot stops driving but continues to perform trilateration. Note that for consistency, our time-division approach to driving autonomously was used across all experiments.

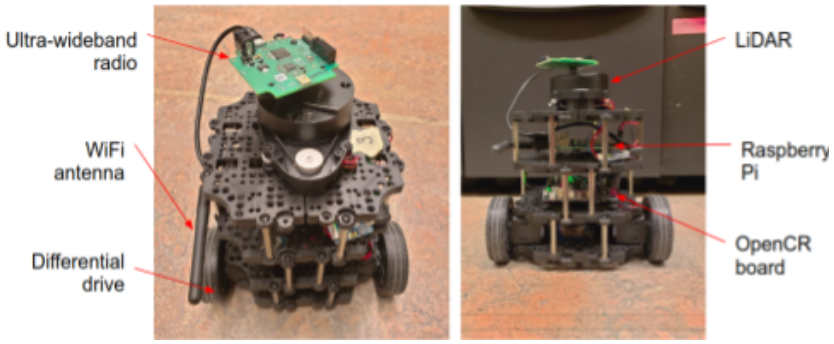

Figure 10: Turtlebot3 Burger with Pozyx UWB anchor.

Parameter	Value
TDMA window size	5 s
UWB ranging rate	20 Hz
Worst case expected communication delay	400 ns

Table 3: Additional relevant parameters for testbed experiments

3.4.2 Results

Fig. 11 shows the merged map created by our testbed in a typical hallway environment using SLAM. Fig. 12 shows the map created by a single robot, illustrating the effect of inaccuracy in the odometry readings. We noticed that the drift in odometry was not consistent across the four robots, but the map merging algorithm mitigated this. For better location estimate accuracy, the robots should localize within the merged map rather than their individual maps.

Fig. 13 shows the merged map created using TEAM. We observe that the wall edges are slightly less well-defined. This is due to noise in the UWB ranging measurements causing small jumps in the position estimate. This noise is primarily attributed to multipath effects, and further characterizing the UWB performance will help improve map accuracy [37]. Fig. 14 shows the individual map of a single robot and highlights the effect of loss of line of sight. This causes the robot to rely on its odometry until UWB signals are available again, at which point its location estimate may jump, leading to discontinuities in the map. Distributed strategies to maintain connectivity as presented in Sec. 4 can prevent these discontinuities.

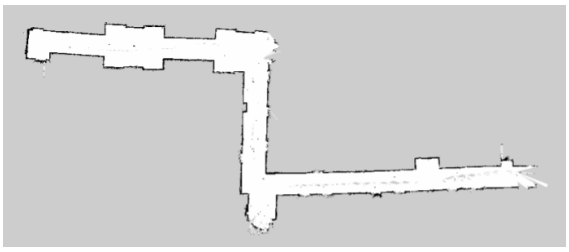

Figure 11: SLAM in the hallway: merged map

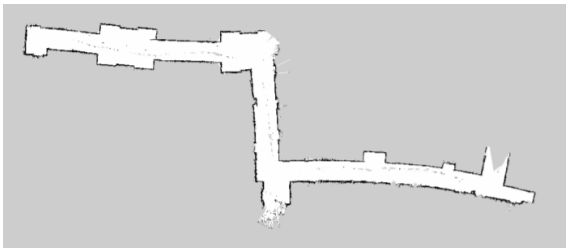

Figure 12: SLAM in the hallway: map created by a single robot

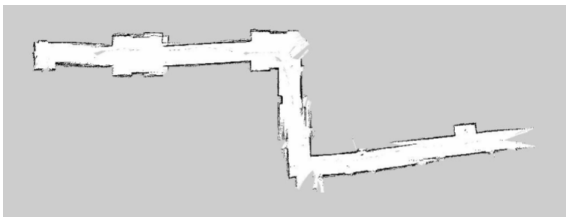

Figure 13: TEAM in the hallway: merged map

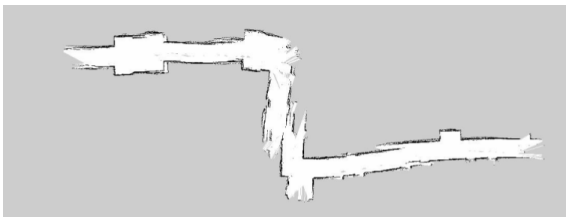

Figure 14: TEAM in the hallway: map created by a single robot

3.5 Summary

In this section we have presented the design, implementation, and evaluation of TEAM, a novel algorithm for localization and mapping in unknown environments with a robotic network. We demonstrate the ability of TEAM to leverage ultra-wideband positioning to generate maps of various environments with high accuracy. Our algorithm significantly reduces the computational complexity and the required rate of LiDAR samples, making it suitable for resource-constrained multi-robot systems, and performs well in feature-deprived environments where SLAM struggles. We observed several challenges for TEAM when evaluating on hardware. The first challenge occurs when connectivity is broken, and robots can no longer range each other. The second challenge occurs when three robots are co-linear, and the fourth cannot uniquely determine its location. From these observations we determined that controlling the configuration of robots to enable communication and localizability are two important objectives, and we consider these explicitly in our second study.

4 Queue-stabilizing Exploration Framework

In this study, we apply control strategies derived from the stability analysis of queues to design an exploration algorithm which myopically maximizes new information subject to constraints on timely data transfer and high localization accuracy. Similar to the closely related work of Benavides *et al.* [92], our work weighs discovering new information with maintaining connectivity and adapts to a human operator's input on the importance of each objective. However their approach assigns a static utility to connectivity, while our approach depends on whether there is new information to share. The closest work to ours is the work by Spirin *et al.* [91], which introduces a constraint on the ratio of information at the data sink and a crude approximation of total information across all mobile robots. Like this prior work, we show desirable emergent behavior by adapting to the amount of untransferred data (in our case we also adapt to how long this data has been waiting). But unlike this work, our approach is more flexible than their role-based approach (*explore* or *return*). Additionally, we explicitly consider multi-hop communication which provides advantages over relaying opportunistically.

4.1 Problem Formulation

The objective is to design a distributed controller for a multi-robot system where each robot r in the set of robots $\mathcal{R}$ can collect information about its environment and store this information in a local map m_r. Each robot can communicate with any robot i for which link quality $f^c_{r,i}$ is above some threshold θ_c. Map information is shared over these communication links such that $m_r \leftarrow m_r \cup m_i$ for each pair (r, i) for which $f^c_{r,i} > \theta_c$. Robots also share their time-stamped location

estimate, which times out after t_{to} time steps. This means a robot has an absolute location estimate for any robot it has communicated with (through one or more hops) in the last t_{to} time steps, and we call this set of robots $\mathcal{C}_r$. Each robot r can also collect a range measurement between itself and any robot i for which link quality $f^l_{r,i}$ is above some threshold θ_l. These measurements containing range estimates and robot identifiers can be obtained, for example, via ultra-wideband radio and used for relative localization [16, 15]. Note that since communication and localization may use different technologies, f^c and f^l may not have the same dependence on distance and may have different levels of noise. We assume the presence of a stationary data sink D at position x_D equipped with the same communication and ranging capabilities as the robots. The goal of the system is to create a complete map at the data sink m_D.

We assume time is discretized, $t \in \{0, 1, ...\}$. The state $S_r(t)$ captures what is known by robot r at time t: the position of the robot $x_r(t)$, the robot's map $m_r(t)$, and the estimated position of all robots $x_i(t)$ in the set $\mathcal{C}_r(t)$. At each time t, the controller deployed on a single robot observes the current state and makes a myopic decision $\alpha_r(t)$ choosing from the discrete finite set of locations which can be reached in the next time step, $\mathcal{A}_r(S_r(t))$. We assume perfect obstacle detection, therefore we limit the decision space to obstacle-free locations and handle obstacle avoidance trivially.

$S_r(t+1)$ depends on the previous state $S_r(t)$ and decision $\alpha_r(t)$. It also depends on unknowns in the environment and decisions $\alpha_i(t) \forall i \in \mathcal{R}$; therefore, the transition probabilities between states are unknown. We design a distributed controller which makes online decisions $\alpha_r(t)$ based solely on the observable state $S_r(t)$ and memory stored in queues $q_r(t)$, $D_r(t)$, $Q_r(t)$, and $Z_r(t)$. We will define these queues in Sec. 4.2.2, 4.2.3, and 4.2.4.

4.1.1 Probability of Failure

In this setting, robots operate in harsh environments and may experience failures caused by wear and tear on the system or unforeseen hazards. In multi-robot systems, robot failure is typically modeled by a constant hazard rate, γ, and a probability of failure given by $Pr[\text{Robot } r \text{ fails at time } t] = \gamma e^{-\gamma t}$ [107]. To consider robots which are constrained by a given resource (e.g. fuel), we could define the probability of failure as a function of the remaining quantity of that resource, rather than time. For example, in [89] the authors propose an exploration strategy which considers remaining battery life.

4.2 Queue-stabilizing Controller

In this section we describe the four objectives of our controller: (1) the task-specific goal of maximizing new information, (2) keeping the average queuing delay low, to prevent data loss upon failure by increasing the network reliability, (3) maintaining overall network connectivity, which encourages the robots to act as relays, and (4) maintaining network localizability, which keeps location uncertainty low to ensure new information is useful. We then formulate the constrained optimization problem and use Lyapunov drift minimization to derive our multi-objective queue-stabilizing controller. Finally, we conclude this section with a theoretical analysis of the controller's performance.

4.2.1 Information theoretic exploration

We represent the environment as an occupancy grid where cells are occupied or free with some probability. The random variable X_i describes the probability that the i^{th} cell is occupied and the random variable Z_i is associated with the observation of the i^{th} cell, where cells take on values in

$\{0, 1\}$. Our sensor model is therefore the conditional probability $Pr(Z_i|X_i)$. Prior to making an observation, each cell is associated with an entropy which depends on the map thus far, $H(X_i|m_r)$. The entropy after taking observation z_i is $H(X_i|m_r, Z_i = z_i)$. The value of observing the i^{th} cell is given by the mutual information or reduction in entropy, $\mathcal{I}(Z_i; X_i|m_r) = H_i(X_i|m_r) - H_i(X_i|m_r; Z_i)$. We assume X_i are independent (as in [26]). We let $\mathcal{Z}(x_r)$ describe the set of map cells which are within the sensor coverage of the robot, and we define the expected information gain associated with a robot position x_r given map m_r as $I(x_r; m_r) = \sum_{i \in \mathcal{Z}(x_r)} \mathcal{I}(Z_i; X_i|m_r)$.

The local information-theoretic controller presented in [33] maximizes $I(\alpha_r(t); m_r(t))$ subject to $\alpha_r(t) \in \mathcal{A}_r(t)$ where $\mathcal{A}_r(t)$ is the set of map cells accessible within one time step. While this one-step control strategy may not lead to optimal trajectories over multiple time steps, it directs the robot towards increasing information gain and constantly adapts to new information.

This controller, like other local search methods, suffers the risk of getting stuck or oscillating in *plateaus*, areas where all location decisions $\alpha_r(t) \in \mathcal{A}_r(t)$ result in the same value of $I(\alpha_r(t); m_r(t))$. To prevent this undesirable behavior, we present a modification: we define the frontier goal, $x_r^f(t)$, as the closest point which offers positive information gain, breaking ties according to maximum information gain and subsequent ties arbitrarily. Now our information-theoretic objective is to minimize the distance to the frontier goal,

$$Y(\alpha_r(t)) = \left\| \alpha_r(t) - x_r^f(t) \right\|_{m_r(t)} \tag{2}$$

where notation $\|x_1 - x_2\|_{m_r}$ refers to the length of the shortest path from x_1 to x_2 in the map m_r, found via Dijkstra's algorithm. This will direct the robot towards the closest point which increases information.

4.2.2 Stabilizing the Delay Queue

We define the communication graph at robot r at time t, $\mathcal{G}_r^c(t)$, to have vertices representing r, data sink D, and other robots in the set $\mathcal{C}_r(t)$. An edge in this graph e_{ij} represents the probability of successful communication between nodes i and j. Assuming link quality f_{ij}^c falls off exponentially with distance, the probability of successful communication has the shape of a sigmoid function [108] and can be modeled as

$$e_{ij} = \begin{cases} 0 & \text{if non-line-of-sight} \\ \frac{1}{1+e^{\eta(\left\|x_i - x_j\right\| - d_{\theta_c})}} & \text{otherwise} \end{cases} \tag{3}$$

where d_{θ_c} is the distance at which $\mathbb{E}[f_{ij}^c]$ falls below θ_c, η defines the steepness of the sigmoid, and we assume radio-frequency-impermeable obstacles such that the probability is zero if an obstacle obstructs the line of sight path.

We assume that links fail independently. We define p_{ij} to be the two-terminal network reliability, the probability that at least one successful path exists between vertices i and j [82, 83]. After enumerating the set of all simple paths from i to j we can exactly compute p_{ij} by constructing a sum of disjoint products [84]. Let π_{ij}^k be the event *path k exists* which connects $i \rightarrow j$. The probability $Pr[\pi_{ij}^k]$ is the product of the edge probabilities which comprise the path. Let ϕ_{ij} be the event *any path $i \rightarrow j$ exists*, which is the Boolean sum over all simple paths between i and j. To calculate $p_{ij} = Pr[\phi_{ij}]$ we employ the approach presented in [85] to transform a set of simple paths into a set of disjoint events. We then sum the probabilities of these disjoint events. This calculation is intractable as the number of robots increases. For larger networks, we use the *k-hop-connectivity* as an approximate metric [14].

We now introduce a queue $q_r(t)$ of untransferred data stored at robot r with the following dynamics:

$$q_r(t+1) = \max[q_r(t) + I(\alpha_r(t); m_r(t)) - b\mathbb{1}(\alpha_r(t)), 0] \tag{4}$$

where $\mathbb{1}(\alpha_r(t))$ indicates that a successful communication path exists between $\alpha_r(t)$ and data sink D, and b is the (constant) finite capacity or quantity of information which can be transmitted in a single time step. Assuming each robot has finite memory, the queue size is bounded by $q_r(t) \leq q_{max}$, and any additional new information will not be stored. We assume that signal propagation time is negligible, so the robot transfers its data and receives an acknowledgement within one time step.

Intuitively, our objective is to minimize the time that data waits in the queue to be transferred. Following [109], we introduce a threshold on the average delay θ_d, and a virtual delay queue with the following dynamics:

$$D_r(t+1) = \max[D_r(t) - \theta_d b\mathbb{1}(\alpha_r(t)), 0] + q_r(t) \tag{5}$$

where we have carefully ensured that if $q_r(t) > 0$, the delay queue is positive. Note that in this paper we use capital letters to differentiate virtual queues from real queues of data. The delay queue grows with each time step that the data queue is non-empty, and decreases when $\mathbb{1}(\alpha_r(t)) = 1$. The goal of our controller is to achieve *mean rate stability*, defined as $\lim_{t\to\infty} \frac{\mathbb{E}[D_r(t)]}{t} = 0$ with probability 1. In Sec. 4.3, we analytically show that actions which maximize p_{rD}, the probability that at least one path exists between r and D, will stabilize this queue.

4.2.3 Connectivity

Intuitively, we expect that if each robot maximizes p_{rD}, the likelihood it can communicate with the data sink, the result will be a network with high connectivity. Additionally, our framework allows us to also control network connectivity explicitly. Closely following [14], we consider the weighted Laplacian matrix, $\mathbf{L}_r$. This matrix is directly constructed from the adjacency matrix, whose elements are given by $f^c_{i,j}$. The Fiedler value, λ_2, quantifies the algebraic connectivity of the graph [86]. If and only if $\lambda_2 > 0$, the graph is connected. A controller designed to increase λ_2 will result in well-connected graphs with several paths from all robots to the data sink. Unlike edge-connectivity and vertex-connectivity, algebraic connectivity is a continuous-valued metric, well-suited for this setting. Additionally, the Fiedler value increases with the number of robots alive and is closely related to the reciprocal of average distance, both useful in this setting [86].

We use notation $\lambda_{2r}(\alpha_r(t))$ to denote the Fiedler value of the Laplacian if robot r makes location decision $\alpha_r(t)$. Note that λ_{2r} indicates only the connectivity of $\mathcal{G}^c_r(t)$, the graph available to robot r. Since each $\mathcal{G}^c_r(t)$ contains the data sink, if each graph is connected, then the overall network is connected. Thus we can take actions which increase connectivity in a distributed fashion. Rather than constraining λ_{2r} at all times, we take the following more flexible approach.

Consider Q_r, a virtual queue with the following dynamics:

$$Q_r(t+1) = \max\left[Q_r(t) + (\theta_\lambda - \lambda_{2r}(\alpha_r(t))), 0\right]. \tag{6}$$

Stabilizing $Q_r(t)$ will result in a time-average expectation of λ_{2r} which is greater than or equal to the constraint θ_λ. Formally, $\overline{\lambda_{2r}} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[\lambda_{2r}(\tau)] \geq \theta_\lambda$.

4.2.4 Localizability

Creating accurate maps requires accurate localization, but algorithms which result in efficient exploration may not result in high localizability. The trade between minimizing map uncertainty and minimizing localization uncertainty was previously explored by Bourgault *et al.* for a single robot [33]. Similarly to the approach of Le *et al.*, we consider the Cramer-Rao bound (CRB), which provides a lower bound on the covariance of any unbiased location estimator [68].

To calculate the CRB in two dimensions, we first calculate the Fisher Information Matrix $\mathbf{F}_r$, a 2×2 matrix whose elements are given by

$$\mathbf{F}_r[i,j] = \frac{1}{(c\sigma_T)^2} \sum_{k \in \mathcal{C}_r \cup D} \frac{(x_{ir} - x_{ik})(x_{jr} - x_{jk})}{{d_k}^2} \tag{7}$$

where $d_k = \sqrt{(x_{1r} - x_{1k})^2 + (x_{2r} - x_{2k})^2}$ and we assume time of arrival measurements. In the above equation, c is the speed of light, σ_T is the variance of the time delay error, and $(x_{1k}$,$x_{2k})$ are the coordinates of robot k. $\mathbf{F}_r$ and the symmetric rigidity matrix are closely related [62].

The Cramer-Rao bound matrix is the inverse of $\mathbf{F}_r$ and gives a lower bound on the covariance (uncertainty) of any unbiased estimator. We want this bound to be small, so we constrain the trace of this matrix, using notation $\text{CRB}_r = \mathbf{tr}(\mathbf{F}_r^{-1})$. We define a virtual queue $Z_r(t)$ with the following dynamics:

$$Z_r(t+1) = \max\left[Z_r(t) + (\text{CRB}_r(\alpha_r(t)) - \theta_{CRB}), 0\right] \tag{8}$$

where $\text{CRB}_r(\alpha_r(t))$ denotes the Cramer-Rao bound for robot r reflecting location decision $\alpha_r(t)$, and θ_{CRB} is a desired threshold on this bound. Stabilizing this virtual queue will result in a desirable time-average expectation. Formally, $\overline{\text{CRB}}_r = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[\text{CRB}_r(\tau)] \leq \theta_{CRB}$. Note that for certain configurations the CRB is unbounded [66]. To fit our optimization framework, in implementation we enforce $\text{CRB}_r = \min\left(\text{CRB}_r, \frac{1}{\epsilon}\right)$. We can choose arbitrarily small ϵ, where the fraction of time the system spends in undesirable configurations is bounded by $\epsilon\theta_{CRB}$. During this time, state estimation algorithms which perform filtering can rely on odometry and features in the environment until a desirable configuration is recovered.

4.3 Optimization

We can now succinctly formulate an optimization problem.

$$\begin{aligned}
\text{Minimize:} \quad & \overline{Y} && (9)\\
\text{Subject to:} \quad & \overline{\lambda_2} \geq \theta_{\lambda_2} && (10)\\
& \overline{\text{CRB}} \leq \theta_{\text{CRB}} && (11)\\
& D(t) \quad \text{is mean rate stable} && (12)\\
& \alpha(t) \in \mathcal{A}(t) \quad \forall t \in \{0, 1, ...\} && (13)
\end{aligned}$$

where $\overline{Y} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[Y(\tau)]$, $Y(t)$ is the distance to new information (Eq. (2)), and the expectation is over $S(t)$ and $\alpha(t)$. The goal is to choose an action in the discrete, finite action space to minimize the distance to new information subject to constraints on connectivity and localizability, all while keeping the delay queue stable. We assume throughout that limits are well-defined, and analyze behavior in the absence of failures. We have dropped the subscript r for readability throughout this and the following subsection.

Let $\Theta(t) = [D(t), Q(t), Z(t)]$ be a concatenated vector of the queues defined in Eq. (5), (6), and (8). We assume throughout that the initial queue values are 0. We define the Lyapunov function,

$$L(\Theta(t)) = \frac{1}{2}k_q D(t)^2 + \frac{1}{2}k_Q Q(t)^2 + \frac{1}{2}k_Z Z(t)^2 \tag{14}$$

as a scalar indicator of the size of this vector, where k_q, k_Q, and k_Z are weights on the priority of each objective. The one-step conditional Lyapunov drift is given by $\Delta(\Theta(t)) = \mathbb{E}[L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)]$. Greedily minimizing $\Delta(\Theta(t)) + k_Y\mathbb{E}[Y(t)|\Theta(t)]$, the *drift-plus-penalty* expression, will lead to mean rate stability for all queues [93]. Here, $k_Y \geq 0$ is a weight on the priority of exploration.

Substituting our queue dynamics gives

$$\begin{aligned}\Delta(\Theta(t)) + k_Y\mathbb{E}[Y(t)|\Theta(t)] \leq B + k_Y\mathbb{E}[Y(t)|\Theta(t)] \\ +\mathbb{E}[k_q D(t)(q(t) - \theta_d b\mathbb{1}(t)) + k_Q Q(t)(\theta_{\lambda_2} - \lambda_2(t)) \\ +k_Z Z(t)(\mathrm{CRB}(t) - \theta_{\mathrm{CRB}})|\Theta(t)]\end{aligned} \tag{15}$$

where $\mathbb{1}(t)$ indicates a communication path exists for this robot at time t, and a constant B upper bounds the expectation of

$$\frac{(\theta_{\lambda_2} - \lambda_2(t))^2}{2} + \frac{(\mathrm{CRB}(t) - \theta_{\mathrm{CRB}})^2}{2} + \frac{q_{max}^2}{2} + \frac{(\theta_d b)^2}{2} \tag{16}$$

given $\Theta(t)$ and holds for all t since we have enforced a bound on CRB and the Fiedler value of a graph is bounded.

We opportunistically minimize the expectations on the righthand side of Eq. (15) which allows our algorithm to be online. Each robot r assumes the probabilities of decisions $\alpha_{i\neq r}(t)$ are uniformly distributed, therefore $\mathbb{E}[x_i(t+1)|x_i(t)] = x_i(t)$. At time t, our distributed controller observes $S(t)$, $q(t)$, $D(t)$, $Q(t)$, and $Z(t)$ and chooses $\alpha(t) \in \mathcal{A}(t)$ to minimize

$$\begin{aligned}&k_Y Y(\alpha(t), S(t)) \\ &+k_q D(t)(q(t) - \theta_d b p_{rD}(\alpha(t), S(t)) \\ &+k_Q Q(t)(\theta_{\lambda_2} - \lambda_2(\alpha(t), S(t))) \\ &+k_Z Z(t)(\mathrm{CRB}(\alpha(t), S(t)) - \theta_{\mathrm{CRB}})\end{aligned} \tag{17}$$

where the expectation that a path exists from r to D is p_{rD} and we have reintroduced the inputs $\alpha(t), S(t)$ to emphasize which variables are functions of the decision and state. We have derived a distributed controller which captures each of the four objectives enumerated at the beginning of this section, and can flexibly focus on one or more of these goals at a given time depending on which term dominates Eq. (17).

4.4 Analysis

The problem described by Eq. (9-13) is *feasible* if there exists any arbitrary control policy, or sequence of decisions, which satisfies all three constraints. In this case, the optimal value of $\overline{Y}$ achieved by a one-step local controller can be achieved arbitrarily closely by the controller given by Eq. (17). Intuitively, large values of k_Y cause the controller to approximate the information-theoretic control strategy presented in Sec. 4.2.1 while meeting the time average constraints. Similarly, if the problem is feasible, the constraints can be achieved arbitrarily closely. The proof of this follows directly from the proof given by Neely in Appendix 4.A [93].

4.4.1 Bounded Delay

If $D(t)$ is stable, queuing theory gives that the rate at which $D(t)$ decreases is at least the rate it increases, or $q^{\text{av}} \leq \theta_d b \mathbb{1}^{\text{av}}$, where the notation $x^{\text{av}} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} x(\tau)$ and we assume limits are well-defined [109]. Little's law states that the size of the queue equals the average delay d^{av} times the rate at which the queue increases, or $q^{\text{av}} = d^{\text{av}} I^{\text{av}}$ [93]. Substituting gives $d^{\text{av}} I^{\text{av}} \leq \theta_d b \mathbb{1}^{\text{av}}$. From this we can derive a bound on the average queuing delay $d^{\text{av}} \leq \frac{b\mathbb{1}^{\text{av}}}{I^{\text{av}}} \theta_d$. This states that any controller which stabilizes $D(t)$ achieves an average wait time of data in $q(t)$ within a multiplicative constant of the delay threshold θ_d.

When we restrict our analysis to the simple case of two competing objectives, setting $k_Q = k_Z = 0$, we claim that the controller given by Eq. (17) will result in $b\mathbb{1}^{\text{av}} = I^{\text{av}}$, and prove by contradiction. If $I^{\text{av}} > b\mathbb{1}^{\text{av}}$, then $q(t) > q_{\max}$ for sufficiently large t, which is not possible. If $I^{\text{av}} < b\mathbb{1}^{\text{av}}$, then there exists an alternate controller which achieves a higher I^{av} (i.e. a lower $\overline{Y}$) and yet still maintains queue stability. Equivalently, at some time t an action exists which results in a smaller value of the expression given by Eq. (17). However, by construction our controller minimizes Eq. (17). Therefore $\frac{b\mathbb{1}^{\text{av}}}{I^{\text{av}}} = 1$ and the bound on average wait time of data in $q(t)$ using our controller is in fact θ_d.

4.4.2 Recovering Connectivity

In the simple case of two competing objectives, connectivity will always be recovered. If $q(t) = 0$, our controller will solely minimize $Y(t)$, which will inevitably result in $q(t) > 0$ assuming a path exists to a frontier. When $q(t) > 0$, there exists $\tau > t$ where $\mathbb{1}(\tau) = 1$ because otherwise $D(t)$ would grow unstable. Our controller will recover connectivity in order to stabilize $D(t)$ assuming it is feasible.

4.4.3 Local Optima and Limit Cycles

In general, when $k_Q, k_Z \geq 0$, our queue-stabilizing framework supports additional objectives and we cannot analytically guarantee connectivity is recovered because of the possibility of local optima and limit cycles. If $\alpha_r(t) = x_r(t)$ strictly minimizes Eq. (17), the robot can get stuck. When this occurs we implement the following recovery control strategy: While $\mathbb{E}[f^c_{r,D}] \leq \theta_c$, instead choose $\alpha_r(t) \in \mathcal{A}_r(t)$ to minimize $\|\alpha_r(t) - D\|_{m_r(t)}$. This moves the robot along the shortest path towards the data sink until it is close enough for direct communication, at which point the robot switches back to the nominal controller. We also switch to this recovery controller when $q_r(t) = q_{max}$, so the robot necessarily recovers connectivity when the data queue is full.

When $k_Q > 0$ or $k_Z > 0$, the robot may sacrifice exploration even when the data queue is empty in order to increase connectivity or localizability. In fact, as we increase the gains k_Q and k_Z, the time averages of $\overline{\lambda_{2r}}$ and $\overline{\text{CRB}}_r$ will converge more quickly to θ_{λ_2} and θ_{CRB} [93]. In this case, the robots will converge to or oscillate within configurations which meet the constraints, much like a strictly constrained exploration strategy would [81]. These stable limit cycles can be avoided by loosening the constraints or decreasing k_Q, k_Z at the cost of reduced connectivity and localizability. In the following section, we observe this in simulation and discuss the performance of our queue-stabilizing controller.

4.5 Simulation Experiments

4.5.1 Setup

Fig. 15 shows two simulation environments with obstacles which prevent communication and ranging. We assume robots start in the vicinity of the data sink and can move into adjacent free cells. Robots have a finite sensing radius of 1.5 with perfect sensing ($Z_i = X_i$) and an expected transmit radius $d_{\theta_c} = 4$ (all units in robot widths). Accurate localization is key to exploration and in the absence of GPS, beacons, or easily detectable landmarks in the environment, relative localization performance is important. We use the Cramer-Rao bound to evaluate uncertainty of relative localization, but for simplicity assume perfect/deterministic movement in simulation. We assume $Pr(X_i = 0) = Pr(X_i = 1)$, so the number of cells visited is the reduction in map entropy. We use $\gamma = 100$, $\theta_d = 10$, $\theta_{\text{CRB}} = 200$, and $\theta_{\lambda_2} = 0.1$. Source code, other parameter values, and accompanying videos for all experiments are available on GitHub [110].

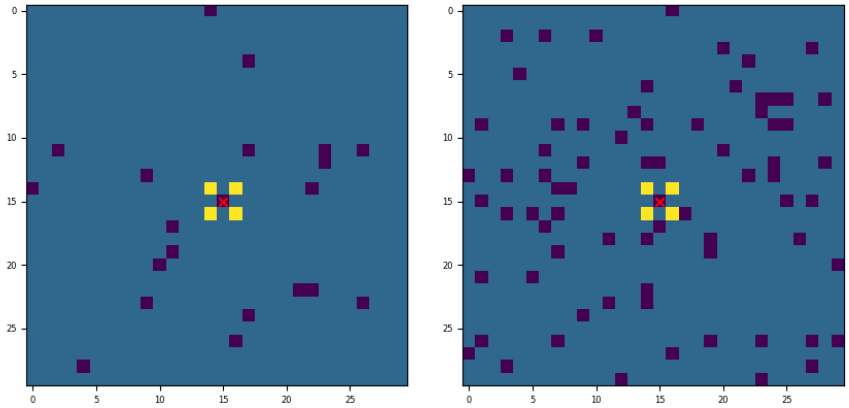

Figure 15: Simulated environments with obstacles (purple), robots (yellow), and data sink (red). **Left:** 4 robots in 30x30 space with 20 obstacles. **Right:** With 75 obstacles. Additional experiments considered 8 robots and 50x50 space, and results are summarized in Table 4.

4.5.2 Compared Approaches

We introduce two simple approaches and two competitive approaches and compare these to our queue-stabilizing (QS) controller.

Unconstrained/information-theoretic (UN): Robots choose $\alpha_r(t) \in \mathcal{A}_r(t)$ to minimize $Y(\alpha_r(t))$. This is a purely information-theoretic controller.

Strictly Constrained (SC): Robots limit their action space to $\mathcal{A}_r(t) = \{\alpha_r(t) : \text{CRB}_r(\alpha_r(t)) \geq \theta_{\text{CRB}}$ and $\lambda_{2r}(\alpha_r(t)) \geq \theta_{\lambda_2}\}$. Then robots choose $\alpha_r(t) \in \mathcal{A}_r(t)$ to minimize $Y(\alpha_r(t))$, and do not move if $\mathcal{A}_r(t)$ is empty.

Time Preference (TP): This controller presented by Spirin *et al.* uses a target ratio ρ comparing the queue to the map size [91]. If $1 - q_r(t)/|m_r(t)| \geq \rho$, where $|m_r(t)|$ is the number of visited cells, robots choose $\alpha_r(t) \in \mathcal{A}_r(t)$ to minimize $Y(\alpha_r(t))$. Otherwise robots choose $\alpha_r(t) \in \mathcal{A}_r(t)$ to minimize $\|\alpha_r(t) - D\|_{m_r(t)}$. In the absence of a direct communication path to the data sink, robots transfer the contents of $q_r(t)$ to a neighboring robot i if $f^c_{r,i} > \theta_c$ and i is closer to D than r. To provide a fair comparison, we implement this relaying for $q_r(t)$ and $D_r(t)$ in our QS approach as well.

Multi-objective (MO): We modify the approach of Benavides *et al.* [92], which weighs connectivity against exploration, to consider our objectives and be suited to local decisions. Robots choose $\alpha_r(t) \in \mathcal{A}_r(t)$ to minimize $w_1 Y(\alpha_r(t)) - w_2 p_{rD}(\alpha(t)) - w_3 \lambda_2(\alpha(t)) + w_4 \text{CRB}_r(\alpha(t))$, where $w_1, w_2, w_3, w_4 > 0$ are priority weights.

4.5.3 Results

Coverage: Fig. 16 shows the average map size at the data sink over time for the set of parameters which maximizes final coverage (solid lines) for each approach in the world shown in Fig. 15 (left). QS outperforms UN, SC, and MO and slightly outperforms TP with respect to map coverage. We depict QS here with $k_Q = k_Z = 0$, i.e., only the delay queue is stabilized. This causes QS and TP to have similar emergent behavior [91], and allows us to closely compare our contribution to the state of the art. One subtle difference between the two is that QS maximizes p_{rD}, explicitly considering the location of our neighbors and taking multi-hop communication into account, whereas TP simply minimizes $\|\alpha_r(t) - D\|_{m_r(t)}$. This is likely why QS performs better. Note that as we increase ρ, k_q (Fig. 16, dashed lines), we see smoother curves caused by incremental data transfer. In cluttered and larger environments where communication is severely limited, increasing connectivity to allow incremental data transfer significantly improves performance relative to UN (by up to 99% and 88% respectively). This finding emphasizes that the choice of k_q, k_Q, k_Z, k_Y which maximizes coverage depends on characteristics of the environment (see Table 4).

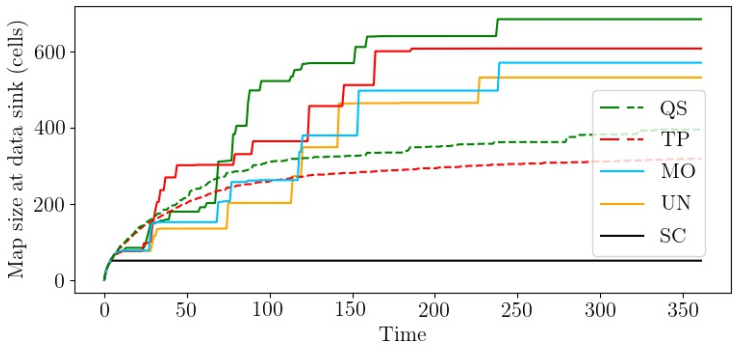

Figure 16: Map cells known at data sink over time for each approach in 4.5.2, averaged over six independent trials, with the following parameters: **QS (our novel approach):** Solid - $k_q = 0.005, k_Q = 0, k_Z = 0, k_Y = 100$, Dashed - $k_q = 1000, k_Q = 0, k_Z = 0, k_Y = 100$, **TP:** Solid - $\rho = 0.35$, Dashed - $\rho = 0.99$, **MO:** $w_1 = 10, w_2 = 0.1, w_3 = 0.1, w_4 = 100$.

Localizability: In Fig. 17, we plot localizability against coverage. We use the time spent below the uncertainty constraint as a localizability metric. Since we have bounded the CRB, this is a more meaningful metric than average CRB but follows the same trend. We see that high localizability can hurt coverage, as discussed in Sec. 4.4.3. QS can improve localizability over TP and UN without sacrificing coverage, and improves coverage over SC without sacrificing localizability. MO can achieve the highest localizability (when minimizing CRB is the sole or primary objective), but suffers with respect to coverage from not adapting to the amount of untransferred data.

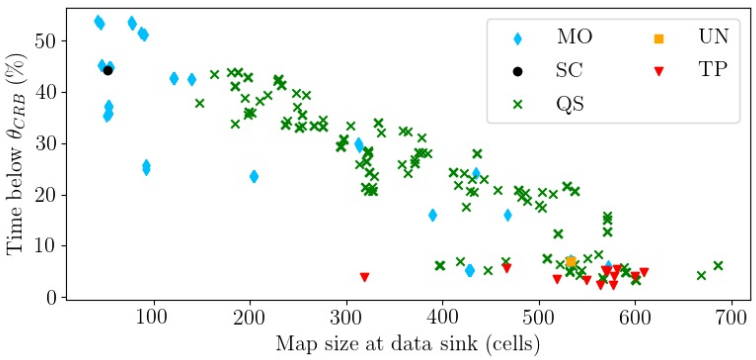

Figure 17: Average percent of time spent with CRB below θ_{CRB} per robot against number of map cells known at data sink for 4 robots in 30x30 space with 20 obstacles. Each point is the average performance of 6 trials, and ideal is at the top right. We show more points for the more parameterized approaches (MO and our novel QS), while SC and UN are captured by a single point because they do not depend on parameters.

Role of virtual queues: Actions which stabilize $D_r(t)$ and $Q_r(t)$ both result in increased connectivity. In Fig. 18 we plot average λ_{2D} against map coverage, noting which of these virtual queues was stabilized. Interestingly, here connectivity appears more correlated with k_q than k_Q. This indicates that stabilizing the delay queue, which explicitly consider multi-hop communication, does encourage overall network connectivity. We also plot localizability against coverage, noting whether $Z_r(t)$ is stabilized. As expected, $k_Z > 0$ is highly correlated with localizability performance.

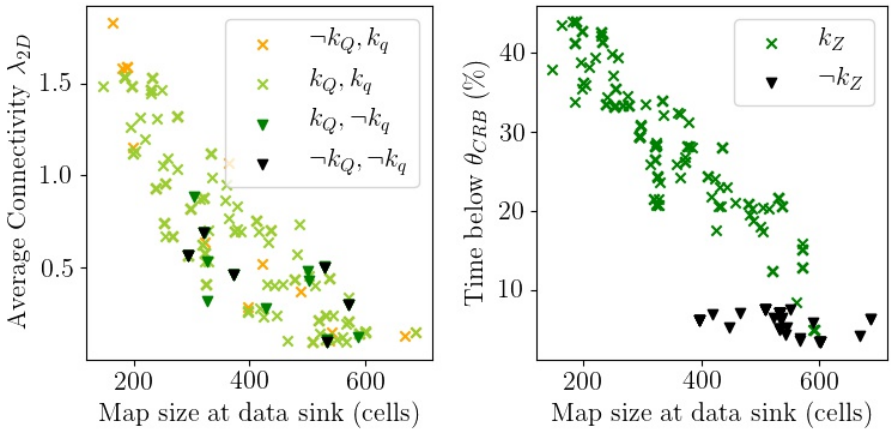

Figure 18: QS connectivity and localizability performance with and without stabilizing the relevant virtual queues. Left: Average λ_{2D} against coverage. Right: Average percent of time spent with CRB below θ_{CRB} per robot against coverage. Notation $\neg k_Q$ indicates $k_Q = 0$, and in both plots ideal is at the top right.

4.6 Summary

In this section we have presented a novel distributed controller for multi-robot exploration which uses ideas from queue-based stochastic network optimization to autonomously decide at each time step, based on the current state of the system, how to weigh network reliability, connectivity, localizability, and information gain. We have demonstrated that this controller can achieve better coverage than the state of the art, reduce localization uncertainty, and result in desirable emergent

behavior. We have also tested our queue-stabilizing controller in a cluttered environment, and an environment with much larger area. For the latter, we loosened the delay bound to $\theta_D = 50$. Our findings comparing the best coverage achieved by QS relative to TP, the leading competitive approach, are briefly summarized in Table 4.

Environment			TP: Max Coverage		QS: Max Coverage				
size	# robots	# obstacles	ρ	coverage	k_q	k_Q	k_Z	k_Y	coverage
30x30	4	20	0.35	608	0.005	0	0	100	685
30x30	8	20	0.5	786	0.005	250	0	100	813
30x30	4	75	0.7	602	0.01	5	0	100	635
50x50	4	50	0.85	689	0.005	125	0	100	691

Table 4: Summary of results: Maximum Coverage

These simulated results support our claim that explicitly accounting for localization performance improves results and intentionally intermittent connectivity can improve exploration coverage while still meeting constraints on queuing delay. However our simulations make an unrealistic assumption about deterministic movement and do not consider the nuance of realistic routing protocols. Additionally, we limit our consideration to a theoretical bound on location uncertainty. Integrating a specific localization and mapping algorithm which leverages collaborative localization, like TEAM, and other future research directions are discussed in the following section.

5 Proposed Future Work

In this section, we present several directions for future research. The first is a natural extension of our previous two studies, the second is an effort to move from fully distributed algorithms to increased collaboration, and the third direction seeks to turn the limitations of ultra-wideband into an advantage.

5.1 Queue-stabilizing Exploration System Implementation

We have previously considered localization and mapping as a separate problem from that of exploration. However, the findings from our second study indicate that exploring while optimizing for localization, a *localization-performance-in-the-loop* approach, has merit. It is therefore a natural direction for future work to integrate the efforts of our two previous studies. The objective of this study is to experimentally evaluate queue-stabilizing exploration on our network of Turtlebot3 Burgers.

Several aspects of our multi-robot system were abstracted in simulation, and should be addressed in detail. While we have demonstrated the feasibility of TEAM, our primary objective is no longer low-complexity and therefore we will consider other methods of collaborative localization and mapping. Several prior works have presented improvements to individual SLAM with UWB data (from fixed anchors) [44, 111, 37]. Unfortunately, available open source SLAM implementations do not inherently accept UWB input. Specifically, Cartographer [112], a graph-based SLAM approach, is a desirable choice as it combines local map building with global scan matching optimization. Modifying the global backend of Cartographer to integrate constraints in the form of inter-robot range measurements is one potential direction for future work. The alternative is to rely on an Extended Kalman Filter (EKF) which incorporate both LiDAR and UWB measurements

in the update step [113]. Ease of extracting the uncertainty, which is important feedback for our exploration algorithm, is one advantage of this approach.

Similarly, in our simulations we have used an idealistic flooding-based routing protocol. For our preliminary experimentation, we have selected Optimized Link State Routing for multi-hop ad hoc communications [103] for ease of implementation via the OLSRd library [105, 106]. One advantage of this protocol is that uses, and conveniently exposes, expected retransmissions as a metric for determining routes, which is closely related to the network reliability metric. Like uncertainty, these observations will serve as feedback for our exploration algorithm. Alternative routing protocols which are best suited for robotic wireless network exist [71], and evaluating their use for our application is another direction for consideration.

Notably, the exploration algorithm we presented in Sec. 4 considers a discrete decision space and discrete time decisions. Our differential drive mobile robots operate in continuous space, and addressing this disconnect will require one of two paths forward: we can continue to treat space as discretized and use waypoint navigation, or we can re-formulate the optimization problem in Eq. (9-13). If we consider renewal frames rather than time steps, where frames have variable durations and depend on the decisions made, we can derive a dynamic policy. Generally, this renewal-theoretic model allows a larger class of problems to be treated, including Markov Decision Problems [93].

Careful consideration of each of these design choices (localization algorithm, routing protocol, exploration algorithm extended to continuous space) will lead to a system architecture for our Turtlebot3 Burgers. We will evaluate this system experimentally to determine how noise in the sensor model, the odometry, and the ranging measurements affect the performance of our exploration algorithm.

5.2 Exploration Efficiency and Localization Accuracy Performance Gains via Increased Collaboration

Throughout this thesis proposal, we have considered purely distributed algorithms. Our exploration framework allows each robot to make decisions independent of other robots, except for knowledge of the location estimates. Our localization and mapping algorithm allows each robot to calculate its location estimate based on only information available from its three nearest neighbors, including ranging information and location estimates. In this section focus on several problem spaces which arise when we consider scaling these solutions to larger networks of robots. Namely,

- as robots rely further on data relaying can we incorporate a delay-tolerant paradigm such that robots predict/account for relay opportunities, in order to achieve better exploration coverage?
- with the increased quantity of UWB inter-robot range measurements, can robots improve their location estimates by leveraging all available measurements?
- as the network scales and medium access becomes a greater challenge, can we design a scheduler and anchor selection mechanism which improves localizability?

To answer each of these questions, we propose moving away from the fully distributed setting by increasing the amount of information shared between robots. In each case there will be a tradeoff between the potential performance gains and the additional communication overhead, and we intend to study this trade.

5.2.1 From Queues to Queuing Networks

In our initial problem formulation, each robot seeks to stabilize its queue. Arrivals are predicted given the expected information gain, and service is predicted based on connectivity with the data sink. However multi-hop communication means that arrivals depend on a robot's neighbor's queue size as well. From a centralized perspective, we can model the multi-robot system as a queuing network (see Fig. 19).

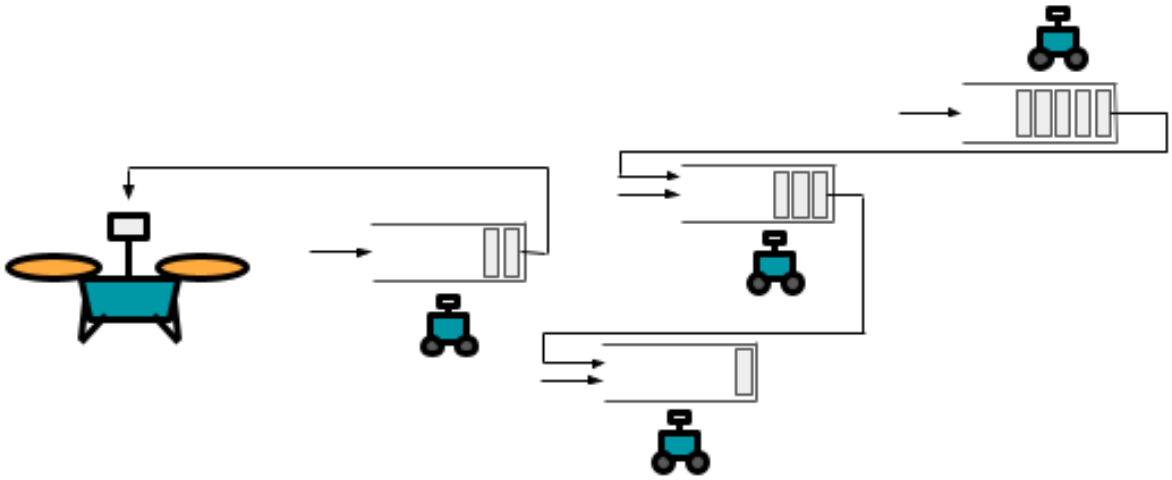

Figure 19: Multi-robot exploration modeled as a queuing network

Relative to a purely distributed approach, we hypothesize that a centralized algorithm with full knowledge of the network would improve exploration efficiency subject to queue stability. But given that our system leverages intermittent connectivity, centralized solutions are not feasible. It may be possible to improve exploration efficiency subject to queue stability by allowing each robot i to share with its neighbors its queue states at time t: $q_i(t)$, $D_i(t)$, $Q_i(t)$, $Z_i(t)$. As we already assume robots share their map information, this introduces very little overhead.

5.2.2 From Trilateration to Network Localization

TEAM uses trilateration and is not robust to missing UWB measurements between robots, which could be the case when radio frequency-impermeable obstacles exist. As the number of robots increases, it may be possible to overcome the challenge of missing measurements via network localization algorithms. This would require making the measurements available to the entire network, or sharing position estimates with the entire network through an iterative approach [60, 61].

We have considered an analogous problem estimating the positions of mobile devices based on Bluetooth Received Signal Strength Indication (RSSI) measurements. Preliminary results on real and simulated data indicate that network localization algorithms can improve inter-device distance estimation beyond directly estimating distance from RSSI. Beyond this, network localization algorithms can estimate distances for device-device pairs when no signal is received, deducing positions from other available measurements. We evaluated several approaches from the sensor network and manifold learning literature on a network of 11 devices. The ground truth locations and RSSI measurements were recorded for a previous work on radio frequency localization [114]. Fig. 20

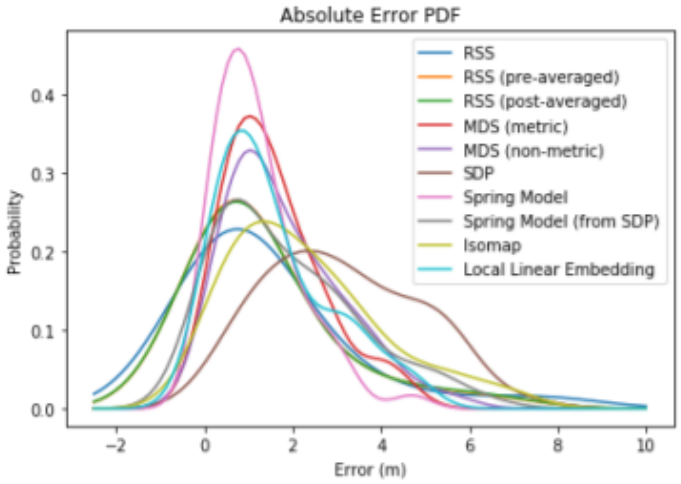

Figure 20: Probability distribution of errors (m) for inter-device estimation using several network localization approaches, fitted with Gaussian kernel density estimation.

shows the likelihood of errors for each approach, and indicates that the spring model can improve distance estimation and therefore relative localization performance.

The spring model uses all available measurements to solve the problem iteratively in a distributed manner (as in, each node performs one step of iteration using only the measurements between itself and its neighbors) to minimze the stress on each device,

$$\text{Stress}_{\text{SM}}(x_i) = F(x_i) = \sum_{j:(i,j)\in\mathcal{N}} f(x_i, x_j) \tag{18}$$

$$f(x_i, x_j) = \frac{\|x_i - x_j\| - d_{ij}}{\sigma} \langle \frac{x_j - x_i}{\|x_i - x_j\|} \rangle \tag{19}$$

where we refer to stress as $F(x_i)$ as it is the sum of forces applied to node i by each node j. These forces, $f(x_i, x_j)$, are analogous to the forces that would be applied if each node-node distance were a stretched or compressed spring. Note that these forces are vectors in the direction of $\langle x_j - x_i \rangle$, and the resulting stress is also a vector. In equation (19), d_{ij} is the distance measurement between i and j and σ is the standard deviation of noise. Scaling the stress by this term allows distances with less uncertainty (typically shorter distances) to have a greater impact on the solution than distances with greater uncertainty.

Our spring model implementation is described in Algorithm 2, where n is the number of nodes, $\mathcal{N}$ is the set of node-node distances, ϵ is a parameter which defines convergence (we use 0.1), and γ is a coefficient applied to the force similar to a step size. To prevent the algorithm from diverging, we choose a step size of roughly $\frac{1}{n}$. We can optionally decrease γ over time for an adaptive step size, or update $x_i \leftarrow x_i + \gamma(F^t + F^{t-1})$ to include momentum in the gradient descent.

Algorithm 2: Spring model for network localization

input : $n, \mathcal{N}, \epsilon, \gamma$
output: $\mathcal{X}$
initialize $\mathcal{X}$ randomly in $\mathcal{R}^2$
for *iteration* t **do**
  for *node* i **do**
    $F = 0$
    for *node* j **do**
      if $(i, j) \in \mathcal{N}$ **then**
        $F \leftarrow F + f(x_i)$
      end
    end
    $x_i \leftarrow x_i + \gamma F$
  end
  stop if $\left\|x_i^t - x_i^{t-1}\right\| \leq \epsilon \forall i \in \{1, ...n\}$
end

Our preliminary results indicate that augmenting TEAM with our spring model can lead to better inter-robot ranging and localization accuracy, particularly when some robots can only connect with one or two other robots.

5.2.3 Optimized TDMA and Anchor Selection

As we increase the number of robots, the key challenge for UWB two-way ranging (TWR) is medium access, in our case the careful design of a TDMA scheduling algorithm. Recently, Cao *et al.* presented a novel distributed TDMA algorithm maximizes channel usage and supports multi-hop networks, where slots can be used by nodes are two hops apart [69]. They use rapid time slot scheduling and update iteratively, to account for dynamic network nodes. Because this process requires exchanging additional information for coordinating time slot usage, we consider this to an instance of using more collaboration to improve performance. The algorithm proposed by these authors extends approximate solutions to the vertex coloring problem in graph theory, which is known to be NP-complete. They also propose an algorithm for network localization which runs on each network node and involves selecting the optimal anchors to define a coordinate system.

We propose an extension of this work which considers anchor selection and the resulting multilateration performance as the optimization objective to the TDMA scheduling algorithm. Rather than maximizing UWB channel usage to implicitly improve localization performance, we propose maximizing estimated localization accuracy, using the Cramer-Rao bound as a relevant metric as we have done in Sec. 4. Because the network configuration changes quickly for our application, the unavoidable conflicts from a new node joining a two-hop cluster (which are resolved in one frame) may cause significant performance degradation [69]. Addressing this concern is one possible direction for future work.

Additionally, anchor selection (or more specifically instantaneous anchor select) has auxiliary consequences for our system. Localization may fail if a robot is moving quickly with respect to the TDMA cycle [69], therefore we elect to keep anchors stationary. This means nodes performing ranging sacrifice mobility, in turn sacrificing exploration. Studying the interdependence of medium access and anchor selection on localization and mapping accuracy and exploration is a direction for proposed future work.

5.3 UWB multipath-assisted localization

UWB technology is characterized by very large bandwidth and thus time resolution on the order of nanoseconds and high material penetrability. Despite this, multipath propagation and non-line-of-sight conditions can make using UWB measurements challenging. Multipath components (MPCs) originate from places where the signal bounced, indicating a surface or obstacle. Equipped with this knowledge, researchers have recently proposed taking advantage of the position-related information contained in MPCs [115]. It is not likely that this information could generate complete maps of unknown environments. However, if at least partial maps exist, Kulmer *et al.* [13] have demonstrated the feasibility of localizing two nodes based on only UWB, without fixed anchors, odometers, or inertial measurement units.

This exciting advancement opens up several possibilities. For example, under strict assumptions (e.g. four planar walls) it may be possible for two or more robots to determine their locations along with the room dimensions which are not known *a priori*. Or for lunar subsurface exploration and other environments where dust accumulation is a significant challenge for the long-term operation of light-based sensors, MPCs may provide a backup option for absolute positioning near the end of the mission. In this case, the objective may be purely navigation, for example to return to a base station or charging station, and designing trajectories which best leverage MPCs could be a direction for future research.

6 Conclusion

The following table presents our anticipated timeline for the next two years.

Objective	Completion date
Framework evaluation	October 2021
More Collaboration: Queueing network	March 2022
More Collaboration: Network localization and anchor selection	October 2022
UWB multipath-assisted localization and mapping	March 2023
Thesis	May 2023

Table 5: Summary of results: maximum coverage

In this proposal we investigate the design, implementation, and evaluation of distributed algorithms for networked multi-robot exploration which optimize for collaborative localization performance and timely data transer. Our proposed solutions extend the collective knowledge in robotic networks and will help shape future space technology for planetary and lunar exploration.

Acknowledgements

This research is supported by NASA Space Technology Research Fellowship No. 80NSSC19K1189.

References

[1] N. R. Council, *Vision and Voyages for Planetary Science in the Decade 2013-2022*. Washington, DC: The National Academies Press, 2011. [Online]. Available: https://www.nap.edu/catalog/13117/vision-and-voyages-for-planetary-science-in-the-decade-2013-2022

[2] L. E. A. Group, "Advancing science of the moon: Report of the specific action team," Houston, Texas, United States of America, August 2017. [Online]. Available: https://www.lpi.usra.edu/leag/reports/ASM-SAT-Report-final.pdf

[3] R. Greeley, "Lava tubes and channels in the lunar marius hills," *The Moon*, vol. 3, no. 3, pp. 289–314, 1971.

[4] F. Horz, "Lava tubes-potential shelters for habitats," in *Lunar bases and space activities of the 21st century*, 1985, pp. 405–412.

[5] W. Whittaker, "Technologies enabling exploration of skylights, lava tubes and caves," *NASA, US, Report, no. NNX11AR42G*, 2012.

[6] E. J. Wyatt, K. Belov, J. Castillo-Rogez, S. Chien, A. Fraeman, J. Gao, S. Herzig, T. Lazio, M. Troesch, and T. Vaquero, "Autonomous networking for robotic deep space exploration," in *Int. Symposium on AI, Robotics, and Automation for Space (ISAIRAS)*, 2018.

[7] T. Vaquero, M. Troesch, and S. Chien, "An approach for autonomous multi-rover collaboration for mars cave exploration: Preliminary results," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS 2018). Also appears at the ICAPS PlanRob*, 2018.

[8] Y. Cao, C. Yang, R. Li, A. Knoll, and G. Beltrame, "Accurate position tracking with a single uwb anchor," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2344–2350.

[9] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision agriculture*, vol. 13, no. 6, pp. 693–712, 2012.

[10] J. Banfi, A. Q. Li, N. Basilico, I. Rekleitis, and F. Amigoni, "Asynchronous multirobot exploration under recurrent connectivity constraints," in *2016 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2016, pp. 5491–5498.

[11] M. Tatum, "Communications coverage in unknown underground environments," Masters's thesis, The Robotics Institute, Carnegie Mellon University, USA, 2020. [Online]. Available: https://www.ri.cmu.edu/wp-content/uploads/2020/07/mtatum_thesis_final.pdf

[12] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.

[13] V. Kumar, D. Rus, and G. S. Sukhatme, "Networked robots," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. 41, pp. 943–958.

[14] E. Stump, A. Jadbabaie, and V. Kumar, "Connectivity management in mobile robot teams," in *2008 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2008, pp. 1525–1530.

[15] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error," in *Int. Joint Conf. AI (IJCAI)*, vol. 15. LAWRENCE ERLBAUM ASSOCIATES LTD, 1997, pp. 1340–1345.

[16] Y. Cao, M. Li, I. Švogor, S. Wei, and G. Beltrame, "Dynamic range-only localization for multi-robot systems," *IEEE Access*, vol. 6, pp. 46 527–46 537, 2018.

[17] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.

[18] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[19] G. A. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 967–973, 2012.

[20] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[21] ——, "Gmapping," OpenSLAM, 2007. [Online]. Available: https://openslam-org.github.io/gmapping.html

[22] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. 1997 IEEE Int. Symposium on Computational Intelligence in Robot. and Autom. (CIRA)*. IEEE, 1997, pp. 146–151.

[23] ——, "Frontier-based exploration using multiple robots," in *Proc. 2nd Int. Conf. Autonomous agents*, 1998, pp. 47–53.

[24] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1375–1392, 2014.

[25] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *Int. Journal Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.

[26] M. Corah and N. Michael, "Efficient online multi-robot exploration via distributed sequential greedy assignment." in *Robotics: Science and Systems (RSS)*, vol. 13, 2017.

[27] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[28] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Decentralised monte carlo tree search for active perception," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 864–879.

[29] S. Thrun, "Monte carlo pomdps." in *Advances in Neural Information Processing Systems*, vol. 12, 1999, pp. 1064–1070.

[30] T. Kollar and N. Roy, "Trajectory optimization using reinforcement learning for map exploration," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 175–196, 2008.

[31] E. Yang and D. Gu, "Multiagent reinforcement learning for multi-robot systems: A survey," tech. rep, Tech. Rep., 2004.

[32] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed multi-robot coordination in area exploration," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 945–955, 2006.

[33] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 2002, pp. 540–545.

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

[35] I. Rekleitis, G. Dudek, and E. Milios, "Multi-robot collaboration for robust exploration," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 7–40, 2001.

[36] R. Kurazume and S. Hirose, "An experimental study of a cooperative positioning system," *Autonomous Robots*, vol. 8, no. 1, pp. 43–52, 2000.

[37] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 547–568, 2014.

[38] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE signal processing magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[39] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, p. 707, 2016.

[40] "Pozyx creator system." [Online]. Available: https://www.pozyx.io/

[41] Y. Zhou, "An efficient least-squares trilateration algorithm for mobile robot localization," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3474–3479.

[42] W. Shule, C. M. Almansa, J. P. Queralta, Z. Zou, and T. Westerlund, "Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems," *Procedia Computer Science*, vol. 175, pp. 357–364, 2020.

[43] T. M. Nguyen, A. H. Zaini, K. Guo, and L. Xie, "An ultra-wideband-based multi-uav localization system in gps-denied environments," in *2016 International Micro Air Vehicles Conference*, 2016.

[44] C. Wang, H. Zhang, T.-M. Nguyen, and L. Xie, "Ultra-wideband aided fast localization and mapping system," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1602–1609.

[45] T.-M. Nguyen, T. H. Nguyen, M. Cao, Z. Qiu, and L. Xie, "Integrated uwb-vision approach for autonomous docking of uavs in gps-denied environments," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9603–9609.

[46] K. Guo, X. Li, and L. Xie, "Ultra-wideband and odometry-based cooperative relative localization with application to multi-uav formation control," *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2590–2603, 2019.

[47] T.-M. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao, and L. Xie, "Distance-based cooperative relative localization for leader-following control of mavs," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3641–3648, 2019.

[48] S. Cao, Y. Zhou, D. Yin, and J. Lai, "Uwb based integrated communication and positioning system for multi-uavs close formation," in *2018 International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)*. Atlantis Press, 2018, pp. 475–484.

[49] L. M. R. Peralta, "Collaborative localization in wireless sensor networks," in *Proceedings of the 2007 International Conference on Sensor Technologies and Applications*, 2007, pp. 94–100.

[50] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[51] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, 2006.

[52] H. Cai, V. W. Zheng, and K. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 09, pp. 1616–1637, 9 2018.

[53] J. Bachrach and C. Taylor, "Localization in sensor networks," *Handbook of sensor networks: Algorithms and Architectures*, vol. 1, pp. 277–289, 2005.

[54] A. J. Izenman, "Introduction to manifold learning," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 5, pp. 439–446, 2012.

[55] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[56] F. Young, "Multidimensional scaling, encyclopedia of statistical sciences (volume 5), kotz-johnson, ed," 1985.

[57] "Distance, similarity, and multidimensional scaling," https://pages.mtu.edu/ shanem/psy5220/daily/Day16/ Mar. 2019.

[58] "Manifold learning," https://scikit-learn.org/stable/modules/manifold.html, 2019.

[59] J. De Leeuw, "Applications of convex analysis to multidimensional scaling," 2005.

[60] C. Savarese, J. M. Rabaey, and J. Beutel, "Location in distributed ad-hoc wireless sensor networks," in *2001 IEEE international conference on acoustics, speech, and signal processing. proceedings (Cat. No. 01CH37221)*, vol. 4. IEEE, 2001, pp. 2037–2040.

[61] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 340–341.

[62] J. Le Ny and S. Chauvière, "Localizability-constrained deployment of mobile robotic networks with noisy range measurements," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 2788–2793.

[63] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems," *Int. Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2015.

[64] H. Wang, Y. Guo, and Z. Dong, "Graph rigidity control of mobile robot networks," in *IEEE Int. Conf. Control & Automation (ICCA)*. IEEE, 2010, pp. 2218–2223.

[65] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. 2nd Int. Conf. Embedded Networked Sensor Systems*, 2004, pp. 50–61.

[66] A. A. Kannan, B. Fidan, G. Mao, and B. D. Anderson, "Analysis of flip ambiguities in distributed network localization," in *2007 Information, Decision and Control*. IEEE, 2007, pp. 193–198.

[67] M. Hernandez, A. Marrs, N. Gordon, S. Maskell, and C. Reed, "Cramér-rao bounds for non-linear filtering with measurement origin uncertainty," in *Proc. 5th Int. Conf.Information Fusion*, vol. 1. IEEE, 2002, pp. 18–25.

[68] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[69] Y. Cao, C. Chen, D. St-Onge, and G. Beltrame, "Distributed tdma for mobile uwb network localization," *IEEE Internet of Things Journal*, 2021.

[70] J.-P. Sheu, C.-M. Chao, W.-K. Hu, and C.-W. Sun, "A clock synchronization algorithm for multihop wireless ad hoc networks," *Wireless Personal Communications*, vol. 43, no. 2, pp. 185–200, 2007.

[71] P. Ghosh, A. Gasparri, J. Jin, and B. Krishnamachari, "Robotic wireless sensor networks," in *Mission-Oriented Sensor Networks and Systems: Art and Science*. Springer, 2019, pp. 545–595.

[72] M. S. I. M. Zin and M. Hope, "A review of uwb mac protocols," in *2010 sixth advanced international conference on telecommunications*. IEEE, 2010, pp. 526–534.

[73] L. Oliveira, L. Almeida, and P. Lima, "Multi-hop routing within tdma slots for teams of cooperating robots," in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*. IEEE, 2015, pp. 1–8.

[74] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Transactions on parallel and distributed systems*, vol. 21, no. 5, pp. 631–640, 2009.

[75] A. Campbell and A. S. Wu, "Multi-agent role allocation: issues, approaches, and multiple perspectives," *Autonomous agents and multi-agent systems*, vol. 22, no. 2, pp. 317–355, 2011.

[76] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[77] Y. Kantaros and M. M. Zavlanos, "Communication-aware coverage control for robotic sensor networks," in *53rd IEEE Conf. Decision and Control (CDC)*. IEEE, 2014, pp. 6863–6868.

[78] A. Muralidharan and Y. Mostofi, "Path planning for a connectivity seeking robot," in *2017 IEEE Globecom Workshops*. IEEE, 2017, pp. 1–6.

[79] C. Dixon and E. W. Frew, "Maintaining optimal communication chains in robotic sensor networks using mobility control," *Mobile Networks and Applications*, vol. 14, no. 3, pp. 281–291, 2009.

[80] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *Int. Journal Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[81] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, 2007.

[82] M. O. Ball, C. J. Colbourn, and J. S. Provan, "Network reliability," *Handbooks in Operations Research and Management Science*, vol. 7, pp. 673–762, 1995.

[83] L. Fratta and U. Montanari, "A boolean algebra method for computing the terminal reliability in a communication network," *IEEE Trans. Circuit Theory*, vol. 20, no. 3, pp. 203–211, 1973.

[84] C. Lucet and J.-F. Manouvrier, "Exact methods to compute network reliability," in *Statistical and Probabilistic Models in Reliability*. Springer, 1999, pp. 279–294.

[85] J. A. Abraham, "An improved algorithm for network reliability," *IEEE Trans. Reliab.*, vol. 28, no. 1, pp. 58–61, 1979.

[86] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The laplacian spectrum of graphs," *Graph Theory, Combinatorics, and Applications*, vol. 2, no. 871-898, p. 12, 1991.

[87] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2017.

[88] V.-C. Pham and J.-C. Juang, "A multi-robot, cooperative, and active slam algorithm for exploration," *Int. Journal of Innovative Comp., Inf. and Control*, vol. 9, no. 6, pp. 2567–2583, 2013.

[89] K. Cesare, R. Skeele, S.-H. Yoo, Y. Zhang, and G. Hollinger, "Multi-uav exploration with limited communication and battery," in *2015 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2015, pp. 2230–2235.

[90] J. De Hoog, S. Cameron, and A. Visser, "Autonomous multi-robot exploration in communication-limited environments," in *Proc. Conf. Towards Autonomous Robotic Systems*. Citeseer, 2010, pp. 68–75.

[91] V. Spirin, S. Cameron, and J. De Hoog, "Time preference for information in multi-agent exploration with limited communication," in *Conference Towards Autonomous Robotic Systems*. Springer, 2013, pp. 34–45.

[92] F. Benavides, C. Ponzoni Carvalho Chanel, P. Monzón, and E. Grampín, "An auto-adaptive multi-objective strategy for multi-robot exploration of constrained-communication environments," *Applied Sciences*, vol. 9, no. 3, p. 573, 2019.

[93] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[94] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conf. on Decision and Control (CDC)*. IEEE, 1990, pp. 2130–2132.

[95] ——, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, 1993.

[96] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[97] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 279–290.

[98] S. Wang, A. Gasparri, and B. Krishnamachari, "Robotic message ferrying for wireless networks using coarse-grained backpressure control," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 498–510, 2016.

[99] A. Gasparri and B. Krishnamachari, "Throughput-optimal robotic message ferrying for wireless networks using backpressure control," in *2014 IEEE 11th Int. Conf. on Mobile Ad Hoc and Sensor Systems*. IEEE, 2014, pp. 488–496.

[100] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 1250–1257.

[101] J. Hörner, "Map-merging for multi-robot system," Bachelor's thesis, Charles University in Prague, Faculty of Mathematics and Physics, Prague, 2016. [Online]. Available: https://is.cuni.cz/webapps/zzp/detail/174125/

[102] "Subt challenge cretise," Aug 2019. [Online]. Available: https://youtu.be/WW4nVdpkyGg

[103] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," Internet Requests for Comments, RFC Editor, RFC 3626, October 2003, http://www.rfc-editor.org/rfc/rfc3626.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3626.txt

[104] T. Idota, "Quadrotor tunnel nav," May 2019.

[105] A. Tonnesen, T. Lopatic, H. Gredler, B. Petrovitsch, A. Kaplan, and S. Turke, "Olsrd: an ad hoc wireless mesh routing daemon," 2008.

[106] C. Andre, "Raspberry Pi OLSRd Tutorial," Feb. 2020. [Online]. Available: https://github.com/ANRGUSC/Raspberry-Pi-OLSRd-Tutorial

[107] S. Stancliff, J. Dolan, and A. Trebi-Ollennu, "Towards a predictive model of mobile robot reliability," *Tech. Rep. CMU-RI-TR-05-38*, 2005.

[108] R. K. Williams, A. Gasparri, and B. Krishnamachari, "Route swarm: Wireless network optimization through mobility," in *2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE, 2014, pp. 3775–3781.

[109] D. Xue and E. Ekici, "Delay-guaranteed cross-layer scheduling in multihop wireless networks," *IEEE/ACM Trans. Networking*, vol. 21, no. 6, pp. 1696–1707, 2012.

[110] L. Clark, "Queue-stabilizing exploration," Oct 2020. [Online]. Available: https://github.com/ANRGUSC/queue-stabilizing-exploration

[111] D. Hoeller, A. Ledergerber, M. Hamer, and R. D'Andrea, "Augmenting ultra-wideband localization with computer vision for accurate flight," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 734–12 740, 2017.

[112] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

[113] R. Labbe, *Kalman and Bayesian Filters in Python.* Github, 2020. [Online]. Available: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python

[114] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: a sequence based technique for rf localization in wireless sensor networks," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, 2005, pp. 285–292.

[115] J. Kulmer, E. Leitinger, S. Grebien, and K. Witrisal, "Anchorless cooperative tracking using multipath channel information," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2262–2275, 2018.