A Queue-Stabilizing Framework for Networked Multi-Robot Exploration

Lillian Clark[1], Joseph Galante[2], Bhaskar Krishnamachari[1], and Konstantinos Psounis[1]

***Abstract*—Motivated by planetary exploration, we consider the problem of deploying a network of mobile robots to explore an unknown environment and share information with a stationary data sink. The configuration of robots affects both network connectivity and the accuracy of relative localization. Robots explore autonomously and can store data locally in their queues. When a communication path exists to the data sink, robots transfer their data. Because robots may fail in a non-deterministic manner, causing loss of the data in their queues, enabling communication is important. However, strict constraints on connectivity and relative positions limit exploration. To take a more flexible approach to managing these multiple objectives, we use Lyapunov-based stochastic optimization to maximize new information while using virtual queues to constrain time-average expectations of metrics of interest. These include queueing delay, network connectivity, and localization uncertainty. The result is a distributed online controller which autonomously and strategically breaks and restores connectivity as needed. We explicitly account for obstacle avoidance, limited sensing ranges, and noisy communication/ranging links with line-of-sight occlusions. We use queuing theory to analyze the average delay experienced by data in our system and guarantee connectivity will be recovered when feasible. We demonstrate in simulation that our queue-stabilizing controller can reduce localization uncertainty and achieve better coverage than two state of the art approaches.**

***Index Terms*—Networked robots, multi-robot systems, connectivity maintenance, localization.**

I. INTRODUCTION

ROBOTIC exploration is an integral component of future space missions, with applications including lunar regolith disturbance measurements [1] and planetary surface environment characterization [1], [2], [3]. Many common approaches to exploration are frontier-based [4], [5], while more recent approaches are based in information theory [6], [7]. Networked multi-robot systems provide advantages for exploration in that they can coordinate to improve efficiency [8], cooperate to improve localization with relative range measurements [9], [10], and provide data relaying capabilities to extend the effective exploration range under connectivity constraints [11]. Ensuring the availability of communication links and sufficient relative range measurements for connectivity [11], [12], [13] and localizability [14], [15], [16] are well-researched objectives, and previous work has considered the trade between task-specific goals and increasing connectivity [17], [18] or task-specific goals and reducing location uncertainty [19].

Less well-researched are exploration strategies for applications where intermittent connectivity is sufficient or even intentional, e.g. space exploration [20]. 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. 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.

Several approaches to intermittent connectivity are explored in the literature, and a survey of communication-restricted multi-robot exploration is provided in [21]. Some take a periodic approach [22], others are rendezvous-based [23], and others are role-based and assign connectivity and exploration as distinct tasks [24], [25]. Banfi *et al.* [26] introduce a recurrent connectivity strategy in which a connected deployment configuration is chosen and robots may lose connectivity en route to this configuration. This approach requires centralization, or a high level of overhead.

Benavides *et al.* [27] derive a multi-objective function which weighs discovering new information with maintaining connectivity and adapts to a human operator's input on the importance of each objective. This approach is advantageous in that it does not require specifying roles or deciding rendezvous locations. But this approach assigns a utility to connectivity which does not depend on whether or not there is new information to share. The closest work to ours is the work by Spirin *et al.* [28], which introduces a constraint on the ratio of information at the data sink and a crude approximation of total information across all mobile robots. Based on this constraint, robots choose a role-based controller (*explore* or *return to the data sink*). The authors show desirable emergent behavior, but their controller does not consider the location of neighboring robots so they are limited to relaying opportunistically.

A. Statement of Contributions

Our approach has the advantages of both balancing competing objectives with weights proportional to the specified

Manuscript received: October 15, 2020; Revised December 31, 2020; Accepted January 25, 2021.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by NASA Space Technology Research Fellowship Grant No. 80NSSC19K1189.

[1]Clark, Krishnamachari, and Psounis are with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California. {lilliamc, bkrishna, kpsounis}@usc.edu

[2]Galante is with NASA Goddard Space Flight Center. joseph.m.galante@nasa.gov

Digital Object Identifier (DOI): see top of this page.

importance (as in [27]), and adapting to the amount of new, untransferred data resulting in desirable emergent behavior (as in [28]). We present a rigorous methodology to unify these approaches using concepts from stochastic network optimization theory, which concerns the design of controllers to maintain the stability of queues [29]. The crux of our approach is that a backlog in the data queue puts pressure on the robot to transmit this data and therefore move to recover a communication path to the data sink. Remarkably, our approach is flexible enough to meet constraints on the time-average expectation of any bounded metric (not just data queue size), so long as a feasible solution which satisfies the constraints exists. Thus the same approach may additionally control connectivity and localization uncertainty. In this paper, we present a specific controller designed for these objectives, but the queue-stabilizing framework could generalize to many metrics.

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 [22]. 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.

B. Preliminaries

A Lyapunov function is a scalar function used to analyze stability, and Lyapunov drift is the change in this function over one time step [29]. Lyapunov techniques have a long history in the field of control theory, but the idea of minimizing Lyapunov drift for queue stability in networks was first presented by Tassiulas and Ephremides in [30]. This work resulted in *max-weight scheduling*, an algorithm which stabilizes a network whenever possible and only requires knowledge of the current network state. In brief, the max-weight algorithm observes channel conditions and current queue backlogs, and decides which queue(s) to serve. The decision maximizes a weighted sum of the service rates, where the weights are proportional to a priority coefficient multiplied by the size of the queue. This highlights the critical components of a queue-stabilizing controller: the static priorities decided by the user, the size of the queue which reflects previous decisions made by the system, and the state of the system in that moment. With no knowledge of the future, this controller provably achieves stability, meaning the time-average queue size is bounded. In [31], Neely introduces the *virtual queue* terminology to address problems like maximizing throughput subject to average power constraints.

In our previous works [32], [33], we apply ideas from queue-stability to the related problem of robotic message ferrying, in which mobile robots are controlled to transfer data from stationary sources to stationary sinks. In [33], we present a centralized solution to robot allocation and deployment when data is generated unpredictably at stationary sources. In this work we instead consider the setting where data is the result of exploratory behavior.

II. 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 [10], [9]. 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.

A. Distributed Online Control

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. III-B, III-C, and III-D.

B. 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 $Pr[\text{Robot } r \text{ fails at time } t] = \gamma e^{-\gamma t}$ [34]. 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 [24] the authors propose an exploration strategy which considers remaining battery life.

III. 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.

A. Information-theoretic Exploration

The objective of information-theoretic exploration is to minimize entropy in the map by maximizing the mutual information gained with new sensor measurements [6]. 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 [6]). 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 [19] 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,

$$Y(\alpha_r(t)) = \left\|\alpha_r(t) - x_r^f(t)\right\|_{m_r(t)} \tag{1}$$

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. This will direct the robot towards the closest point which increases information.

B. Stabilizing the Delay Queue via Network Reliability

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 [35] and can be modeled as

$$e_{ij} = \begin{cases} 0 & \text{if non-line-of-sight} \\ \frac{1}{1+e^{\eta(\|x_i - x_j\| - d_{\theta_c})}} & \text{otherwise} \end{cases} \tag{2}$$

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 probability that at least one successful path exists between vertices i and j, which is referred to as the two-terminal network reliability and is known to be NP-hard to compute for a general graph [36], [37]. 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 [38]. Let π_{ij}^k be the event *path k exists* which connects $i \to j$. 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 $i \to j$ 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 [39] 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, and we discuss an approximate heuristic for larger networks in Sec. IV-C.

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

$$q_r(t+1) = \max[q_r(t) + I(\alpha_r(t); m_r(t)) - b\mathbb{1}(\alpha_r(t)), 0] \tag{3}$$

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 [40], we introduce a threshold on the average delay θ_d, and a virtual delay queue with the following dynamics:

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

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. III-E, 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.

C. 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 [11], we introduce a 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 second-smallest eigenvalue of this matrix, λ_2, is known as the Fiedler value and quantifies the algebraic connectivity of the graph [41]. 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. 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:

$$Q_r(t+1) = \max[Q_r(t) + (\theta_\lambda - \lambda_{2r}(\alpha_r(t))), 0]. \tag{5}$$

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, $\overline{\lambda_{2r}} = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}[\lambda_{2r}(\tau)] \geq \theta_\lambda$.

D. 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 [19]. In [14], [15], [16], the authors consider localizability in robotic networks, deriving results from rigidity theory. Briefly, *rigidity* is a property of a graph which means that the edges present are enough to constrain the shape of the graph [42].

However, even rigid graphs can experience ambiguities which lead to challenges in robotic network localization [43]. Therefore we consider the Cramer-Rao bound (CRB) as a metric, which provides a lower bound on the covariance of any unbiased location estimator [44]. The estimator can use received signal strength, time of arrival, or angle of arrival. Moreover, CRB has been shown to accurately predict performance [45] and is closely related to other relevant metrics like the geometric dilution of precision [44]. 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

$$\mathbf{F}_r[i,j] = \frac{1}{(c\sigma_T)^2} \sum_{k \in \mathcal{C}_r \cup D} \frac{(x_{ir} - x_{ik})(x_{jr} - x_{jk})}{{d_k}^2} \tag{6}$$

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 [14].

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, and so following [14] 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:

$$Z_r(t+1) = \max[Z_r(t) + (\text{CRB}_r(\alpha_r(t)) - \theta_{CRB}), 0] \tag{7}$$

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 [43]. To fit our optimization framework, in implementation we enforce $\text{CRB}_r = \min(\text{CRB}_r, \frac{1}{\epsilon})$. 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.

E. Lyapunov Optimization

We can now succinctly formulate an optimization problem.

$$\begin{aligned}
\text{Minimize:} \quad & \overline{Y} && (8)\\
\text{Subject to:} \quad & \overline{\lambda_2} \geq \theta_{\lambda_2} && (9)\\
& \overline{\text{CRB}} \leq \theta_{\text{CRB}} && (10)\\
& D(t) \quad \text{is mean rate stable} && (11)\\
& \alpha(t) \in \mathcal{A}(t) \quad \forall t \in \{0, 1, ...\} && (12)
\end{aligned}$$

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. (1)), 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 subsections.

Let $\Theta(t) = [D(t), Q(t), Z(t)]$ be a concatenated vector of the queues defined in Eq. (4), (5), and (7). We assume throughout that the initial queue values are 0. We define the Lyapunov function,

$$L(\Theta(t)) = \frac{1}{2}k_q D(t)^2 + \frac{1}{2}k_Q Q(t)^2 + \frac{1}{2}k_Z Z(t)^2 \quad (13)$$

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 [29]. Here, $k_Y \geq 0$ is a weight on the priority of exploration.

Substituting our queue dynamics gives

$$\begin{aligned} \Delta(\Theta(t)) + k_Y\mathbb{E}[Y(t)|\Theta(t)] \leq B + k_Y\mathbb{E}[Y(t)|\Theta(t)] \\ +\mathbb{E}[k_q D(t)(q(t) - \theta_d b\mathbb{1}(t)) + k_Q Q(t)(\theta_{\lambda_2} - \lambda_2(t)) \\ +k_Z Z(t)(\text{CRB}(t) - \theta_{\text{CRB}})|\Theta(t)] \end{aligned} \quad (14)$$

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

$$\frac{(\theta_{\lambda_2} - \lambda_2(t))^2}{2} + \frac{(\text{CRB}(t) - \theta_{\text{CRB}})^2}{2} + \frac{q_{max}^2}{2} + \frac{(\theta_d b)^2}{2} \quad (15)$$

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 right-hand side of Eq. (14) 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

$$\begin{aligned} &k_Y Y(\alpha(t), S(t)) \\ &+k_q D(t)(q(t) - \theta_d b p_{rD}(\alpha(t), S(t)) \\ &+k_Q Q(t)(\theta_{\lambda_2} - \lambda_2(\alpha(t), S(t))) \\ &+k_Z Z(t)(\text{CRB}(\alpha(t), S(t)) - \theta_{\text{CRB}}) \end{aligned} \quad (16)$$

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. (16).

F. Analysis

The problem described by Eq. (8-12) 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. (16). Intuitively, large values of k_Y cause the controller to approximate the information-theoretic control strategy presented in Sec. III-A 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 [29].

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 [40]. 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 $q^{\text{av}} = d^{\text{av}} I^{\text{av}}$ [29]. 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. (16) 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. (16). However, by construction our controller minimizes Eq. (16). 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.

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.

G. 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. (16), the robot can get stuck. When this occurs we implement the following recovery control strategy: While $\mathbb{E}[f_{r,D}^c] \leq \theta_c$, 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} [29]. In this case, the robots will converge to or oscillate within configurations which meet the constraints, much like a strictly constrained exploration strategy would [13]. 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.

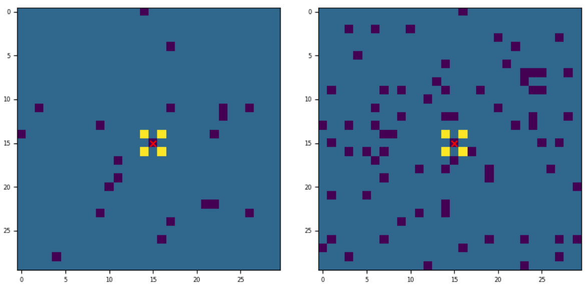

Fig. 1. Simulated environments with obstacles (purple), robots (yellow), and data sink (red). **Left:** 4 robots in 30x30 space with 20 obstacles. **Right:** With 75 obstacles. Additional experiments considered 8 robots and 50x50 space, and results are summarized in Table I.

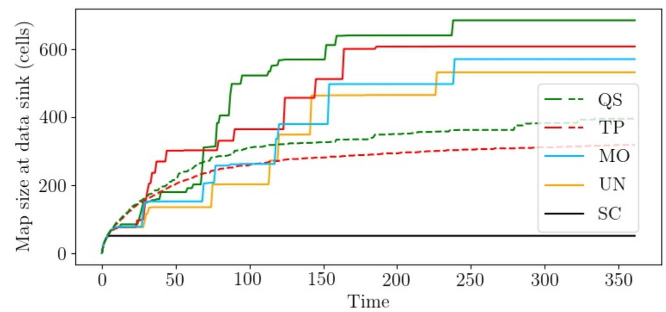

Fig. 2. Map cells known at data sink over time for each approach in IV-A, averaged over six independent trials, with the following parameters: **QS (our novel approach):** Solid - $k_q = 0.005, k_Q = 0, k_Z = 0, k_Y = 100$, Dashed - $k_q = 1000, k_Q = 0, k_Z = 0, k_Y = 100$, **TP:** Solid - $\rho = 0.35$, Dashed - $\rho = 0.99$, **MO:** $w_1 = 10, w_2 = 0.1, w_3 = 0.1, w_4 = 100$.

IV. Performance Results

Fig. 1 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 [46].

A. Compared Approaches

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

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

2) 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.

3) Time Preference (TP): This controller presented by Spirin *et al.* uses a target ratio ρ comparing the queue to the map size [28]. 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.

4) Multi-objective (MO): We modify the approach of Benavides *et al.* [27], 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.

B. Performance Analysis

1) Coverage: Fig. 2 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. 1 (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 [28], 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. 2, 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 I).

2) Localizability: In Fig. 3, 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. III-G. 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.

3) Role of virtual queues: Actions which stabilize $D_r(t)$ and $Q_r(t)$ both result in increased connectivity. In Fig. 4 we plot average λ_{2D} against map coverage, noting which of these

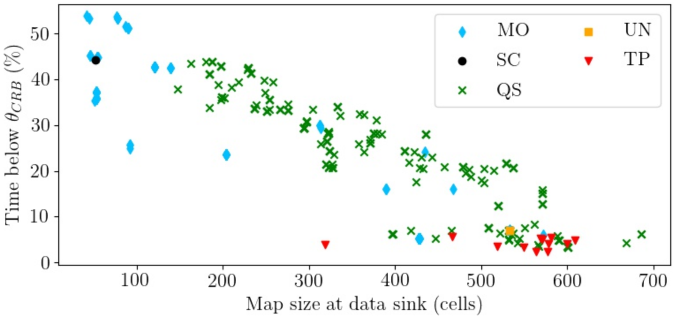

Fig. 3. Average percent of time spent with CRB below θ_{CRB} per robot against number of map cells known at data sink for 4 robots in 30x30 space with 20 obstacles. Each point is the average performance of 6 trials, and ideal is at the top right. We show more points for the more parameterized approaches (MO and our novel QS), while SC and UN are captured by a single point because they do not depend on parameters.

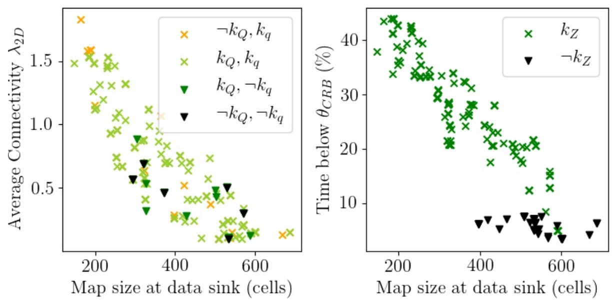

Fig. 4. QS connectivity and localizability performance with and without stabilizing the relevant virtual queues. Left: Average λ_{2D} against coverage. Right: Average percent of time spent with CRB below θ_{CRB} per robot against coverage. Notation $\neg k_Q$ indicates $k_Q = 0$, and in both plots ideal is at the top right.

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.

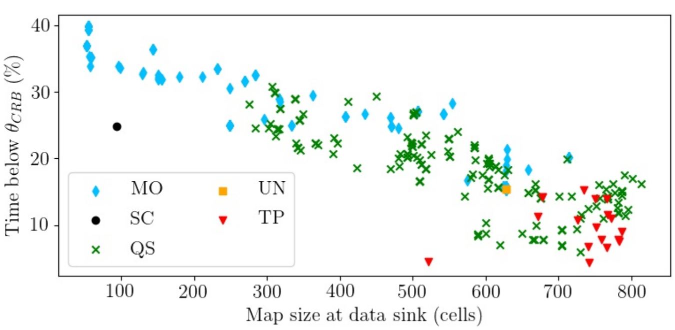

Fig. 5. Average percent of time spent with CRB below θ_{CRB} per robot against number of map cells known at data sink for 8 robots in 30x30 space with 20 obstacles. Each point is the average performance of 6 trials using 8-hop-connectivity, and ideal is at the top right.

Environment			TP: Max Cov.		QS: Max Cov.				
size	# rob.	# obs.	ρ	cov.	k_q	k_Q	k_Z	k_Y	cov.
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 I
SUMMARY OF RESULTS: MAXIMUM COVERAGE

C. Additional Experiments

In order to scale to larger networks, since exact calculation of p_{rD} grows intractable as the size of the network increases, we suggest a closely related metric. The k-hop-connectivity of vertices i and j in a graph is the weighted number of paths $i \to j$ of k hops or fewer, where weights indicate probability of success along the path [11]. In Fig. 5 we demonstrate results for eight robots using 8-hop-connectivity as a heuristic, and QS again slightly outperforms TP with respect to coverage and localizability.

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 I.

V. CONCLUSIONS AND FUTURE WORK

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 show that stabilizing a virtual delay queue allows us to constrain the average time between collecting data and transmitting it to the data sink, and extending this to meet hard deadlines is a direction for future work. Other directions for future work include revisiting the assumption of error-free movement and studying the relationship between the environment and gains k_q, k_Q, k_Z, and k_Y. For a set of gains, the system could be mathematically modeled as a Markov chain [47], and a formal analysis could lead to interesting results.

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, "Strategic knowledge gaps - 2: The "moon first" human exploration scenario," August 2016. [Online]. Available: https://www.nasa.gov/exploration/library/skg.html

[3] ——, "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

[4] 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.

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

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

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

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

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

[47] K. Lerman and A. Galstyan, "A general methodology for mathematical analysis of multi-agent systems," *ISI-TR-529, USC Information Sciences Institute, Marina del Rey, CA*, 2001.