Communication Jamming-Aware Robot Path Adaptation

Jonathan Diller[1] Neil Dantam[1] John Rogers[2] Qi Han[1]

***Abstract*—Robot networks are necessary for a variety of monitoring and surveillance applications where robots must communicate observations. This work considers robot networks that must operate in an area affected by potential communication jamming and excessive radio frequency noise. Specifically, we look at deploying a reinforcement robot to strengthen an existing network of robots that are already performing a monitoring task. The robot's objective is to find a location that maximizes the network's signal-to-noise ratio. We present a Gaussian process approach to predict areas of excessive radio frequency noise, enabling the robot to avoid these areas and relay more data. We evaluate our algorithm's performance in physical robot experiments using a Clearpath Jackal robot and a USRP software-defined radio to broadcast a jamming signal. Our results show major improvements in networking metrics over a baseline and demonstrate the need for jamming-aware robot planning.**

***Index Terms*—robotics, robot networks, self-adaptive behavior**

I. Introduction

Robots offer the potential to collect remote sensor data for monitoring, surveillance, and detection in hazardous situations [1], [2]. Robots may be arranged into *robot networks*, where the robots (which may be stationary for periods of time) collect data from the environment and relay that data to a central location through other robots. However, wireless interference—whether inadvertent or due to malicious jamming—may prevent the robots from communicating. Unlike stationary sensors, robots offer a key ability to overcome jamming: they can move outside of the jammed area if they know where that area is.

A major challenge in mapping jamming areas and regions with excessive noise is the high variance in radio frequency signals. A Gaussian process as a regression model is a probabilistic way to map out natural phenomena, such as radio frequencies, with a confidence metric for a prediction. *In this work, we use a Gaussian process approach for mobile robots to predict areas with communication jamming and excessive radio frequency noise and then incorporate this prediction into a robot path-planning algorithm for network reinforcement.*

Figure 1 illustrates our problem setup. Robot C must move to a position where it can relay data between robots A and B with minimal packet loss in the presence of wireless jamming. We assume that the robot has no prior knowledge of the jammer and must determine in real-time the best location to relay data through exploration. To the best of our knowledge, we are the first to propose a practical method for avoiding jamming in robot networks and implement it on a physical testbed to evaluate our solution's feasibility.

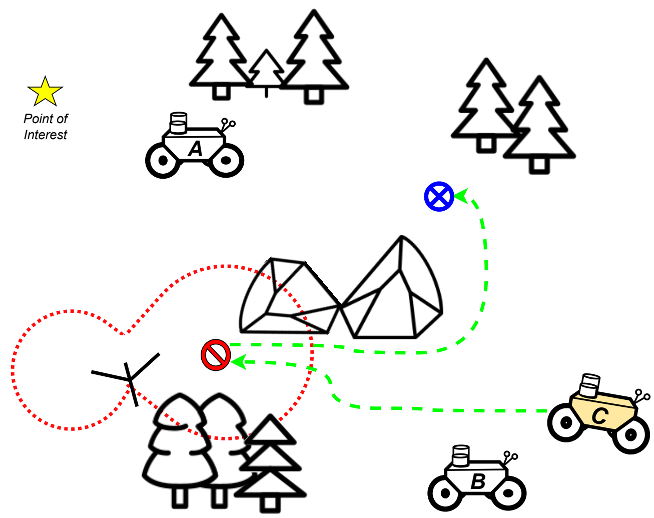


Fig. 1: General problem setup. The region inside the red dashed lines is affected by signal jamming. Robot C must position itself to reinforce the connection between robots A and B. Robot C initially picks the red marker as its position. Once there, robot C detects the jammer and determines that the best deployment location is the blue X.

In this work we make the following contributions:

1) we use a Gaussian Process to infer regions affected by a jammer without the need to fully map the region;
2) we present a jamming-aware online algorithm to deploy mobile robots; and
3) we validate our approach through physical robot experiments using Universal Software Radio Peripheral (USRP) software-defined radio platforms and a Clearpath Jackal robot.

The following section provides a review of related works. Section III formally defines our problem and the models and assumptions we use. Section IV presents our proposed methods for solving this problem and Section V describes our experimental setup and results.

II. Related Work

In this section we summarize different types of wireless jamming, how to detect and locate jammers, and methods for countering and avoiding jamming with robots.

[1]Colorado School of Mines, USA. {jdiller, ndantam, qhan}@mines.edu
[2]DEVCOM Army Research Laboratory, USA. john.g.rogers59.civ@army.mil
This work is supported in part by the ARL DCIST W911NF-17-2-0181 and NSF CNS-1823245.

A. Types of Communication Jamming

In general, there are four main categories of communication jamming techniques: (1) proactive; (2) reactive; (3) function-specific; and (4) smart and hybrid techniques [3]. Examples of proactive jamming include broadcasting constant noise over a wireless channel or sending out a constant stream of small, empty data packets [4]. Reactive jammers wait for other devices to attempt to communicate over a wireless channel then employ one of the proactive techniques. Function-specific jammers are designed for a specific function such as interfering with frequency hopping techniques and often affect multiple wireless channels. Smart and hybrid jammers take advantage of known wireless network architectures to disrupt larger networks.

B. Detecting & Locating Jamming

How to detect and counter a jamming device depends on the jamming technique being used. Proactive, reactive, and some function-specific jamming attacks can usually be detected using various wireless network quality metrics such as the received signal strength indicator, the signal-to-noise ratio, or packet loss rate. The more advanced jamming approaches may require technique-specific detection strategies depending on the approach used. We assume that the robot is capable of detecting and measuring jamming at a given position using well-studied techniques [3], [4], [5], [6].

If the wireless signal of a jammer is detected, then locating the jamming device using a mobile agent could be formulated as a signal mapping problem. An example of general signal mapping with a robot is seen in [7], where the authors use a Gaussian Process to locate a wireless access point. The signal mapping problem requires the robot to explore large areas of the environment which can be time-consuming. In our work, we consider how a robot can determine where to position itself by only mapping out relevant areas.

C. Countering & Avoiding Jamming with Robots

Typical jamming countermeasures include changing the transmission channel or rerouting network traffic. In this work, we assume that these methods are not sufficient and the robot must relocate itself (often referred to as "spatial retreat" [5]).

A lot of related work in spatial retreat assumes that the location of the jammer is known. For non-mobile jammers with known positions, the regions affected by jamming can be considered stationary obstacles. Mobile jammer scenarios with known or partially known positions are often solved using game theory approaches [8], [9]. In our work, the robot has no prior knowledge of jammed regions and must detect a jammer in real-time.

Spatial retreat methods for unmapped jammers often require the robot to drive away from the jammer. In [5], [10] the authors propose moving networked devices in a random direction until the device is no longer affected by jamming and then following the perimeter of the jammed area to attempt to reestablish network topology. The authors in [11] propose following a signal-to-noise ratio gradient to reposition robots away from noise sources. Our work builds on these prior solutions by adding a method for predicting noise and jamming. Additionally, these previous solutions were never evaluated on physical robot systems. In this work, we evaluate our solution on a physical robot testbed and discuss how to implement our solution on modern-day robots.

III. Problem Formulation & Modeling

In this section we introduce the scenario considered, discuss the models used, formally define the *Robot-Network-Deployment Problem*, and state the major assumptions made in this work.

A. Scenario Formulation

We address the robot network scenario illustrated in Figure 1. Let $\mathcal{X}$ be a two-dimensional continuous space consisting of disjoint obstacle region $\mathcal{X}_{\text{obs}}$ and free space $\mathcal{X}_{\text{free}}$. We assume that any point in $\mathcal{X}_{\text{free}}$is reachable by a robot, which can be determined using motion planning infeasibility proofs [12]. Let $\mathcal{X}_{\text{ns}}$ be a map of the radio frequency (RF) noise in $\mathcal{X}$. We assume that $\mathcal{X}_{\text{obs}}$ and $\mathcal{X}_{\text{free}}$ are known a priori but $\mathcal{X}_{\text{ns}}$ is unknown and may contain sub-regions with excessively high RF noise from other networked devices or communications jammers.

Let *A* and *B* be two robots already deployed in $\mathcal{X}$ at points x_a and x_b, respectively. A third robot *C* starts from point *p* and must position itself in $\mathcal{X}_{\text{free}}$ to reinforce a wireless connection between *A* and *B*.

B. Networking Models

For a relay robot to make communication-aware control decisions, we need a mathematical model for how well two robots can transmit data between each other. For this, we propose using a signal-to-noise ratio (SNR).

The signal strength received at a network node is inversely proportional to the distance between the sending node and the receiving node. As presented in [13], the power of a radio frequency signal measured at the receiver in a multi-propagation environment can be modeled as:

$$P(r) = P(r_0)(\frac{r_0}{r})^n \tag{1}$$

where r is the distance between the two network nodes and $P(r_0)$ is a reference power measurement at reference distance r_0. For small-scale communication systems, such as IEEE 802.11 (WiFi), r_0 is normally 1 m - 100 m. Variable n is the path-loss exponent which describes how the signal dampens with respect to distance and is environment dependent. In free space, we expect $n \approx 2$ while in an obstructed indoor environment we should expect $n \in [4, 6]$. In decibels, $P(r)$ can be represented as:

$$P(r)_{dB} = P(r_0)_{dB} + 10nlog(\frac{r_0}{r}) \tag{2}$$

Reference signal $P(r_0)_{dB}$ can be determined experimentally by measuring signal strength at distance r_0. However, a poorly measured $P(r_0)_{dB}$ could anchor a predicted distance to signal power relationship to a point that fails to properly represent the mapping from distance to signal power. Due to the stochastic nature of wireless communication, we instead incorporate $P(r_0)_{dB}$ and r_0 into an optimization problem to find n. If given a set of measurements I, where I consists of matching pairs of received signal strength measurement p_i and distance r_i between transmitter and receiver, then we can determine n as:

$$\min_{n,c} \sum_{i \in I} \left((-10nlog(r_i) + c) - p_i \right)^2 \tag{3}$$

where $c = P(r_0)_{dB} + 10nlog(r_0)$. Solving (3) only requires obtaining I and does not require us to select an r_0 or measure $P(r_0)_{dB}$ as determining both become part of the optimization problem.

Suppose that robot *A* at position x_a is sending data to robot *B* at position x_b. After solving for n and c for *A* and *B* in the deployment environment, we can predict the received signal strength of *A* sending data to *B* as:

$$P_{AB}(d_{ab})_{dB} = -10nlog(d_{ab}) + c \tag{4}$$

where d_{ab} is the straight line distance from x_a to x_b. If the RF noise at x_b is $P_n(x_b)_{dB}$, then the SNR of a data transmission from *A* to *B* will be:

$$SNR_{AB}(x_a, x_b) = \overbrace{P_{AB}(d_{ab})_{dB}}^{\text{signal strength at B}} - \overbrace{P_n(x_b)_{dB}}^{\text{RF noise B}} \tag{5}$$

Note that Equation 4 should consider if the two robots have line-of-sight (LoS) (i.e. there are no obstacles between the two robots). In practice, we recommend determining two versions of Equation 4: one for when the robots have line-of-sight (LoS) with each other and another for when the robots do not (nLoS).

C. Problem Definition

Reinforcement robot *C*'s goal is to explore $\mathcal{X}$ and position itself in a location that maximizes SNR for relaying data from source robot *A* to sink robot *B* while avoiding malicious communication jamming. The utility of deploying *C* at some point x_c to act as a wireless relay between *A* and *B* depends on how well it can receive data from *A* and relay it back to *B*. Let the utility of placing *C* at point x_c be

$$U(C) = \min\Big(\overbrace{SNR_{AC}(x_a, x_c)}^{\text{SNR: A to C}}, \overbrace{SNR_{CB}(x_c, x_b)}^{\text{SNR: C to B}} \Big), \tag{6}$$

which returns the smaller expected SNR between the wireless connections from *A* to *C* and from *C* to *B*.

We formally define the *Robot-Network-Deployment Problem* as follows:

Definition 1 (Robot-Network-Deployment Problem)**.** *Given space $\mathcal{X}$ consisting of disjoint obstacle region $\mathcal{X}_{\text{obs}}$ and free space $\mathcal{X}_{\text{free}}$, locations of robots A and B, and starting position p of robot C, find location x_c that maximizes $U(C)$, such that $x_c \in \mathcal{X}_{\text{free}}$ and the robot does not traverse $\mathcal{X}_{\text{obs}}$.*

After experimentally determining $SNR_{AC}()$ and $SNR_{CB}()$, we could solve the *Robot-Network-Deployment Problem* offline if we knew $P_n(x)_{dB}$ for any $x \in \mathcal{X}$. However, because we assume no prior knowledge of $\mathcal{X}_{\text{ns}}$, the RF noise in $\mathcal{X}$, we do not know $P_n(x)_{dB}$ for all $x \in \mathcal{X}$ and the robot must explore $\mathcal{X}$ to determine an optimal x_c.

IV. Proposed Algorithms

To solve the *Robot-Network-Deployment Problem*, we propose an online approach that measures and responds to jamming. The key novelty in our approach is representing RF noise and jammed areas using Gaussian processes, measuring noise online, and updating the representation of jammed areas.

A. Gaussian Processes to Represent Jamming

In general, a GP is a collection of random variables where any subset of two or more of these random variables has a multivariate Gaussian distribution [14]. In GP regression, we assume that the regression function $\mathbf{f}_*$ that maps inputs to outputs is distributed as a Gaussian process. This is represented mathematically as a normal distribution:

$$\mathbf{f}_* \sim \mathcal{N}\big(\overbrace{\mathbf{0}}^{mean}, \overbrace{\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)}^{covariance}\big) \tag{7}$$

where the distribution has a mean of 0 and $\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$ is the covariance matrix of prior input knowledge. If we have t prior samples, we can use the set of prior input knowledge $\mathbf{X}_t$ and prior outputs $\mathbf{y}_t$ to find a vector of weight $\mathbf{w} = [\mathbf{K}(\mathbf{X}_t, \mathbf{X}_t) + \sigma_\epsilon^2 \mathbf{I}]^{-1} \mathbf{y}_t$ where σ_ϵ^2 is the covariance of the input noise. To make a prediction of a new input x, we then compute:

$$\sum_{i=0}^{t} w_i k(x_i, x) \tag{8}$$

where w_i is the i^{th} entry in $\mathbf{w}$ and $k(x_i, x)$ is the covariance between prior sample x_i and x.

To represent RF noise as a GP we use the location in $\mathcal{X}$ as the input to the GP and noise strength at that location as the output. By fitting a GP to location and noise strength measurements, we can predict regions of high RF noise and jamming even for areas we have not directly measured. The robot uses a GP to predict jamming and RF noise, i.e. $P_n(x)_{dB}$, for points in $\mathcal{X}$ that have not been explored.

We address areas with high uncertainty by only accepting predictions that have low variance in the GP model at that location. Specifically, if the variance of a prediction exceeds a maximum allowable threshold ρ, we discard that prediction and assume that the RF noise strength for that cell is equal to the noise floor of the wireless spectrum. The noise floor will vary depending on the environment and equipment used, but is generally around -100 dB. Ignoring weak predictions prevents the robot from making decisions with low-confidence data.

B. Updating the Jamming Gaussian Process

While the robot explores the environment, it measures the current jamming signals. If jamming is detected, then the robot records the strength of the signal in dB and the location where the reading was taken. If there is no jamming signal present, then the robot records the noise floor of the wireless channel being used, which is also recorded in dB.

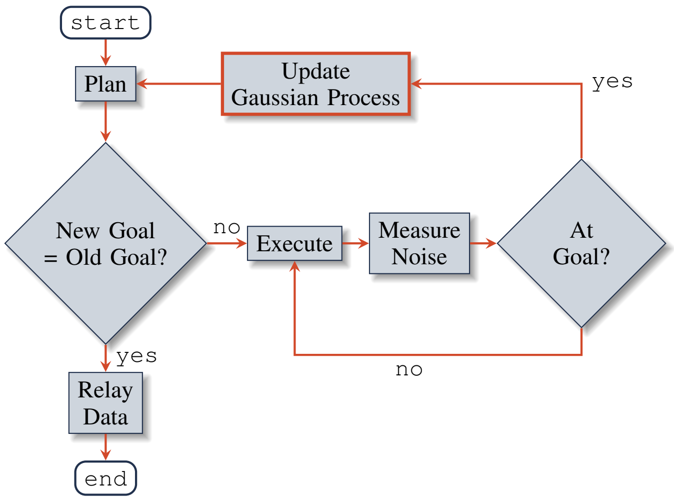


Fig. 2: Flowchart of the proposed online algorithm to solve the *Robot-Network-Deployment Problem*.

To update its knowledge of jammed areas, the robot uses the signal recordings as prior knowledge for the GP. The location where a signal was recorded is used as the prior input and the recorded signal strength is used as the prior output. The strength of a wireless signal at any given point tends to vary greatly due to a multitude of factors, so we group reading locations into discrete map coordinates and average the reading values to prevent excessive variations in our prior knowledge.

C. Planning & Execution

Our online algorithm for solving the *Robot-Network-Deployment Problem* and finding x_c is illustrated in Figure 2. The robot explores the environment until it finds an unjammed location that maximizes signal strength between A to B. First, the robot uses the GP to find an optimal goal location that it expects will avoid jamming and maximize SNR. Second, the robot plans and moves to that location, measuring jamming along the way. Third, the robot updates the GP and runs the algorithm again until the algorithm converges on a deployment location.

To identify a goal location, the robot conducts a linear scan through a discretized map of $\mathcal{X}$. For each discrete location x' in $\mathcal{X}$, the robot calculates the utility gained, $U(C)$, from positioning itself at x'. To compute $U(C)$, the robot considers whether or not it has line-of-sight (LoS) with both A and B, using different versions of $SNR_{AC}()$ and $SNR_{CB}()$ based on LoS. The robot determines LoS using Bresenham's line algorithm on a discretized version of $\mathcal{X}_{\text{free}} \bigcup \mathcal{X}_{\text{obs}}$ [15]. To predict noise for each cell, the robot either queries prior measurements for that cell or makes an RF noise prediction as described in Section IV-A. The robot selects a goal location, x'_c, as the x' with the greatest value of $U(C)$. Larger discretizations of $\mathcal{X}$ will reduce the number of cells the planner must search at the cost of position accuracy while smaller discretizations of $\mathcal{X}$ will improve accuracy but increase search time.

Next, the robot plans a path to goal location x'_c and moves. For 2D space $\mathcal{X}$, we plan paths by running A* on a discretized version of $\mathcal{X}_{\text{free}} \bigcup \mathcal{X}_{\text{obs}}$. When the robot reaches x'_c, it repeats the steps described above to find a new goal location x''_c. If $x'_c = x''_c$, then the algorithm has converged on a final goal location and the robot stops exploring the environment. If $x'_c \neq x''_c$, then the robot continues to run the exploration algorithm until the algorithm converges on a final location.

V. Performance Evaluation

We evaluated our algorithm in two different case studies using physical robots and radios. The first was in a network of hallways in a large city-block shape, and the second was in a long, straight hallway with an intersecting passage in the middle. We used a Clearpath Jackal robot as the robot to be deployed, a Raspberry Pi 4 and laptop to model additional robots already deployed, and the USRP B205mini-i software-defined radio (SDR) platform from Ettus Research as a jammer. We chose to use an SDR for the jamming device because they are flexible and give us control over the jammer's behavior. We used the Better Approach to Mobile Ad-hoc Networking (BATMAN) routing protocol to establish an ad-hoc network between robots [16].

In the following subsections, we discuss further discuss our experimental setup and discuss the results of our two case studies. We used packet loss rate and Netflix's VMAF scores on video streams as our evaluation metrics [17]. Our results show that communication jamming can impede robot networks and that using a GP for noise prediction can improve networked robot deployment.

A. Generation and Detection of Jamming Signals

We created a WiFi jammer by broadcasting a Gaussian noise over a single channel at the 2.4 *GHz* band. The noise is generated with a 22 *MHz* bandwidth, which covers the entirety of a 2.4 *GHz* WiFi channel. This noise will disrupt other wireless devices on the same frequency by corrupting other wireless signals or by keeping the channel busy, preventing well-behaved devices from transmitting.

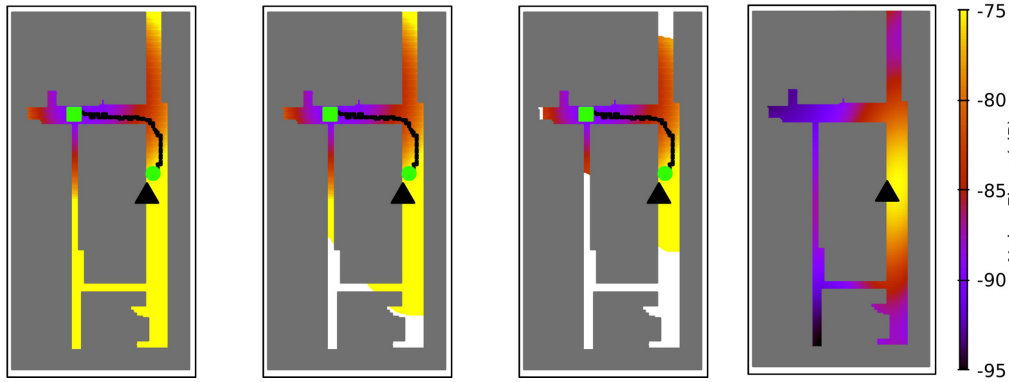
To sense the presence of the jammer, we used a second B205mini-i SDR that measures the noise floor of the 2.4 *GHz* band. The noise detecting SDR takes rapid signal measurements over the designated WiFi channel and averages them together to determine the noise floor of the channel. If the average of the readings is notably higher than the normal noise floor then we can tell that constant noise is being used to disrupt the wireless channel.

B. Determining n and c

As discussed in Section III-B, to use Equation 4 (our online algorithm's utility function) to predict signal strength for any point in $\mathcal{X}$ we must solve Equation 3 (determine n and c) by taking signal strength measurements at varying distances in the deployment environment for each pair of robots.

We set up one Raspberry Pi (acting as robot A) and a second Raspberry Pi (attached to robot C for data relay) at varying

 http://dx.doi.org/10.1109/DCOSS-IoT58021.2023.00120



(a) $\rho = 1.0$ (b) $\rho = 0.25$ (c) $\rho = 0.0625$ (d) Fully mapped area

Fig. 3: Results of mapping a hallway with a GP while adjusting ρ. Figures 3a, 3b, 3c show the results of the robot making a short move then mapping the noise floor with varying values of ρ. The black triangle is the location of the jammer. The black line shows the path of the robot where the green square is the robot's starting position and the green circle is where the robot stopped.

distances apart and measured the received signal strength at *C* both with and without LoS. We repeated these experiments for the Raspberry Pi attached to *C* and a laptop used as a robot *B*. We ran this experiment in both the block hallway and the long hallway, taking a total of 3,500 measurements for each setup.

To determine values for n and c for each setup, we used the SciPy implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm on Eq. (3). The BFGS algorithm is an iterative numerical optimization algorithm that determines the gradient of an unconstrained non-linear function by approximating the Hessian matrix of the function using the secant method [18]. We chose the BFGS algorithm because it is a Newton-like method that avoids the need to repeatedly solve for the Hessian on large sets of input data.

Figure 4 shows the results of our experiments to determine $P()_{dB}$ between robots A and C in the block hallway. For the LoS case, we found $n = 0.630$ and $c = -60.86$ and for NLoS we found $n = 1.111$ and $c = -60.24$. We note that these values for n are much lower than estimated in subsection III-B. We believe this is due to the low transmission gain setting on the robots, which is needed to force multi-hop connections at short ranges.

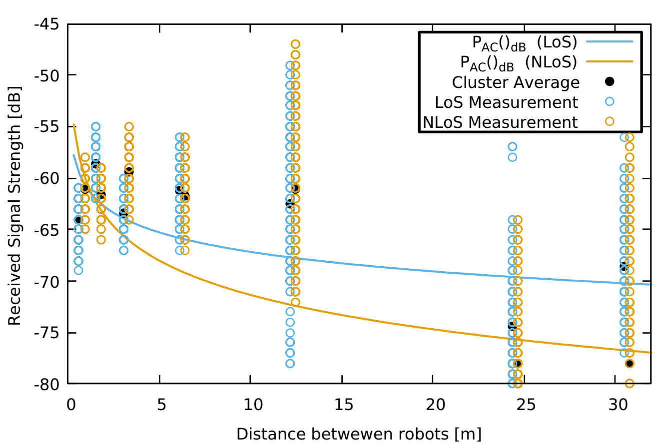


Fig. 4: Experiments to determine $P()_{dB}$ between robots *A* and *C* in the block hallway.

C. *Impact of Variance Threshold (ρ) on GP Prediction Error*

To evaluate the impact of the variance threshold (ρ) on overall model error, we compared GP-based predictions for varying values of ρ against a fully-explored map and found the mean percentage error (MPE) in the prediction. The results are shown in Table I. Figures 3a, 3b, and 3c show a heat map of the GP predictions with $\rho = 1$, 0.25, and 0.0625, respectively. Figure 3d shows the actual AoE of the jammer after fully mapping out the entire space. Our results show that MPE starts to exceed 5% when the variance exceeds 0.0625 and recommend a value of 0.05 for ρ.

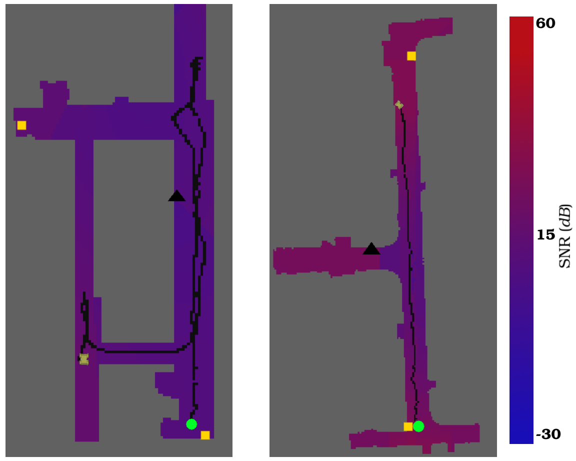
D. *Physical Robot Experiments*

We evaluated our algorithm in two areas as shown in Figure 5 located in Brown Hall on the Colorado School of Mines campus. The first is in a network of hallways in the shape of a city block, and the area has partitioned walls with small offices in the closed-off areas. The second is in a long hallway with a large T-intersection in the middle, and the area has block walls with laboratories in the closed-off areas.

We compared our algorithm ("GP") against a jamming-agnostic approach ("JA"). JA selects a position for *C* that maximizes equation (6) and does not consider the jammer. We

$\rho =$	1	0.5	0.25	0.125	0.0625	0.0313
MPE	22.120	20.149	12.822	8.134	5.379	3.499

TABLE I: Mean percentage error (MPE) as the value of ρ changes.



(a) (b)

Fig. 5: Example results of our algorithm on the physical testbed for the block hallway (a) and the long hallway (b). The black triangles show the location of the jammer, the yellow squares depict the deployed robots, and the green circles show the starting point of the network reinforcement robot. The robot icon shows the final deployment location and the SNR heat map shows the robot's final SNR prediction.

ran five trials in each testing area. We set $\rho = 0.05$ and experimentally determine n and c as described in subsection V-B for both areas. Figure 5 shows example runs of our algorithm.

Once the algorithm terminates and the robot reaches its final goal, we attempted to ping node A from node B (via C) 200 times and recorded the packet-loss-ratio (PLR). We also attempted to stream a one-minute video from A to B (via C) and computed the VMAF score of the received video. For both of these metrics, a larger number is better. Table II shows the results of our trials in both areas. On average, our GP method experienced a PLR of 0.865, an improvement of 251.6% over the JA baseline. The average VMAF score for our GP method was 9.19, a 173.1% improvement compared to the JA score of 3.365.

	Block Hallway		**Long Hallway**		**Total**	
	JA	GP	JA	GP	JA	GP
PLR	0.578	0.999	0.401	0.731	0.4895	**0.865**
Std. dev.	0.528	0.002	0.292	0.302	0.413	0.246
VMAF	8.014	8.660	0	9.720	4.007	**9.190**
Std. dev.	1.651	2.787	0	4.124	4.365	3.365

TABLE II: Average packet loss ratio (PLR) and VMAF scores with standard deviation for each test case.

VI. Conclusions

In this study, we investigated a robotics application where a ground robot positions itself in a robot network. We proposed an online algorithm for the robot to explore the area and determine a suitable location to relay data while monitoring for malicious communications jamming. We proposed a method using a Gaussian process to predict the Area of Effect of a jammer that limits predictions based on the variance in the model.

We evaluated our proposed algorithm in a series of physical robot experiments in two indoor areas. Our GP-based approach for predicting RF noise leads to a 252% improvement in the packet-loss-ratio of a baseline and a 173% improvement in video stream quality. These results highlighted the need for jamming-aware robot planning algorithms and how noise prediction can improve connectivity in networked robots.

Future work will focus on the multi-robot variation of this problem. We also see future areas of work in heterogeneous vehicle problems such as collaborative aerial and ground vehicle teams that must accomplish a task while avoiding communications jamming.

References

[1] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: an up-to-date survey," *Applied System Innovation*, vol. 3, no. 1, p. 14, 2020.

[2] G. Tuna, V. C. Gungor, and K. Gulez, "An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters," *Ad Hoc Networks*, vol. 13, pp. 54–68, 2014.

[3] K. Grover, A. Lim, and Q. Yang, "Jamming and anti-jamming techniques in wireless networks: A survey," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 17, no. 4, pp. 197–215, 2014.

[4] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 46–57.

[5] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: defenses against wireless denial of service," in *Proceedings of the 3rd ACM workshop on Wireless security*, 2004, pp. 80–89.

[6] M. Çakiroğlu and A. T. Özcerit, "Jamming detection mechanisms for wireless sensor networks," in *Proceedings of the 3rd international conference on Scalable information systems*, 2008, pp. 1–8.

[7] J. Fink and V. Kumar, "Online methods for radio signal mapping with mobile robots," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1940–1945.

[8] S. Bhattacharya and T. Başar, "Game-theoretic analysis of an aerial jamming attack on a uav communication network," in *proceedings of the 2010 American control conference*. IEEE, 2010, pp. 818–823.

[9] ——, "Graph-theoretic approach for connectivity maintenance in mobile networks in the presence of a jammer," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 3560–3565.

[10] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE network*, vol. 20, no. 3, pp. 41–47, 2006.

[11] C. Dixon and E. W. Frew, "Maintaining optimal communication chains in robotic sensor networks using mobility control," *Mobile Networks and Applications*, vol. 14, pp. 281–291, 2009.

[12] S. Li and N. T. Dantam, "Learning proofs of motion planning infeasibility," in *Robotics: Science and Systems*, 2021.

[13] M. Safak, *Digital communications*. John Wiley & Sons, 2017.

[14] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[15] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems journal*, vol. 4, no. 1, pp. 25–30, 1965.

[16] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better approach to mobile ad-hoc networking (batman)," *IETF draft*, pp. 1–24, 2008.

[17] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. Cock, "Vmaf: The journey continues," *Netflix Technology Blog*, vol. 25, 2018.

[18] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.